New Theoretical Frameworks for Machine Learning

Maria-Florina Balcan

CMU-CS–08-153
September 15th, 2008

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Avrim Blum, Chair
Manuel Blum
Yishay Mansour
Tom Mitchell
Santosh Vempala

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was sponsored by the National Science Foundation under grant numbers IIS-0121678, IIS-0312814, CCF-0514922, CCR-0122581, the U.S. Army Research Office under grant number DAAD-190213089, Google, and the IBM Ph.D. Fellowship. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity

Keywords: Data Dependent Concept Spaces, Clustering, Value of Unlabeled Data, Semi-supervised Learning, Active Learning, Co-training, Similarity-based Learning, Kernels, Margins, Low-Dimensional Mappings, Sample Complexity, Mechanism and Auction Design, Random Sampling Mechanisms, Profit Maximization.

In memoria tatalui meu. Vei ramane vesnic in suflet, inima si gand!

Abstract

This thesis has two primary thrusts. The first is developing new models and algorithms for important modern and classic learning problems. The second is establishing new connections between Machine Learning and Algorithmic Game Theory.

The formulation of the PAC learning model by Valiant [201] and the Statistical Learning Theory framework by Vapnik [203] have been instrumental in the development of machine learning and the design and analysis of algorithms for supervised learning. However, while extremely influential, these models do not capture or explain other important classic learning paradigms such as Clustering, nor do they capture important emerging learning paradigms such as Semi-Supervised Learning and other ways of incorporating unlabeled data in the learning process. In this thesis, we develop the first analog of these general discriminative models to the problems of Semi-Supervised Learning and Clustering, and we analyze both their algorithmic and sample complexity implications. We also provide the first generalization of the well-established theory of learning with kernel functions to case of general pairwise similarity functions and in addition provide new positive theoretical results for Active Learning. Finally, this dissertation presents new applications of techniques from Machine Learning to Algorithmic Game Theory, which has been a major area of research at the intersection of Computer Science and Economics.

In machine learning, there has been growing interest in using unlabeled data together with labeled data due to the availability of large amounts of unlabeled data in many contemporary applications. As a result, a number of different semi-supervised learning methods such as Co-training, transductive SVM, or graph based methods have been developed. However, the underlying assumptions of these methods are often quite distinct and not captured by standard theoretical models. This thesis introduces a new discriminative model (a PAC or Statistical Learning Theory style model) for semi-supervised learning, that can be used to reason about many of the different approaches taken over the past decade in the Machine Learning community. This model provides a unified framework for analyzing when and why unlabeled data can help in the semi-supervised learning setting, in which one can analyze both sample-complexity and algorithmic issues. In particular, our model allows us to address in a unified way key issues such as "Under what conditions will unlabeled data help and by how much?" and "How much data should I expect to need in order to perform well?".

Another important part of this thesis is Active Learning for which we provide several new theoretical results. In particular, this dissertation includes the first active learning algorithm which works in the presence of arbitrary forms of noise, as well as a few margin based active learning algorithms.

In the context of Kernel methods (another flourishing area of machine learning research), this thesis shows how Random Projection techniques can be used to convert a given kernel function into an explicit, distribution dependent set of features, which can then be fed into more general (not necessarily kernelizable) learning algorithms. In addition, this work shows how such methods can be extended to more general pairwise similarity functions and also gives a formal theory that matches the standard intuition that a good kernel function is one that acts as a good measure of similarity. We thus strictly generalize and simplify the existing theory of kernel methods. Our approach brings a new perspective as well as a much

simpler explanation for the effectiveness of kernel methods, which can help in the design of good kernel functions for new learning problems.

We also show how we can use this perspective to help thinking about Clustering in a novel way. While the study of clustering is centered around an intuitively compelling goal (and it has been a major tool in many different fields), reasoning about it in a generic and unified way has been difficult, in part due to the lack of a general theoretical framework along the lines we have for supervised classification. In our work we develop the first general discriminative clustering framework for analyzing accuracy without probabilistic assumptions.

This dissertation also contributes with new connections between Machine Learning and Mechanism Design. Specifically, this thesis presents the first general framework in which machine learning methods can be used for reducing mechanism design problems to standard algorithmic questions for a wide range of revenue maximization problems in an unlimited supply setting. Our results substantially generalize the previous work based on random sampling mechanisms – both by broadening the applicability of such mechanisms and by simplifying the analysis. From a learning perspective, these settings present several unique challenges: the loss function is discontinuous and asymmetric, and the range of bidders' valuations may be large.

Acknowledgments

CMU is certainly the best place and Avrim Blum is certainly the best advisor for a learning theory thesis. Under his guidance I was able to think at a fundamental level about a variety of types of machine learning questions: both classic and modern, both technical and conceptual. In addition, he shared with me his incredibly sharp insights and great expertise in other areas, including but not limited to game theory and algorithms. I can definitely say we have worked together on the most interesting problems one could imagine.[1] Among many other things, he taught me how in a research problem it is most crucial above all, to single out the key questions and then to solve them most elegantly.

During my Ph.D years I also had the privilege to learn, observe, and "steal" many tricks of the trade from other awesome researchers. In particular, I would like to name my thesis committee members Manuel Blum, Yishay Mansour, Tom Mitchell, and Santosh Vempala, as well as two other valuable collaborators: Michael Kearns and Tong Zhang. Yishay has always been a great source of technical insight in a variety of areas ranging from pure machine learning topics to algorithms or game theory. Santosh has also lent me his insights in seemingly impossible technical questions. Tom and Manuel besides being the most charismatic faculty members at CMU, have also been great models for shaping up my own perspective on computer science research. Michael and Tong have also been particularly great to interact with.

I would like to thank all my other collaborators and co-authors, inside and outside CMU, for making both the low-level research (paper writing, slide preparing, conference calling, coffee drinking) and high level research (idea sharing, reference pointing, and again coffee drinking) a lot of fun. Particular gratitude goes to Alina Beygelzimer, John Langford, Ke Yang, Adam Kalai, Anupam Gupta, Jason Hartline, and Nathan Srebro.

I am grateful to George Necula for laying out the Ph.D. map for me, for advertising CMU as one of the best places on Earth, and for all his altruistic advice in the past seven years. Andrew Gilpin has been my only office mate (ever!) who has learnt key phrases in a foreign language especially so that he can interact with me better (and well, to visit Romania as well). Much encouragement or feedback on various things (including various drafts[2] and talks) I have gotten from Steve Hanneke, Ke Yang, and Stanislav Funiak.

My undergraduate days at the University of Bucharest would have simply not been the same without Professors Luminita State, Florentina Hristea, and Ion Vaduva. They have all influenced me with their style, taste, and charisma.

I am indebted to all my friends (older and newer) and my family for their support throughout the years. Particularly, I thank my brother Marius both for his brilliant idea to nickname me Nina (when he was too young to appreciate longer names) and for coping with having a studious sister.[3] Most of what I am today is due to my parents Ioana and Dumitru who, without even planning to make a scientist out of me, offered me an extraordinary model on how to be serious and thorough about what I should chose to do in my life.

And of course, I thank Doru for always being "one of a kind".

[1] Though sometimes it took us a while to formulate and agree on them.

[2] All the remaining typos in this thesis are entirely my responsibility.

[3] Finishing this thesis on September 15th is my birthday present for him.

Contents

Chapter 1

Introduction

The formulation of the classic discriminative models for Supervised Learning, namely the PAC learning model by Valiant [201] and the Statistical Learning Theory framework by Vapnik [203], were instrumental in the development of machine learning and the design and analysis of algorithms for supervised learning. However, while very influential, these models do not capture or explain other important classic learning paradigms such as Clustering, nor do they capture important emerging learning paradigms such as Semi-Supervised Learning and other ways of incorporating unlabeled data in the learning process. In this thesis, we develop new frameworks and algorithms for addressing key issues in several important classic and modern learning paradigms. In particular, we study Semi-Supervised Learning, Active Learning, Learning with Kernels and more general similarity functions, as well as Clustering. In addition, we present new applications of techniques from Machine Learning to emerging areas of Computer Science, such as Auction and Mechanism Design.

We start with a high level presentation of our work, and then in Section 1.1 we give a more detailed overview of the main contributions of this thesis in each of the main directions. In Section 1.2 we summarize the main results and describe the structure of this thesis, as well as provide bibliographic information.

New Frameworks and Algorithms for Machine Learning Over the years, machine learning has grown into a broad discipline that has produced fundamental theories of learning processes, as well as learning algorithms that are routinely used in commercial systems for speech recognition, computer vision, and spam detection, to name just a few. The primary theoretical advances have been for *passive supervised* learning problems [172], where a target function (e.g., a classifier) is estimated using only labeled examples which are considered to be drawn i.i.d. from the whole population. For example, in spam detection an automatic classifier to label emails as spam or not would be trained using a sample of previous emails labeled by a human user. However, for most contemporary practical problems there is often useful additional information available in form of cheap and plentiful *unlabeled* data: e.g., unlabeled emails for the spam detection problem. As a consequence, there has recently been substantial interest in Semi-Supervised Learning, a method for using unlabeled data together with labeled data to improve learning. Several different semi-supervised learning algorithms have been developed and numerous successful experimental results have been reported. However the underlying assumptions of these methods are quite different and their effectiveness cannot be explained by standard learning models (the PAC model or the Statistical Learning Theory framework). While many of these methods had theoretical justification under specific assumptions, there has been no unified framework for semi-supervised learning in general. In this thesis, we develop a comprehensive theoretical framework that provides a unified way for thinking about semi-supervised learning; this model can be used to reason about many of the different approaches taken

over the past decade in the machine learning community.[1]

In the context of Active Learning (another modern learning paradigm in which the algorithm can interactively ask for the labels of unlabeled examples of its own choosing), we present several new theoretical results. In particular we describe the first active learning procedure that works in the presence of arbitrary forms of noise. This procedure relies only upon the assumption that samples are drawn i.i.d. from some underlying distribution and it makes no assumptions about the mechanism producing the noise (e.g., class/target misfit, fundamental randomization, etc.). We also present theoretical justification for margin-based algorithms which have proven quite successful in practical applications, e.g., in text classification [199].

Another important component of this thesis is the development of more intuitive and more operational explanations for well-established learning paradigms, for which a solid theory did exist, but it was too abstract and disconnected from practice. In particular, in the context of Kernel methods (a state of the art technique for supervised learning and a flourishing area of research in modern machine learning), we develop a theory of learning with similarity functions that provides theoretical justification for the common intuition that a good kernel function is one that acts as a good measure of similarity. This theory is strictly more general and involves more tangible quantities than those used by the traditional analysis.

Finally, we also present a new perspective on the classic Clustering problem. Problems of clustering data from pairwise similarity information are ubiquitous in science and as a consequence clustering received substantial attention in many different fields for many years. The theoretical work on the topic has generally been of two types: either on algorithms for (approximately) optimizing various distance-based objectives such as k-median, k-means, and min-sum, or on clustering under probabilistic "generative model" assumptions such as mixtures of Gaussian or related distributions. In this thesis we propose a new approach to analyzing the problem of clustering. We consider the goal of approximately recovering an unknown target clustering using a similarity function (or a weighted graph), given only the assumption of certain natural properties that the similarity or weight function satisfies with respect to the desired clustering. Building on our models for learning with similarity functions in the context of supervised classification, we provide the first general discriminative clustering framework for analyzing clustering accuracy without probabilistic assumptions. In this model we directly address the fundamental question of what kind of information a clustering algorithm needs in order to produce a highly accurate clustering of the data, and we analyze both information theoretic and algorithmic aspects.

At a technical level, a common characteristic of many of the models we introduce to study these learning paradigms (e.g., semi-supervised learning or learning and clustering via similarity functions) is the use of *data dependent concept spaces*, which we expect to be a major line of research in the next years in machine learning. The variety of results we present in these models relies on a very diverse set of insights and techniques from Algorithms and Complexity, Empirical Processes and Statistics, Optimization, as well as Geometry and Embeddings.

Connections between Machine Learning and Algorithmic Game Theory This thesis also includes a novel application of machine learning techniques to *automate* aspects of Mechanism Design and formally address the problem of market analysis, as well as development of pricing algorithms with improved guarantees over previous methods.

Developing algorithms for a highly distributed medium such as the Internet requires a careful consideration of the objectives of the various parties in the system. As a consequence, Mechanism Design has become an increasingly important part of algorithmic research and computer science more generally in

[1] This model appears in a recent book about Semi-Supervised Learning [27] and it can be used to explain when and why unlabeled data can help in many of the specific methods given in the other chapters of the book.

recent years. Mechanism design can be thought of as a distinct form of algorithm design, where a central entity must perform some computation (e.g., resource allocation or decision making) under the constraint that the agents supplying the inputs have their own interest in the outcome of the computation. As a result, it is desirable that the employed procedure be incentive compatible, meaning that it should be in each agent's best interest to report truthfully, or to otherwise act in a well-behaved manner. Typical examples of such mechanisms are auctions of products (e.g., software packages) or pricing of shared resources (e.g. network links) where the central entity would use inputs (bids) from the agents in order to allocate goods in a way that maximizes its revenue. Most of the previous work on incentive compatible mechanism design for revenue maximization has been focused on very restricted settings [122, 174] (e.g., one item for sale and/or single parameter agents), and many of the previous incentive compatible mechanisms have been "hand-crafted" for the specific problem at hand. In this thesis we use techniques from machine learning to provide a *generic reduction* from the incentive-compatible mechanism design question to more standard algorithmic questions, for a wide variety of revenue-maximization problems, in an unlimited supply setting.

1.1 Overview

A more detailed overview of this thesis follows below.

1.1.1 Incorporating Unlabeled Data in the Learning Process

As mentioned earlier, machine learning has traditionally focused on problems of learning a task from labeled examples only. However, for many contemporary practical problems such as classifying web pages or detecting spam, there is often additional information available; in particular, for many of these settings unlabeled data is often much cheaper and more plentiful than labeled data. As a consequence, there has recently been substantial interest in using unlabeled data together with labeled data for learning [59, 62, 135, 141, 159, 176, 181, 215], since clearly, if useful information can be extracted from it that reduces dependence on labeled examples, this can be a significant benefit [58, 172].

There are currently several settings that have been considered for incorporating unlabeled data in the learning process. Here, in addition to a set of labeled examples drawn at random from the underlying data distribution, it is assumed that the learning algorithm can also use a (usually much larger) set of unlabeled examples from the same distribution.

A first such setting is passive *Semi-Supervised Learning* (which we will refer to as SSL) [6]. What makes unlabeled data so useful in the SSL context and what many of the SSL methods exploit, is that for a wide variety of learning problems, the natural regularities of the problem involve not only the *form* of the function being learned by also how this function *relates* to the distribution of data. For example, in many problems one might expect the target function should cut through low density regions of the space, a property used by the transductive SVM algorithm [141]. In other problems one might expect the target to be self-consistent in some way, a property used by Co-training [62]. Unlabeled data is then potentially useful in this setting because, in principle, it allows one to reduce search space from the whole set of hypotheses, down to the set of *a-priori* reasonable ones with respect to the underlying distribution.

A second setting which has been considered for incorporating unlabeled data in the learning process which has been increasingly popular for the past few years, is *Active Learning* [86, 94]. Here, the learning algorithm has both the capability of drawing random unlabeled examples from the underlying distribution, and that of asking for the labels of *any* of these examples. The hope is that a good classifier can be learned with significantly fewer labels by *actively* directing the queries to *informative* examples. As opposed to

the SSL setting, and similarly to the classical supervised learning settings (PAC and Statistical Learning Theory settings) the only prior belief about the learning problem in the active learning setting is that the target function (or a good approximation of it) belongs to a given concept class. Luckily, it turns out that for simple concept classes such as linear separators on the line one can achieve an *exponential* improvement (over the usual supervised learning setting) in the labeled data sample complexity, under no additional assumptions about the learning problem [86, 94]. In general, however, for more complicated concept classes, the speed-ups achievable in the active learning setting depend on the match between the distribution over example-label pairs and the hypothesis class. Furthermore, there are simple examples where active learning does not help at all, not even in the realizable case [94].

In this thesis we study both Active Learning and Semi-Supervised Learning. For the semi-supervised learning problem, we provide a *unified discriminative model* (i.e., a PAC or Statistical Learning Theory style model) that captures many of the ways unlabeled data is typically used, and provides a very general framework for thinking about this issue. This model provides a unified framework for analyzing when and why unlabeled data can help, in which one can discuss both sample-complexity and algorithmic issues. Our model can be viewed as an extension of the standard PAC model, where in addition to a concept class C, one also proposes a compatibility function (an abstract prior): a type of compatibility that one believes the target concept should have with the underlying distribution of data. For example, such a belief could be that the target should cut through a low-density region of space, or that it should be self-consistent in some way as in co-training. This belief is then explicitly represented in the model. Unlabeled data is then potentially helpful in this setting because it allows one to estimate compatibility over the space of hypotheses, and to reduce the size of the search space from the whole set of hypotheses C down to those that, according to one's assumptions, are a-priori reasonable with respect to the distribution. After proposing the model, we analyze fundamental sample-complexity issues in this setting such as "How much of each type of data one should expect to need in order to learn well?", and "What are the basic quantities that these numbers depend on?". We present a variety of sample-complexity bounds, both in terms of uniform-convergence results—which apply to any algorithm that is able to find rules of low error and high compatibility—as well as ϵ-cover-based bounds that apply to a more restricted class of algorithms but can be substantially tighter. For instance, we describe several natural cases in which ϵ-cover-based bounds can apply even though with high probability there still exist bad hypotheses in the class consistent with the labeled and unlabeled examples. Finally, we present several PAC-style algorithmic results in this model. Our main algorithmic result is a new algorithm for Co-Training with linear separators that, if the distribution satisfies independence given the label, requires only a single labeled example to learn to any desired error rate ϵ and is computationally efficient (i.e., achieves PAC guarantees). This substantially improves on the results of [62] which required enough labeled examples to produce an initial weak hypothesis. We describe these results in Chapter 2.

For the active learning problem, we prove for the first time, the feasibility of agnostic active learning. Specifically we propose and analyze the first active learning algorithm that finds an ϵ-optimal hypothesis in any hypothesis class, when the underlying distribution has arbitrary forms of noise. We also analyze margin based active learning of linear separators. We discuss these results in Chapter 5. Finally, we mention recent work in which we have shown that in an asymptotic model for active learning where one bounds the number of queries the algorithm makes before it finds a good function (i.e. one of arbitrarily small error rate), but not the number of queries before it *knows* it has found a good function, one can obtain significantly better bounds on the number of label queries required to learn than in the traditional active learning models.

In addition to being helpful in the semi-supervised Learning and active learning settings, unlabeled data becomes useful in other settings as well, both in partially supervised learning models and, of course,

in purely unsupervised learning (e.g., clustering). In this thesis we study the use of unlabeled data in the context of learning with Kernels and more general similarity functions. We also analyze how to effectively use unlabeled data for Clustering with non-interactive feedback. We discuss these in turn below.

1.1.2 Similarity Based Learning

Kernel functions have become an extremely popular tool in machine learning, with an attractive theory as well [133, 139, 187, 190, 203]. They are used in domains ranging from Computer Vision [132] to Computational Biology [187] to Language and Text Processing [139], with workshops, (e.g. [2, 3, 4, 5]), books [133, 139, 187, 190] [203], and large portions of major conferences (see, e.g., [1]) devoted to kernel methods. In this thesis, we strictly generalize and simplify the existing theory of Kernel Methods. Our approach brings a new perspective as well as a much simpler explanation for the effectiveness of kernel methods, which can help in the design of good kernel functions for new learning problems.

A kernel is a function that takes in two data objects (which could be images, DNA sequences, or points in R^n) and outputs a number, with the property that the function is symmetric and positive-semidefinite. That is, for any kernel K, there must exist an (implicit) mapping ϕ, such that for all inputs x, x' we have $K(x, x') = \phi(x) \cdot \phi(x')$. The kernel is then used inside a "kernelized" learning algorithm such as SVM or kernel-perceptron as the way in which the algorithm interacts with the data. Typical kernel functions for structured data include the polynomial kernel $K(x, x') = (1 + x \cdot x')^d$ and the Gaussian kernel $K(x, x') = e^{-||x-x'||^2/2\sigma^2}$, and a number of special-purpose kernels have been developed for sequence data, image data, and other types of data as well [88, 89, 157, 173, 193].

The theory behind kernel functions is based on the fact that many standard algorithms for learning linear separators, such as SVMs and the Perceptron algorithm, can be written so that the only way they interact with their data is via computing dot-products on pairs of examples. Thus, by replacing each invocation of $x \cdot x'$ with a kernel computation $K(x, x')$, the algorithm behaves exactly as if we had explicitly performed the mapping $\phi(x)$, even though ϕ may be a mapping into a very high-dimensional space (dimension n^d for the polynomial kernel) or even an infinite-dimensional space (as in the case of the Gaussian kernel). Furthermore, these algorithms have convergence rates that depend only on the *margin* of the best separator, and not on the dimension of the space in which the data resides [18, 191]. Thus, kernel functions are often viewed as providing much of the power of this implicit high-dimensional space, without paying for it computationally (because the ϕ mapping is only implicit) or in terms of sample size (if the data is indeed well-separated in that space).

While the above theory is quite elegant, it has a few limitations. First, when designing a kernel function for some learning problem, the intuition typically employed is that a good kernel would be one that serves as a good similarity function for the given problem [187]. On the other hand, the above theory talks about margins in an implicit and possibly very high-dimensional space. So, in this sense the theory is not that helpful for providing intuition when selecting or designing a kernel function. Second, it may be that the most natural similarity function for a given problem is not positive-semidefinite, and it could require substantial work, possibly reducing the quality of the function, to coerce it into a legal form. Finally, from a complexity-theoretic perspective, it is somewhat unsatisfying for the explanation of the effectiveness of some algorithm to depend on properties of an implicit high-dimensional mapping that one may not even be able to calculate. In particular, the standard theory at first blush has a "something for nothing" feel to it (all the power of the implicit high-dimensional space without having to pay for it) and perhaps there is a more prosaic explanation of what it is that makes a kernel useful for a given learning problem. For these reasons, it would be helpful to have a theory that involved more tangible quantities.

In this thesis we provide new theories that address these limitations in two ways. First, we show how Random Projection techniques can be used to convert a given kernel function into an explicit, distribution

dependent, set of features, which can then be fed into more general (not necessarily kernelizable) learning algorithms. Conceptually, this result suggests that designing a good kernel function is much like designing a good feature space. From a practical perspective it provides an alternative to "kernelizing" a learning algorithm: rather than modifying the algorithm to use kernels, one can instead construct a mapping into a low-dimensional space using the kernel and the data distribution, and then run an un-kernelized algorithm over examples drawn from the mapped distribution.

Second, we also show how such methods can be extended to more *general* pairwise similarity functions and also give a formal theory that matches the standard intuition that a good kernel function is one that acts as a good measure of similarity. In particular, we define a notion of what it means for a pairwise function $K(x, x')$ to be a "good similarity function" for a given learning problem that (a) does not require the notion of an implicit space and allows for functions that are not positive semi-definite, (b) is provably sufficient for learning, and (c) is broad, in sense that a good kernel in the standard sense (large margin in the implicit ϕ-space) will also satisfy our definition of a good similarity function, though with some loss in the parameters. This framework provides the first rigorous explanation for why a kernel function that is good in the large-margin sense can also formally be viewed as a good measure of similarity, thereby giving formal justification to a common intuition about kernels. We start by analyzing a first notion of a good similarity function in Section 3.3 and analyze its relationship with the usual notion of a good kernel function. We then present a slightly different and broader notion that we show it provides even better kernels to similarity translation. Any large-margin kernel function is a good similarity function under the new definition, and while we still incur some loss in the parameters, this loss is much smaller than under the prior definition, especially in terms of the final labeled sample-complexity bounds. In particular, when using a valid kernel function as a similarity function, a substantial portion of the previous sample-complexity bound can be transferred over to merely a need for *un*labeled examples. We also show our new notion is *strictly more general* than the notion of a large margin kernel. We discuss these results in Section 3.4. In Chapter 6 other random projection results for the case where K is in fact a valid kernel.

1.1.3 Clustering via Similarity Functions

Problems of clustering data from pairwise similarity information are ubiquitous in science [8, 19, 83, 91, 95, 138, 146, 147, 151, 205]. A typical example task is to cluster a set of emails or documents according to some criterion (say, by topic) by making use of a pairwise similarity measure among data objects. In this context, a natural example of a similarity measure for document clustering might be to consider the fraction of important words that two documents have in common.

While the study of clustering is centered around an intuitively compelling goal (and it has been a major tool in many different fields), it has been difficult to reason about it at a general level in part due to the lack of a theoretical framework along the lines we have for supervised classification.

In this thesis we develop the first general discriminative framework for Clustering, i.e. a framework for analyzing clustering accuracy without making strong probabilistic assumptions. In particular, we present a theoretical approach to the clustering problem that directly addresses the fundamental question of how good the similarity measure must be in terms of its relationship to the desired ground-truth clustering (e.g., clustering by topic) in order to allow an algorithm to cluster well. Very strong properties and assumptions are needed if the goal is to produce a single approximately-correct clustering; however, we show that if we relax the objective and allow the algorithm to produce a hierarchical clustering such that desired clustering is close to some *pruning* of this tree (which a user could navigate), then we can develop a general theory of natural properties that are sufficient for clustering via various kinds of algorithms. This framework is an analogue of the PAC learning model for clustering, where the natural object of study, rather than being a concept class, is instead a property of the similarity information with respect to the desired ground-truth

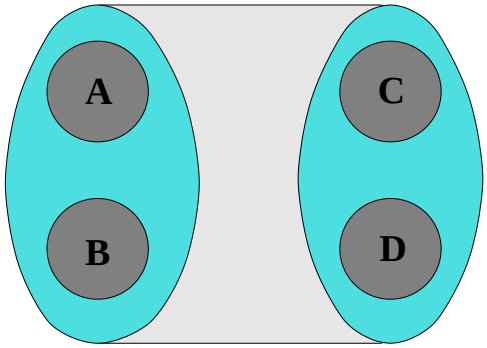

Figure 1.1: Data lies in four regions A, B, C, D (e.g., think of as documents on baseball, football, TCS, and AI). Suppose that $K(x, y) = 1$ if x and y belong to the same region, $K(x, y) = 1/2$ if $x \in A$ and $y \in B$ or if $x \in C$ and $y \in D$, and $K(x, y) = 0$ otherwise. Even assuming that all points are more similar to other points in their own cluster than to any point in any other cluster, there are still multiple consistent clusterings, including two consistent 3-clusterings (($A \cup B$, C, D) or (A, B, $C \cup D$)). However, there is a single hierarchical decomposition such that any consistent clustering is a pruning of this tree.

clustering.

As indicated above, the main difficulty that appears when phrasing the problem in this general way is that if one defines success as outputting *a single clustering* that closely approximates the correct clustering, then one needs to assume very strong conditions on the similarity function. For example, if the function provided by the domain expert is extremely good, say $K(x, y) > 1/2$ for all pairs x and y that should be in the same cluster, and $K(x, y) < 1/2$ for all pairs x and y that should be in different clusters, then we could just use it to recover the clusters in a trivial way. However, if we just slightly weaken this condition to simply require that all points x are more similar to all points y from their own cluster than to any points y from any other clusters, then this is no longer sufficient to uniquely identify even a good approximation to the correct answer. For instance, in the example in Figure 1.1, there are multiple clusterings consistent with this property (one with 1 cluster, one with 2 clusters, two with 3 clusters, and one with 4 clusters). Even if one is told the correct clustering has 3 clusters, there is no way for an algorithm to tell which of the two (very different) possible solutions is correct. In fact, results of Kleinberg [151] can be viewed as effectively ruling out a broad class of scale-invariant properties like this one as being sufficient for producing the correct answer.

In our work we overcome this problem by considering two relaxations of the clustering objective that are natural for many clustering applications. The first is to allow the algorithm to produce a small *list* of clusterings such that at least one of them has low error[2]. The second is (as mentioned above) to allow the clustering algorithm to produce a *tree* (a hierarchical clustering) such that the correct answer is approximately some pruning of this tree. For instance, the example in Figure 1.1 has a natural hierarchical decomposition of this form. Both relaxed objectives make sense for settings in which we imagine the output being fed to a user who will then decide what she likes best. For example, with the tree relaxation, we allow the clustering algorithm to effectively say: "I wasn't sure how specific you wanted to be, so if any of these clusters are too broad, just click and I will split it for you." We then show that with these relaxations, a number of interesting, natural learning-theoretic and game-theoretic properties can be defined that each are sufficient to allow an algorithm to cluster well.

For concreteness, we shall summarize in the following our main results. First, we consider a family

[2]So, this is similar in spirit to list-decoding in coding theory.

of stability-based properties, showing that a natural generalization of the "stable marriage" property is sufficient to produce a hierarchical clustering. (The property is that no two subsets $A \subset C$, $A' \subset C'$ of clusters $C \neq C'$ in the correct clustering are both more similar on average to each other than to the rest of their own clusters.) Moreover, a significantly weaker notion of stability (which we call "stability of large subsets") is also sufficient to produce a hierarchical clustering, but requires a more involved algorithm. We also show that a weaker "average-attraction" property (which is provably not enough to produce a single correct hierarchical clustering) is sufficient to produce a small list of clusterings, and give generalizations to even weaker conditions that are related to the notion of large-margin kernel functions. We develop a notion of the *clustering complexity* of a given property (the minimum possible list length that can be guaranteed by any algorithm) and provide both upper and lower bounds for the properties we consider. This notion is analogous to notions of capacity in classification [72, 103, 203] and it provides a formal measure of the inherent usefulness of a given property. We show that properties implicitly assumed by approximation algorithms for standard graph-based objective functions can be viewed as special cases of some of the properties considered above.

We also show how our algorithms can be extended to the inductive case, i.e., by using just a constant-sized sample, as in property testing. While most of our algorithms extend in a natural way, for certain properties their analysis requires more involved arguments using regularity-type results of [14, 113].

More generally, our framework provides a formal way to analyze what properties of a similarity function would be sufficient to produce low-error clusterings, as well as what algorithms are suited for a given property. For some of our properties we are able to show that known algorithms succeed (e.g. variations of bottom-up hierarchical linkage based algorithms). However, for the most general ones, e.g., the stability of large subsets property, we need new algorithms that are able to take advantage of them. In fact, the algorithm we develop for the stability of the large subsets property combines learning-theoretic approaches used in Chapter 3 (and described in Section 1.1.2) with linkage-style methods. We describe these results in Chapter 4.

1.1.4 Mechanism Design, Machine Learning, and Pricing Problems

In this thesis we also present explicit connections between Machine Learning Theory and certain contemporary problems in Economics.

With the Internet developing as the single most important arena for resource sharing among parties with diverse and selfish interests, traditional algorithmic and distributed systems need to be combined with the understanding of game-theoretic and economic issues [177]. A fundamental research endeavor in this new field is the design and analysis of auction mechanisms and pricing algorithm [70, 121, 124, 129, 129]. In this thesis we show how machine learning methods can be used in the design of auctions and other pricing mechanisms with guarantees on their performance.

In particular, we show how sample complexity techniques from statistical learning theory can be used to reduce problems of incentive-compatible mechanism design to standard algorithmic questions, for a wide range of revenue-maximizing problems in an unlimited supply setting. In doing so, we obtain a unified approach for considering a variety of profit maximizing mechanism design problems, including many that have been previously considered in the literature. We show how techniques from in machine learning theory can be used both for analyzing and designing our mechanisms. We apply our reductions to a diverse set of revenue maximizing pricing problems, such as the problem of auctioning a digital good, the attribute auction problem, and the problem of item pricing in unlimited supply combinatorial auctions.

For concreteness, in the following paragraphs, we shall give more details on the setting we study in our work. Consider a seller with multiple digital goods or services for sale, such as movies, software, or network services, over which buyers may have complicated preferences. In order to sell these items

through an incentive-compatible auction mechanism, this mechanism should have the property that each bidder is offered a set of prices that do not depend on the value of her bid. The problem of designing a revenue-maximizing auction is known in the economics literature as the optimal auction design problem [171]. The classical model for optimal auction design assumes a Bayesian setting in which players' valuations (types) are drawn from some probability distribution that furthermore is known to the mechanism designer. For example, to sell a single item of fixed marginal cost, one should set the price that maximizes the profit margin per sale times the probability a random person would be willing to buy at that price. However, in complex or non-static environments, these assumptions become unrealistic. In these settings, machine learning can provide a natural approach to the design of near-optimal mechanisms without such strong assumptions or degree of prior knowledge.

Specifically, notice that while a truthful auction mechanism should have the property that the prices offered to some bidder i do not depend on the value of her bid, they can depend on the amounts bid by other bidders j. From a machine learning perspective, this is very similar to thinking of bidders as "examples" and our objective being to use information from examples $j \neq i$ to produce a good prediction with respect to example i. Thus, without presuming a known distribution over bidders (or even that bidders come from any distribution at all) perhaps if the number of bidders is sufficiently large, enough information can be learned from some of them to perform well on the rest. In this thesis we formalize this idea and show indeed that sample-complexity techniques from machine learning theory [18, 203] can be adapted to this setting to give quantitative bounds for this kind of approach. More generally, we show that sample complexity analysis can be applied to convert incentive-compatible mechanism design problems to more standard algorithm-design questions, in a wide variety of revenue-maximizing auction settings.

Our reductions imply that for these problems, given an algorithm for the *non* incentive-compatible pricing problem, we can convert it into an algorithm for the incentive-compatible mechanism design problem that is only a factor of $(1+\epsilon)$ worse, as long as the number of bidders is sufficiently large as a function of an appropriate measure of complexity of the class of allowable pricing functions. We apply these results to the problem of auctioning a digital good, to the attribute auction problem which includes a wide variety of discriminatory pricing problems, and to the problem of item-pricing in unlimited-supply combinatorial auctions.From a machine learning perspective, these settings present several challenges: in particular, the *loss function* is discontinuous, is asymmetric, and has a large range.

The high level idea of our most basic reduction is based on the notion of a random sampling auction. For concreteness, let us imagine we are selling a collection of n goods or services of zero marginal cost to us, to n bidders who may have complex preference functions over these items, and our objective is to achieve revenue comparable to the best possible assignment of prices to the various items we are selling. So, technically speaking, we are in the setting of maximizing revenue in an unlimited supply combinatorial auction. Then given a set of bids S, we perform the following operations. We first randomly partition S into two sets S_1 and S_2. We then consider the purely algorithmic problem of finding the best set of prices p_1 for the set of bids S_1 (which may be difficult but is purely algorithmic), and the best set of prices p_2 for the set of bids S_2. We then use p_1 as offer prices for bidders in S_2, giving each bidder the bundle maximizing revealed valuation minus price, and use p_2 as offer prices for bidders in S_1. We then show that even if bidders' preferences are extremely complicated, this mechanism will achieve revenue close to that of the best fixed assignment of prices to items so long as the number of bidders is sufficiently large compared to the number of items for sale. For example, if all bidders' valuations on the grand bundle of all n items lie in the range $[1, h]$, then $O(hn/\epsilon^2)$ bidders are sufficient so that with high probability, we come within a $(1 + \epsilon)$ factor of the optimal fixed item pricing. Or, if we cannot solve the algorithmic problem exactly (since many problems of this form are often NP-hard [25, 26, 32, 129]), we lose only a $(1 + \epsilon)$ factor over whatever approximation our method for solving the algorithmic problem gives us.

More generally, these methods apply to a wide variety of pricing problems, including those in which bidders have both public and private information, and also give a formal framework in which one can address other interesting design issues such as how fine-grained a market segmentation should be. This framework provides a unified approach to considering a variety of profit maximizing mechanism design problems including many that have been previously considered in the literature. Furthermore, our results substantially generalize the previous work on random sampling mechanisms by both broadening the applicability of such mechanisms and by simplifying the analysis.

Some of our techniques give suggestions for the design of mechanisms and others for their analysis. In terms of design, these include the use of discretization to produce smaller function classes, and the use of structural-risk minimization to choose an appropriate level of complexity of the mechanism for a given set of bidders. In terms of analysis, these include both the use of basic sample-complexity arguments, and the notion of multiplicative covers for better bounding the true complexity of a given class of offers.

Finally, from a learning perspective, this mechanism-design setting presents a number of technical challenges when attempting to get good bounds: in particular, the payoff function is discontinuous and asymmetric, and the payoffs for different offers are non-uniform. For example, we develop bounds based on a different notion of covering number than typically used in machine learning, in order to obtain results that are more meaningful for this mechanism design setting. We describe these results in Chapter 7.

1.2 Summary of the Main Results and Bibliographic Information

This thesis is organized as follows.

- In Chapter 2 we present the first general discriminative model for Semi-Supervised learning. In this model we provide a variety of algorithmic and sample complexity results and we also show how it can be used to reason about many of the different semi-supervised learning approaches taken over the past decade in the machine learning community. Much of this chapter is based on work that appears in [23], [27]. Other related work we have done on Co-training (which we briefly mention) appears in [28].
- In Chapter 3 we provide a theory of learning with general similarity functions (that is, functions which are not necessarily legal kernels). This theory provides conditions on the suitability of a similarity function for a given learning problem in terms of more tangible and more operational quantities than those used by the standard theory of kernel functions. In addition to being provably more general than the standard theory, our framework provides the first rigorous explanation for why a kernel function that is good in the large-margin sense can also formally be viewed as a good measure of similarity, thereby giving formal justification to a common intuition about kernels. In this chapter we analyze both algorithmic and sample complexity issues, and this is mostly based on work that appears in [24], [38], and [39].
- In Chapter 4 we study Clustering and we present the first general framework for analyzing clustering accuracy without probabilistic assumptions. Again, in this chapter we consider both algorithmic and information theoretic aspects. This is mainly based on work that appears in [40], but also includes parts from the recent work in [42].
- In Chapter 5 we analyze Active Learning and present two main results. In Section 5.1, we provide a generic active learning algorithm that works in the presence of arbitrary forms of noise. This section is focused mostly on sample complexity aspects and the main contribution here is to provide the first positive result showing that active learning can provide a significant improvement over passive learning even in the presence of arbitrary forms of noise. In Section 5.2 we analyze a natural

margin-based active learning strategy for learning linear separators (which queries points near the hypothesized decision boundary). We provide a detailed analysis (both sample complexity and algorithmic) both in the realizable case and in a specific noisy setting related to the Tsybakov noise condition. This chapter is based on work that appears in [30], [35], and [33]. We also briefly mention other recent work on the topic [41].

- In Chapter 6 we present additional results on learning with kernel functions. Specifically, we show how Random Projection techniques can be used to "demystify" kernel functions. We show that in the presence of a large margin, a kernel can be efficiently converted into a mapping to a low dimensional space; in particular, we present a computationally efficient procedure that, given black-box access to the kernel and unlabeled data, generates a small number of features that approximately preserve both separability and margin. This is mainly based on work that appears in [31].
- In Chapter 7 we show how model selection and sample complexity techniques in machine learning can be used to convert difficult mechanism design problems to more standard algorithmic questions for a wide range of pricing problems. We present a unified approach for considering a variety of profit maximizing mechanism design problems, such as the problem of auctioning a digital good, the attribute auction problem (which includes many discriminatory pricing problems), and the problem of item pricing in unlimited supply combinatorial auctions. These results substantially generalize the previous work on random sampling mechanisms by both broadening the applicability of such mechanisms (e.g., to multi-parameter settings), and by simplifying and refining the analysis. This chapter is mainly based on work that appears in [29] and [36] and it is focused on using machine learning techniques for providing a generic reduction from the incentive-compatible mechanism design question to more standard algorithmic questions, without also attempting to address the algorithmic questions as well. In other related work (which for coherence and space limitations is not included in this thesis) we have also considered various algorithmic problems that arise in this context [26], [25], [32] and [37].

While we discuss both technical and conceptual connections between the various learning protocols and paradigms studied throughout the thesis, each chapter can also be read somewhat independently.

Chapter 2

A Discriminative Framework for Semi-Supervised Learning

There has recently been substantial interest in *semi-supervised* learning — a paradigm for incorporating unlabeled data in the learning process — since any useful information that reduces the amount of labeled data needed for learning can be a significant benefit. Several techniques have been developed for doing this, along with experimental results on a variety of different learning problems. Unfortunately, the standard learning frameworks for reasoning about supervised learning do not capture the key aspects and the assumptions underlying these *semi*-supervised learning methods.

In this chapter we describe an augmented version of the PAC model designed for semi-supervised learning, that can be used to reason about many of the different approaches taken over the past decade in the Machine Learning community. This model provides a unified framework for analyzing when and why unlabeled data can help in the semi-supervised learning setting, in which one can analyze both sample-complexity and algorithmic issues. The model can be viewed as an extension of the standard PAC model where, in addition to a concept class C, one also proposes a compatibility notion: a type of compatibility that one believes the target concept should have with the underlying distribution of data. Unlabeled data is then potentially helpful in this setting because it allows one to estimate compatibility over the space of hypotheses, and to reduce the size of the search space from the whole set of hypotheses C down to those that, according to one's assumptions, are a-priori reasonable with respect to the distribution. As we show, many of the assumptions underlying existing semi-supervised learning algorithms can be formulated in this framework.

After proposing the model, we then analyze sample-complexity issues in this setting: that is, how much of each type of data one should expect to need in order to learn well, and what the key quantities are that these numbers depend on. Our work is the first to address such important questions in the context of semi-supervised learning in a unified way. We also consider the algorithmic question of how to efficiently optimize for natural classes and compatibility notions, and provide several algorithmic results including an improved bound for Co-Training with linear separators when the distribution satisfies independence given the label.

2.1 Introduction

As mentioned in Chapter 1, given the easy availability of unlabeled data in many settings, there has been growing interest in methods that try to use such data together with the (more expensive) labeled data for learning. In particular, a number of semi-supervised learning techniques have been developed for

doing this, along with experimental results on a variety of different learning problems. These include label propagation for word-sense disambiguation [210], co-training for classifying web pages [62] and improving visual detectors [159], transductive SVM [141] and EM [176] for text classification, graph-based methods [215], and others. The problem of learning from labeled and unlabeled data has been the topic of several ICML workshops [15, 117] as well as a recent book [82] and survey article [214].

What makes unlabeled data so useful and what many of these methods exploit, is that for a wide variety of learning problems, the natural regularities of the problem involve not only the *form* of the function being learned by also how this function *relates* to the distribution of data. For example, in many problems one might expect the target function should cut through low density regions of the space, a property used by the transductive SVM algorithm [141]. In other problems one might expect the target to be self-consistent in some way, a property used by Co-training [62]. Unlabeled data is potentially useful in these settings because it then allows one to reduce the search space to a set which is a-priori reasonable with respect to the underlying distribution.

Unfortunately, however, the underlying assumptions of these semi-supervised learning methods are not captured well by standard theoretical models. The main goal of this chapter is to propose a *unified theoretical framework* for semi-supervised learning, in which one can analyze when and why unlabeled data can help, and in which one can discuss both sample-complexity and algorithmic issues in a discriminative (PAC-model style) framework.

One difficulty from a theoretical point of view is that standard discriminative learning models do not allow one to specify relations that one believes the target should have with the underlying distribution. In particular, both in the PAC model [69, 149, 201] and the Statistical Learning Theory framework [203] there is purposefully a complete disconnect between the data distribution D and the target function f being learned. The only prior belief is that f belongs to some class C: even if the data distribution D is known fully, any function $f \in C$ is still possible. For instance, in the PAC model, it is perfectly natural (and common) to talk about the problem of learning a concept class such as DNF formulas [162, 206] or an intersection of halfspaces [47, 61, 153, 204] over the uniform distribution; but clearly in this case unlabeled data is useless — you can just generate it yourself. For learning over an unknown distribution, unlabeled data can help somewhat in the standard models (e.g., by allowing one to use distribution-specific algorithms and sample-complexity bounds [53, 144]), but this does not seem to capture the power of unlabeled data in practical semi-supervised learning methods.

In *generative* models, one *can* easily talk theoretically about the use of unlabeled data, e.g., [76, 77]. However, these results typically make strong assumptions that essentially imply that there is only one natural distinction to be made for a given (unlabeled) data distribution. For instance, a typical generative model would be that we assume positive examples are generated by one Gaussian, and negative examples are generated by another Gaussian. In this case, given enough unlabeled data, we could in principle recover the Gaussians and would need labeled data only to tell us which Gaussian is the positive one and which is the negative one.[1] However, this is too strong an assumption for most real-world settings. Instead, we would like our model to allow for a distribution over data (e.g., documents we want to classify) where there are a number of plausible distinctions we might want to make. In addition, we would like a general framework that can be used to model many different uses of unlabeled data.

2.1.1 Our Contribution

In this chapter, we present a discriminative (PAC-style framework) that bridges between these positions and can be used to help think about and analyze many of the ways unlabeled data is typically used. This

[1] [76, 77] do not assume Gaussians in particular, but they do assume the distributions are distinguishable, which from this perspective has the same issue.

framework extends the PAC learning model in a way that allows one to express not only the form of target function one is considering, but also relationships that one hopes the target function and underlying distribution will possess. We then analyze both sample-complexity issues—that is, how much of each type of data one should expect to need in order to learn well—as well as algorithmic results in this model. We derive bounds for both the realizable (PAC) and agnostic (statistical learning framework) settings.

Specifically, the idea of the proposed model is to augment the PAC notion of a *concept class*, which is a set of functions (such as linear separators or decision trees), with a notion of *compatibility* between a function and the data distribution that we hope the target function will satisfy. Rather than talking of "learning a concept class C," we will talk of "learning a concept class C under compatibility notion χ." For example, suppose we believe there should exist a low-error linear separator, and that furthermore, if the data happens to cluster, then this separator does not slice through the middle of any such clusters. Then we would want a compatibility notion that penalizes functions that do, in fact, slice through clusters. In this framework, the ability of unlabeled data to help depends on two quantities: first, the extent to which the target function indeed satisfies the given assumptions, and second, the extent to which the distribution allows this assumption to rule out alternative hypotheses. For instance, if the data does not cluster at all (say the underlying distribution is uniform in a ball), then all functions would equally satisfy this compatibility notion and the assumption is not useful. From a Bayesian perspective, one can think of this as a PAC model for a setting in which one's prior is not just over functions, but also over how the function and underlying distribution relate to each other.

To make our model formal, we will need to ensure that the degree of compatibility be something that can be *estimated from a finite sample*. To do this, we will require that the compatibility notion χ in fact be a function from $C \times X$ to $[0, 1]$, where the compatibility of a hypothesis h with the data distribution D is then $\mathbf{E}_{x\sim D}[\chi(h, x)]$. That is, we require that the degree of *in*compatibility be a kind of unlabeled loss function, and the incompatibility of a hypothesis h with a data distribution D is a quantity we can think of as an "unlabeled error rate" that measures how a-priori unreasonable we believe some proposed hypothesis to be. For instance, in the example above of a "margin-style" compatibility, we could define $\chi(f, x)$ to be an increasing function of the distance of x to the separator f. In this case, the unlabeled error rate, $1 - \chi(f, D)$, is a measure of the probability mass close to the proposed separator. In co-training, where each example x has two "views" ($x = \langle x_1, x_2\rangle$), the underlying belief is that the true target c^* can be decomposed into functions $\langle c_1^*, c_2^*\rangle$ over each view such that for most examples, $c_1^*(x_1) = c_2^*(x_2)$. In this case, we can define $\chi(\langle f_1, f_2\rangle, \langle x_1, x_2\rangle) = 1$ if $f_1(x_1) = f_2(x_2)$, and 0 if $f_1(x_1) \neq f_2(x_2)$. Then the compatibility of a hypothesis $\langle f_1, f_2\rangle$ with an underlying distribution D is $\Pr_{\langle x_1,x_2\rangle\sim D}[f_1(x_1) = f_2(x_2)]$.

This framework allows us to analyze the ability of a finite unlabeled sample to reduce our dependence on labeled examples, as a function of (1) the compatibility of the target function (i.e., how correct we were in our assumption) and (2) various measures of the "helpfulness" of the distribution. In particular, in our model, we find that unlabeled data can help in several distinct ways.

- If the target function is highly compatible with D and belongs to C, then if we have enough unlabeled data to estimate compatibility over all $f \in C$, we can in principle reduce the size of the search space from C down to just those $f \in C$ whose estimated compatibility is high. For instance, if D is "helpful", then the set of such functions will be much smaller than the entire set C. In the agnostic case we can do (unlabeled)-data-dependent structural risk minimization to trade off labeled error and incompatibility.
- By providing an estimate of D, unlabeled data can allow us to use a more refined distribution-specific notion of "hypothesis space size" such as Annealed VC-entropy [103], Rademacher complexities [43, 72, 155] or the size of the smallest ϵ-cover [53], rather than VC-dimension [69, 149]. In fact, for many natural notions of compatibility we find that the sense in which unlabeled data

reduces the "size" of the search space is best described in these distribution-specific measures.

- Finally, if the distribution is especially helpful, we may find that not only does the set of compatible $f \in C$ have a small ϵ-cover, but also the elements of the cover are far apart. In that case, if we assume the target function is fully compatible, we may be able to learn from even fewer labeled examples than the $\Omega(1/\epsilon)$ needed just to *verify* a good hypothesis. For instance, as one application of this, we show that under the assumption of independence given the label, one can efficiently perform Co-Training of linear separators from a single labeled example!

Our framework also allows us to address the issue of how much *unlabeled* data we should expect to need. Roughly, the "VCdim$/\epsilon^2$" form of standard sample complexity bounds now becomes a bound on the number of *unlabeled* examples we need to uniformly estimate compatibilities. However, technically, the set whose VC-dimension we now care about is not C but rather a set defined by both C and χ: that is, the overall complexity depends both on the complexity of C and the complexity of the notion of compatibility (see Section 2.3.1). One consequence of our model is that if the target function and data distribution are both well behaved with respect to the compatibility notion, then the sample-size bounds we get for labeled data can substantially beat what one could hope to achieve through pure labeled-data bounds, and we illustrate this with a number of examples through the chapter.

2.1.2 Summary of Main Results

The primary contributions of this chapter are the following. First, as described above, we develop a new discriminative (PAC-style) model for semi-supervised learning, that can be used to analyze when unlabeled data can help and how *much* unlabeled data is needed in order to gain its benefits, as well as the algorithmic problems involved. Second, we present a number of sample-complexity bounds in this framework, both in terms of uniform-convergence results—which apply to any algorithm that is able to find rules of low error and high compatibility—as well as ϵ-cover-based bounds that apply to a more restricted class of algorithms but can be substantially tighter. For instance, we describe several natural cases in which ϵ-cover-based bounds can apply even though with high probability there still exist bad hypotheses in the class consistent with the labeled and unlabeled examples. Finally, we present several PAC-style algorithmic results in this model. Our main algorithmic result is a new algorithm for Co-Training with linear separators that, if the distribution satisfies independence given the label, requires only a single labeled example to learn to any desired error rate ϵ *and* is computationally efficient (i.e., achieves PAC guarantees). This substantially improves on the results of [62] which required enough labeled examples to produce an initial weak hypothesis, and in the process we get a simplification to the noisy halfspace learning algorithm of [64].

Our framework has helped analyze many of the existing semi-supervised learning methods used in practice and has guided the development of new semi-supervised learning algorithms and analyses. We discuss this further in Section 2.6.1.

2.1.3 Structure of this Chapter

We begin by describing the general setting in which our results apply as well as several examples to illustrate our framework in Section 2.2. We then give results both for *sample complexity* (in principle, how much data is needed to learn) and *efficient algorithms*. In terms of sample-complexity, we start by discussing uniform convergence results in Section 2.3.1. For clarity we begin with the case of finite hypothesis spaces in Section 2.3.1, and then discuss infinite hypothesis spaces in Section 2.3.1. These results give bounds on the number of examples needed for any learning algorithm that produces a compatible

hypothesis of low empirical error. We also show how in the agnostic case we can do (unlabeled)-data-dependent structural risk minimization to trade off labeled error and incompatibility in Section 2.3.1. To achieve tighter bounds, in Section 2.3.2 we give results based on the notion of ϵ-cover size. These bounds hold only for algorithms of a specific type (that first use the unlabeled data to choose a small set of "representative" hypotheses and then choose among the representatives based on the labeled data), but can yield bounds substantially better than with uniform convergence (e.g., we can learn even though there exist bad $h \in C$ consistent with the labeled and unlabeled examples).

In Section 2.4, we give our algorithmic results. We begin with a particularly simple class C and compatibility notion χ for illustration, and then give our main algorithmic result for Co-Training with linear separators. In Section 2.5 we discuss a transductive analog of our model, connections with generative models and other ways of using unlabeled data in machine learning, as well as the relationship between our model and the Luckiness Framework [191] developed in the context of supervised learning. Finally, in Section 2.6 we discuss some implications of our model and present our conclusions, as well a number of open problems.

2.2 A Formal Framework

In this section we introduce general notation and terminology we use throughout the chapter, and describe our model for semi-supervised learning. In particular, we formally define what we mean by a *notion of compatibility* and we illustrate it through a number of examples including margins and co-training.

We will focus on binary classification problems. We assume that our data comes according to a fixed unknown distribution D over an instance space X, and is labeled by some unknown target function $c^* : X \to \{0, 1\}$. A learning algorithm is given a set S_L of labeled examples drawn i.i.d. from D and labeled by c^* as well as a (usually larger) set S_U of unlabeled examples from D. The goal is to perform some optimization over the samples S_L and S_U and to output a hypothesis that agrees with the target over most of the distribution. In particular, the error rate (also called "0-1 loss") of a given hypothesis f is defined as $err(f) = err_D(f) = \Pr_{x\sim D}[f(x) \neq c^*(x)]$. For any two hypotheses f_1, f_2, the distance with respect to D between f_1 and f_2 is defined as $d(f_1, f_2) = d_D(f_1, f_2) = \Pr_{x\sim D}[f_1(x) \neq f_2(x)]$. We will use $\widehat{err}(f)$ to denote the empirical error rate of f on a given labeled sample (i.e., the fraction of mistakes on the sample) and $\hat{d}(f_1, f_2)$ to denote the empirical distance between f_1 and f_2 on a given unlabeled sample (the fraction of the sample on which they disagree). As in the standard PAC model, a *concept class* or *hypothesis space* is a set of functions over the instance space X. In the "realizable case", we make the assumption that the target is in a given class C, whereas in the "agnostic case" we do not make this assumption and instead aim to compete with the best function in the given class C.

We now formally describe what we mean by a notion of compatibility. A *notion of compatibility* is a mapping from a hypothesis f and a distribution D to $[0, 1]$ indicating how "compatible" f is with D. In order for this to be estimable from a finite sample, we require that compatibility be an expectation over individual examples.[2] Specifically, we define:

Definition 2.2.1 *A* legal notion of compatibility *is a function $\chi : C \times X \to [0, 1]$ where we (overloading notation) define $\chi(f, D) = \mathbf{E}_{x\sim D}[\chi(f, x)]$. Given a sample S, we define $\chi(f, S)$ to be the empirical average of χ over the sample.*

[2]One could imagine more general notions of compatibility with the property that they can be estimated from a finite sample and all our results would go through in that case as well. We consider the special case where the compatibility is an expectation over individual examples for simplicity of notation, and because most existing semi-supervised learning algorithms used in practice do satisfy it.

Note 1 *One could also allow compatibility functions over k-tuples of examples, in which case our (unlabeled) sample-complexity bounds would simply increase by a factor of k. For settings in which D is actually known in advance (e.g., transductive learning, see Section 2.5.1) we can drop this requirement entirely and allow any notion of compatibility $\chi(f, D)$ to be legal.*

Definition 2.2.2 *Given compatibility notion χ, the* incompatibility *of f with D is $1 - \chi(f, D)$. We will also call this its* unlabeled error rate*, $err_{unl}(f)$, when χ and D are clear from context. For a given sample S, we use $\widehat{err}_{unl}(f) = 1 - \chi(f, S)$ to denote the empirical average over S.*

Finally, we need a notation for the set of functions whose incompatibility (or unlabeled error rate) is at most some given value τ.

Definition 2.2.3 *Given value τ, we define $C_{D,\chi}(\tau) = \{f \in C : err_{unl}(f) \leq \tau\}$. So, e.g., $C_{D,\chi}(1) = C$. Similarly, for a sample S, we define $C_{S,\chi}(\tau) = \{f \in C : \widehat{err}_{unl}(f) \leq \tau\}$*

We now give several examples to illustrate this framework:

Example 1**.** Suppose examples are points in R^d and C is the class of linear separators. A natural belief in this setting is that data should be "well-separated": not only should the target function separate the positive and negative examples, but it should do so by some reasonable *margin* γ. This is the assumption used by Transductive SVM, also called Semi-Supervised SVM (S^3VM) [55, 81, 141]. In this case, if we are given γ up front, we could define $\chi(f, x) = 1$ if x is farther than distance γ from the hyperplane defined by f, and $\chi(f, x) = 0$ otherwise. So, the incompatibility of f with D is the *probability mass within distance* γ of the hyperplane $f \cdot x = 0$. Alternatively, if we do not want to commit to a specific γ in advance, we could define $\chi(f, x)$ to be a smooth function of the distance of x to the separator, as done in [81]. Note that in contrast, defining compatibility of a hypothesis based on the largest γ such that D has probability mass *exactly zero* within distance γ of the separator would *not* fit our model: it cannot be written as an expectation over individual examples and indeed would not be a good definition since one cannot distinguish "zero" from "exponentially close to zero" from a small sample of unlabeled data.

Example 2**.** In co-training [62], we assume examples x each contain two "views": $x = \langle x_1, x_2 \rangle$, and our goal is to learn a pair of functions $\langle f_1, f_2 \rangle$, one on each view. For instance, if our goal is to classify web pages, we might use x_1 to represent the words on the page itself and x_2 to represent the words attached to links pointing *to* this page from other pages. The hope underlying co-training is that the two parts of the example are generally consistent, which then allows the algorithm to bootstrap from unlabeled data. For example, *iterative co-training* uses a small amount of labeled data to learn some initial information (e.g., if a link with the words "my advisor" points to a page then that page is probably a faculty member's home page). Then, when it finds an unlabeled example where one side is confident (e.g., the link says "my advisor"), it uses that to label the example for training over the other view. In *regularized co-training*, one attempts to directly optimize a weighted combination of accuracy on labeled data and agreement over unlabeled data. These approaches have been used for a variety of learning problems, including named entity classification [87], text classification [116, 175], natural language processing [182], large scale document classification [180], and visual detectors [159]. As mentioned in Section 2.1, the assumptions underlying this method fit naturally into our framework. In particular, we can define the incompatibility of some hypothesis $\langle f_1, f_2 \rangle$ with distribution D as $\Pr_{\langle x_1, x_2 \rangle \sim D}[f_1(x_1) \neq f_2(x_2)]$. Similar notions are given in subsequent work of [184, 196] for other types of learning problems (e.g. regression) and for other loss functions.

Example 3**.** In transductive graph-based methods, we are given a set of unlabeled examples connected in a graph g, where the interpretation of an edge is that we believe the two endpoints of the edge should have the *same* label. Given a few labeled vertices, various graph-based methods then attempt to use them to infer labels for the remaining points. If we are willing to view D as a distribution over *edges*

(a uniform distribution if g is unweighted), then as in co-training we can define the incompatibility of some hypothesis f as the probability mass of edges that are cut by f, which then motivates various cut-based algorithms. For instance, if we require f to be boolean, then the mincut method of [59] finds the most-compatible hypothesis consistent with the labeled data; if we allow f to be fractional and define $1 - \chi(f, \langle x_1, x_2 \rangle) = (f(x_1) - f(x_2))^2$, then the algorithm of [215] finds the most-compatible consistent hypothesis. If we do not wish to view D as a distribution over edges, we could have D be a distribution over *vertices* and broaden Definition 2.2.1 to allow for χ to be a function over *pairs* of examples. In fact, as mentioned in Note 1, since we have perfect knowledge of D in this setting we can allow any compatibility function $\chi(f, D)$ to be legal. We discuss more connections with graph-based methods in Section 2.5.1.

Example 4**.** As a special case of co-training, suppose examples are pairs of points in R^d, C is the class of linear separators, and we believe the two points in each pair should both be on the *same* side of the target function. (So, this is a version of co-training where we require $f_1 = f_2$.) The motivation is that we want to use pairwise information as in Example 3, but we also want to use the features of each data point. For instance, in the word-sense disambiguation problem studied by [210], the goal is to determine which of several dictionary definitions is intended for some target word in a piece of text (e.g., is "plant" being used to indicate a tree or a factory?). The local context around each word can be viewed as placing it into R^d, but the edges correspond to a completely different type of information: the belief that if a word appears twice in the same document, it is probably being used in the *same* sense both times. In this setting, we could use the same compatibility function as in Example 3, but rather than having the concept class C be all possible functions, we restrict C to just linear separators.

Example 5**.** In a related setting to co-training considered by [158], examples are single points in X but we have a pair of hypothesis spaces $\langle C_1, C_2 \rangle$ (or more generally a k-tuple $\langle C_1, \ldots, C_k \rangle$), and the goal is to find a pair of hypotheses $\langle f_1, f_2 \rangle \in C_1 \times C_2$ with low error over labeled data and that agree over the distribution. For instance, if data is sufficiently "well-separated", one might expect there to exist both a good linear separator and a good decision tree, and one would like to use this assumption to reduce the need for labeled data. In this case one could define compatibility of $\langle f_1, f_2 \rangle$ with D as $\Pr_{x \sim D}[f_1(x) = f_2(x)]$, or the similar notions given in [158, 189].

2.3 Sample Complexity Results

We now present several sample-complexity bounds that can be derived in this framework, showing how unlabeled data, together with a suitable compatibility notion, can reduce the need for labeled examples. We do not focus on giving the tightest possible bounds, but instead on the types of bounds and the quantities on which they depend, in order to better understand what it is about the learning *problem* one can hope to leverage from with unlabeled data.

The high-level structure of all of these results is as follows. First, given enough unlabeled data (where "enough" will be a function of some measure of the complexity of C and possibly of χ as well), we can uniformly estimate the true compatibilities of all functions in C using their empirical compatibilities over the sample. Then, by using this quantity to give a preference ordering over the functions in C, in the realizable case we can reduce "C" down to "the set of functions in C whose compatibility is not much larger than the true target function" in bounds for the number of *labeled* examples needed for learning. In the agnostic case we can do (unlabeled)-data-dependent structural risk minimization to trade off labeled error and incompatibility. The specific bounds differ in terms of the exact complexity measures used (and a few other issues) and we provide examples illustrating when and how certain complexity measures can be significantly more powerful than others. Moreover, one can prove fallback properties of these procedures — the number of labeled examples required is never much worse than the number of labeled examples

required by a standard supervised learning algorithm. However, if the assumptions happen to be right, one can significantly benefit by using the unlabeled data.

2.3.1 Uniform Convergence Bounds

We begin with uniform convergence bounds (later in Section 2.3.2 we give tighter ϵ-cover bounds that apply to algorithms of a particular form). For clarity, we begin with the case of finite hypothesis spaces where we measure the "size" of a set of functions by just the number of functions in the set. We then discuss several issues that arise when considering infinite hypothesis spaces, such as what is an appropriate measure for the "size" of the set of compatible functions, and the need to account for the complexity of the compatibility notion itself. Note that in the standard PAC model, one typically talks of either the realizable case, where we assume that the target function c^* belongs to C, or the agnostic case where we allow any target function c^* [149]. In our setting, we have the additional issue of *unlabeled* error rate, and can either make an a-priori assumption that the target function's unlabeled error is low, or else provide a bound in which our sample size (or error rate) depends on whatever its unlabeled error happens to be. We begin in Sections 2.3.1 and 2.3.1 with bounds for the the setting in which we assume $c^* \in C$, and then in Section 2.3.1 we consider the agnostic case where we remove this assumption.

Finite hypothesis spaces

We first give a bound for the "doubly realizable" case where we assume $c^* \in C$ and $err_{unl}(c^*) = 0$.

Theorem 2.3.1 *If $c^* \in C$ and $err_{unl}(c^*) = 0$, then m_u unlabeled examples and m_l labeled examples are sufficient to learn to error ϵ with probability $1 - \delta$, where*

$$m_u = \frac{1}{\epsilon}\left[\ln|C| + \ln\frac{2}{\delta}\right] \quad \textit{and} \quad m_l = \frac{1}{\epsilon}\left[\ln|C_{D,\chi}(\epsilon)| + \ln\frac{2}{\delta}\right].$$

In particular, with probability at least $1 - \delta$, all $f \in C$ with $\widehat{err}(f) = 0$ and $\widehat{err}_{unl}(f) = 0$ have $err(f) \leq \epsilon$.

Proof: The probability that a given hypothesis f with $err_{unl}(f) > \epsilon$ has $\widehat{err}_{unl}(f) = 0$ is at most $(1-\epsilon)^{m_u} < \frac{\delta}{2|C|}$ for the given value of m_u. Therefore, by the union bound, the number of unlabeled examples is sufficient to ensure that with probability $1 - \frac{\delta}{2}$, only hypotheses in $C_{D,\chi}(\epsilon)$ have $\widehat{err}_{unl}(f) = 0$. The number of labeled examples then similarly ensures that with probability $1 - \frac{\delta}{2}$, none of those whose true error is at least ϵ have an empirical error of 0, yielding the theorem. ■

Interpretation: If the target function indeed is perfectly correct and compatible, then Theorem 2.3.1 gives sufficient conditions on the number of examples needed to ensure that an algorithm that optimizes both quantities over the observed data will, in fact, achieve a PAC guarantee. To emphasize this, we will say that an algorithm efficiently PAC_{unl}-learns the pair (C, χ) if it is able to achieve a PAC guarantee using time and sample sizes polynomial in the bounds of Theorem 2.3.1. For a formal definition see Definition 2.3.1 at the end of this section.

We can think of Theorem 2.3.1 as bounding the number of labeled examples we need as a function of the "helpfulness" of the distribution D with respect to our notion of compatibility. That is, in our context, a helpful distribution is one in which $C_{D,\chi}(\epsilon)$ is small, and so we do not need much labeled data to identify a good function among them. We can get a similar bound in the situation when the target function is not fully compatible:

Theorem 2.3.2 *If $c^* \in C$ and $err_{unl}(c^*) = t$, then m_u unlabeled examples and m_l labeled examples are sufficient to learn to error ϵ with probability $1 - \delta$, for*

$$m_u = \frac{2}{\epsilon^2}\left[\ln|C| + \ln\frac{4}{\delta}\right] \quad \textit{and} \quad m_l = \frac{1}{\epsilon}\left[\ln|C_{D,\chi}(t+2\epsilon)| + \ln\frac{2}{\delta}\right].$$

In particular, with probability at least $1 - \delta$, the $f \in C$ that optimizes $\widehat{err}_{unl}(f)$ subject to $\widehat{err}(f) = 0$ has $err(f) \leq \epsilon$.

Alternatively, given the above number of unlabeled examples m_u, for any *number of labeled examples m_l, with probability at least $1 - \delta$, the $f \in C$ that optimizes $\widehat{err}_{unl}(f)$ subject to $\widehat{err}(f) = 0$ has*

$$err(f) \quad \leq \quad \frac{1}{m_l}\left[\ln|C_{D,\chi}(err_{unl}(c^*) + 2\epsilon)| + \ln\frac{2}{\delta}\right]. \tag{2.1}$$

Proof: By Hoeffding bounds, m_u is sufficiently large so that with probability at least $1 - \delta/2$, all $f \in C$ have $|\widehat{err}_{unl}(f) - err_{unl}(f)| \leq \epsilon$. Thus, $\{f \in C : \widehat{err}_{unl}(f) \leq t + \epsilon\} \subseteq C_{D,\chi}(t + 2\epsilon)$. For the first implication, the given bound on m_l is sufficient so that with probability at least $1 - \delta$, all $f \in C$ with $\widehat{err}(f) = 0$ and $\widehat{err}_{unl}(f) \leq t+\epsilon$ have $err(f) \leq \epsilon$; furthermore, $\widehat{err}_{unl}(c^*) \leq t+\epsilon$, so such a function f exists. Therefore, with probability at least $1-\delta$, the $f \in C$ that optimizes $\widehat{err}_{unl}(f)$ subject to $\widehat{err}(f) = 0$ has $err(f) \leq \epsilon$, as desired. For second implication, inequality (2.1) follows immediately by solving for the labeled estimation-error as a function of m_l. ■

Interpretation: Theorem 2.3.2 has several implications. Specifically:

1. If we can optimize the (empirical) unlabeled error rate subject to having zero empirical labeled error, then to achieve low true error it suffices to draw a number of labeled examples that depends logarithmically on the number of functions in C whose unlabeled error rate is at most 2ϵ greater than that of the target c^*.
2. Alternatively, for *any* given number of labeled examples m_l, we can provide a bound (given in equation 2.1) on our error rate that again depends logarithmically on the number of such functions, i.e., with high probability the function $f \in C$ that optimizes $\widehat{err}_{unl}(f)$ subject to $\widehat{err}(f) = 0$ has $err(f) \leq \frac{1}{m_l}\left[\ln|C_{D,\chi}(err_{unl}(c^*) + 2\epsilon)| + \ln\frac{2}{\delta}\right]$.
3. If we have a desired maximum error rate ϵ and do not know the value of $err_{unl}(c^*)$ but have the ability to draw additional labeled examples as needed, then we can simply do a standard "doubling trick" on m_l. On each round, we check if the hypothesis f found indeed has sufficiently low empirical unlabeled error rate, and we spread the "δ" parameter across the different runs. See, e.g., Corollary 2.3.6 in Section 2.3.1.

Finally, before going to infinite hypothesis spaces, we give a simple Occam-style version of the above bounds for this setting. Given a sample S, let us define $\mathsf{desc}_S(f) = \ln|C_{S,\chi}(\widehat{err}_{unl}(f))|$. That is, $\mathsf{desc}_S(f)$ is the description length of f (in "nats") if we sort hypotheses by their empirical compatibility and output the index of f in this ordering. Similarly, define $\epsilon\text{-}\mathsf{desc}_D(f) = \ln|C_{D,\chi}(err_{unl}(f) + \epsilon)|$. This is an upper-bound on the description length of f if we sort hypotheses by an ϵ-approximation to the their true compatibility. Then we immediately get a bound as follows:

Corollary 2.3.3 *For any set S of unlabeled data, given m_l labeled examples, with probability at least $1 - \delta$,* all *$f \in C$ satisfying $\widehat{err}(f) = 0$ and $\mathsf{desc}_S(f) \leq \epsilon m_l - \ln(1/\delta)$ have $err(f) \leq \epsilon$. Furthermore, if $|S| \geq \frac{2}{\epsilon^2}[\ln|C| + \ln\frac{2}{\delta}]$, then with probability at least $1 - \delta$, all $f \in C$ satisfy $\mathsf{desc}_S(f) \leq \epsilon\text{-}\mathsf{desc}_D(f)$.*

Interpretation: The point of this bound is that an algorithm can use observable quantities (the "empirical description length" of the hypothesis produced) to determine if it can be confident that its true error rate

is low (I.e., if we can find a hypothesis with $\mathsf{desc}_S(f) \leq \epsilon m_l - \ln(1/\delta)$ and $\widehat{err}(f) = 0$, we can be confident that it has error rate at most ϵ). Furthermore, if we have enough unlabeled data, the observable quantities will be no worse than if we were learning a slightly less compatible function using an infinite-size unlabeled sample.

Note that if we begin with a non-distribution-dependent ordering of hypotheses, inducing some description length $\mathsf{desc}(f)$, and our compatibility assumptions turn out to be wrong, then it could well be that $\mathsf{desc}_D(c^*) > \mathsf{desc}(c^*)$. In this case our use of unlabeled data would end up hurting rather than helping. However, notice that by merely interleaving the initial ordering and the ordering produced by S, we get a new description length $\mathsf{desc}_{new}(f)$ such that

$$\mathsf{desc}_{new}(f) \leq 1 + \min(\mathsf{desc}(f), \mathsf{desc}_S(f)).$$

Thus, up to an additive constant, we can get the best of both orderings.

Also, if we have the ability to purchase additional labeled examples until the function produced is sufficiently "short" compared to the amount of data, then we can perform the usual stratification and be confident whenever we find a consistent function f such that $\mathsf{desc}_S(f) \leq \epsilon m_l - \ln(\frac{m_l(m_l+1)}{\delta})$, where m_l is the number of labeled examples seen so far.

Efficient algorithms in our model Finally, we end this section with a definition describing our goals for efficient learning algorithms, based on the above sample bounds.

Definition 2.3.1 *Given a class C and compatibility notion χ, we say that an algorithm efficiently PAC_{unl}-learns the pair (C, χ) if, for any distribution D, for any target function $c^* \in C$ with $err_{unl}(c^*) = 0$, for any given $\epsilon > 0$, $\delta > 0$, with probability at least $1 - \delta$ it achieves error at most ϵ using $\mathrm{poly}(\log |C|, 1/\epsilon, 1/\delta)$ unlabeled examples and $\mathrm{poly}(\log |C_{D,\chi}(\epsilon)|, 1/\epsilon, 1/\delta)$ labeled examples, and with time which is poly$(\log |C|, 1/\epsilon, 1/\delta)$.*

We say that an algorithm semi-agnostically PAC_{unl}-learns (C, χ) if it is able to achieve this guarantee for any $c^* \in C$ even if $err_{unl}(c^*) \neq 0$, using labeled examples poly$(\log |C_{D,\chi}(err_{unl}(c^*)+\epsilon)|, 1/\epsilon, 1/\delta)$.

Infinite hypothesis spaces

To reduce notation, we will assume in the rest of this chapter that $\chi(f, x) \in \{0, 1\}$ so that $\chi(f, D) = \mathrm{Pr}_{x\sim D}[\chi(f, x) = 1]$. However, all our sample complexity results can be easily extended to the general case.

For infinite hypothesis spaces, the first issue that arises is that in order to achieve uniform convergence of *unlabeled* error rates, the set whose complexity we care about is not C but rather $\chi(C) = \{\chi_f : f \in C\}$ where $\chi_f : X \to \{0, 1\}$ and $\chi_f(x) = \chi(f, x)$. For instance, suppose examples are just points on the line, and $C = \{f_a(x) : f_a(x) = 1 \text{ iff } x \leq a\}$. In this case, $\mathrm{VCdim}(C) = 1$. However, we could imagine a compatibility function such that $\chi(f_a, x)$ depends on some complicated relationship between the real numbers a and x. In this case, $\mathrm{VCdim}(\chi(C))$ is much larger, and indeed we would need many more unlabeled examples to estimate compatibility over all of C.

A second issue is that we need an appropriate measure for the "size" of the set of surviving functions. VC-dimension tends not to be a good choice: for instance, if we consider the case of Example 1 (margins), then even if data is concentrated in two well-separated "blobs", the set of compatible separators still has as large a VC-dimension as the entire class even though they are all very similar with respect to D (see, e.g., Figure 2.1 after Theorem 2.3.5 below). Instead, it is better to consider distribution dependent complexity measures such as annealed VC-entropy [103] or Rademacher averages [43, 72, 155]. For this we introduce some notation. Specifically, for any C, we denote by $C[m, D]$ the expected number of splits of m points (drawn i.i.d.) from D using concepts in C. Also, for a given (fixed) $S \subseteq X$, we will denote by $\overline{S}$ the

uniform distribution over S, and by $C[m, \overline{S}]$ the expected number of splits of m points from $\overline{S}$ using concepts in C. The following is the analog of Theorem 2.3.2 for the infinite case.

Theorem 2.3.4 *If $c^* \in C$ and $err_{unl}(c^*) = t$, then m_u unlabeled examples and m_l labeled examples are sufficient to learn to error ϵ with probability $1 - \delta$, for*

$$m_u = O\left(\frac{VCdim\left(\chi(C)\right)}{\epsilon^2}\ln\frac{1}{\epsilon} + \frac{1}{\epsilon^2}\ln\frac{2}{\delta}\right)$$

and

$$m_l = \frac{2}{\epsilon}\left[\ln\left(2C_{D,\chi}(t+2\epsilon)[2m_l, D]\right) + \ln\frac{4}{\delta}\right],$$

where recall $C_{D,\chi}(t+2\epsilon)[2m_l, D]$ is the expected number of splits of $2m_l$ points drawn from D using concepts in C of unlabeled error rate $\leq t + 2\epsilon$. In particular, with probability at least $1 - \delta$, the $f \in C$ that optimizes $\widehat{err}_{unl}(f)$ subject to $\widehat{err}(f) = 0$ has $err(f) \leq \epsilon$.

Proof: Let S be the set of m_u unlabeled examples. By standard VC-dimension bounds (e.g., see Theorem A.1.1 in Appendix A.1.1) the number of unlabeled examples given is sufficient to ensure that with probability at least $1 - \frac{\delta}{2}$ we have $|\Pr_{x\sim S}[\chi_f(x) = 1] - \Pr_{x\sim D}[\chi_f(x) = 1]| \leq \epsilon$ for all $\chi_f \in \chi(C)$. Since $\chi_f(x) = \chi(f, x)$, this implies that we have $|\widehat{err}_{unl}(f) - err_{unl}(f)| \leq \epsilon$ for all $f \in C$. So, the set of hypotheses with $\widehat{err}_{unl}(f) \leq t + \epsilon$ is contained in $C_{D,\chi}(t + 2\epsilon)$.

The bound on the number of labeled examples now follows directly from known concentration results using the expected number of partitions instead of the maximum in the standard VC-dimension bounds (e.g., see Theorem A.1.2 in Appendix A.1.1). This bound ensures that with probability $1 - \frac{\delta}{2}$, none of the functions $f \in C_{D,\chi}(t + 2\epsilon)$ with $err(f) \geq \epsilon$ have $\widehat{err}(f) = 0$.

The above two arguments together imply that with probability $1 - \delta$, all $f \in C$ with $\widehat{err}(f) = 0$ and $\widehat{err}_{unl}(f) \leq t + \epsilon$ have $err(f) \leq \epsilon$, and furthermore c^* has $\widehat{err}_{unl}(c^*) \leq t + \epsilon$. This in turn implies that with probability at least $1 - \delta$, the $f \in C$ that optimizes $\widehat{err}_{unl}(f)$ subject to $\widehat{err}(f) = 0$ has $err(f) \leq \epsilon$ as desired. ■

We can also give a bound where we specify the number of labeled examples as a function of the *unlabeled sample*; this is useful because we can imagine our learning algorithm performing some calculations over the unlabeled data and then deciding how many labeled examples to purchase.

Theorem 2.3.5 *If $c^* \in C$ and $err_{unl}(c^*) = t$, then an unlabeled sample S of size*

$$O\left(\frac{\max[VCdim(C), VCdim(\chi(C))]}{\epsilon^2}\ln\frac{1}{\epsilon} + \frac{1}{\epsilon^2}\ln\frac{2}{\delta}\right)$$

is sufficient so that if we label m_l examples drawn uniformly at random from S, where

$$m_l > \frac{4}{\epsilon}\left[\ln(2C_{S,\chi}(t+\epsilon)\left[2m_l, \overline{S}\right]) + \ln\frac{4}{\delta}\right]$$

then with probability at least $1 - \delta$, the $f \in C$ that optimizes $\widehat{err}_{unl}(f)$ subject to $\widehat{err}(f) = 0$ has $err(f) \leq \epsilon$.

Proof: Standard VC-bounds (in the same form as for Theorem 2.3.4) imply that the number of *labeled* examples m_l is sufficient to guarantee the conclusion of the theorem with "$err(f)$" replaced by "$err_{\overline{S}}(f)$" (the error with respect to $\overline{S}$) and "ϵ" replaced with "$\epsilon/2$". The number of *unlabeled* examples is enough to ensure that, with probability $\geq 1 - \frac{\delta}{2}$, for all $f \in C$, $|err(f) - err_{\overline{S}}(f)| \leq \epsilon/2$. Combining these two statements yields the theorem. ■

Note that if we assume $err_{unl}(c^*) = 0$, then we can use the set $C_{S,\chi}(0)$ instead of $C_{S,\chi}(t+\epsilon)$ in the formula giving the number of labeled examples in Theorem 2.3.5.

Note: Notice that for the setting of Example 1, in the worst case (over distributions D) this will essentially recover the standard margin sample-complexity bounds for the number of labeled examples. In particular, $C_{S,\chi}(0)$ contains only those separators that split S with margin $\geq \gamma$, and therefore, $s = \left|C_{S,\chi}(0)[2m_l, \overline{S}]\right|$ is no greater than the maximum number of ways of splitting $2m_l$ points with margin γ. However, if the distribution is helpful, then the bounds can be much better because there may be many fewer ways of splitting S with margin γ. For instance, in the case of two well-separated "blobs" illustrated in Figure 2.1, if S is large enough, we would have just $s = 4$.

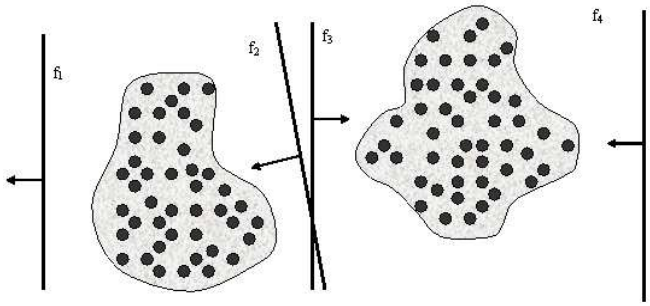

Figure 2.1: Linear separators with a margin-based notion of compatibility. If the distribution is uniform over two well-separated "blobs" and the unlabeled set S is sufficiently large, the set $C_{S,\chi}(0)$ contains only four different partitions of S, shown in the figure as f_1, f_2, f_3, and f_4. Therefore, Theorem 2.3.5 implies that we only need $O(1/\epsilon)$ labeled examples to learn well.

Theorem 2.3.5 immediately implies the following stratified version, which applies to the case in which one repeatedly draws labeled examples until that number is sufficient to justify the most-compatible hypothesis found.

Corollary 2.3.6 *An unlabeled sample S of size*

$$O\left(\frac{\max[VCdim(C), VCdim(\chi(C))]}{\epsilon^2}\ln\frac{1}{\epsilon} + \frac{1}{\epsilon^2}\ln\frac{2}{\delta}\right)$$

is sufficient so that with probability $\geq 1-\delta$ we have that simultaneously for every $k \geq 0$ the following is true: if we label m_k examples drawn uniformly at random from S, where

$$m_k > \frac{4}{\epsilon}\left[\ln\left(2C_{S,\chi}((k+1)\epsilon)\left[2m_k, \overline{S}\right]\right) + \ln\frac{4(k+1)(k+2)}{\delta}\right]$$

then all $f \in C$ with $\widehat{err}(f) = 0$ and $\widehat{err}_{unl}(f) \leq (k+1)\epsilon$ have $err(f) \leq \epsilon$.

Interpretation: This corollary is an analog of Theorem 2.3.3 and it justifies a stratification based on the estimated unlabeled error rates. That is, beginning with $k = 0$, one draws the specified number of examples and checks to see if a sufficiently compatible hypothesis can be found. If so, one halts with success, and if not, one increments k and tries again. Since $k \leq \frac{1}{\epsilon}$, we clearly have a fallback property: the number of labeled examples required is never much worse than the number of labeled examples required by a standard supervised learning algorithm.

If one does not have the ability to draw additional labeled examples, then we can fix m_l and instead stratify over estimation error as in [45]. We discuss this further in our agnostic bounds in Section 2.3.1 below.

The agnostic case

The bounds given so far have been based on the assumption that the target function belongs to C (so that we can assume there will exist $f \in C$ with $\widehat{err}(f) = 0$). One can also derive analogous results for the agnostic (unrealizable) case, where we do not make that assumption. We first present one immediate bound of this form, and then show how we can use it in order to trade off labeled and unlabeled error in a near-optimal way. We also discuss the relation of this to a common "regularization" technique used in semi-supervised learning. As we will see, the differences between these two point to certain potential pitfalls in the standard regularization approach.

Theorem 2.3.7 *Let $f_t^* = \text{argmin}_{f \in C}[err(f)|err_{unl}(f) \leq t]$. Then an unlabeled sample S of size*

$$O\left(\frac{\max[VCdim(C), VCdim(\chi(C))]}{\epsilon^2}\log\frac{1}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{2}{\delta}\right)$$

and a labeled sample of size

$$m_l \geq \frac{8}{\epsilon^2}\left[\log\left(2C_{D,\chi}(t+2\epsilon)[2m_l, D]\right) + \log\frac{4}{\delta}\right]$$

is sufficient so that with probability $\geq 1-\delta$, the $f \in C$ that optimizes $\widehat{err}(f)$ subject to $\widehat{err}_{unl}(f) \leq t+\epsilon$ has $err(f) \leq err(f_t^*) + \epsilon + \sqrt{\log(4/\delta)/(2m_l)} \leq err(f_t^*) + 2\epsilon$.

Proof: The given unlabeled sample size implies that with probability $1-\delta/2$, all $f \in C$ have $|\widehat{err}_{unl}(f) - err_{unl}(f)| \leq \epsilon$, which also implies that $\widehat{err}_{unl}(f_t^*) \leq t+\epsilon$. The labeled sample size, using standard VC bounds (e.g, Theorem A.1.3 in the Appendix A.1.2) imply that with probability at least $1-\delta/4$, all $f \in C_{D,\chi}(t+2\epsilon)$ have $|\widehat{err}(f) - err(f)| \leq \epsilon$. Finally, by Hoeffding bounds, with probability at least $1-\delta/4$ we have

$$\widehat{err}(f_t^*) \leq err(f_t^*) + \sqrt{\log(4/\delta)/(2m_l)}.$$

Therefore, with probability at least $1-\delta$, the $f \in C$ that optimizes $\widehat{err}(f)$ subject to $\widehat{err}_{unl}(f) \leq t+\epsilon$ has

$$err(f) \leq \widehat{err}(f) + \epsilon \leq \widehat{err}(f_t^*) + \epsilon \leq err(f_t^*) + \epsilon + \sqrt{\log(4/\delta)/(2m_l)} \leq err(f_t^*) + 2\epsilon,$$

as desired. ■

Interpretation: Given a value t, Theorem 2.3.7 bounds the number of labeled examples needed to achieve error at most ϵ larger than that of the best function f_t^* of unlabeled error rate at most t. Alternatively, one can also state Theorem 2.3.7 in the form more commonly used in statistical learning theory: given *any* number of labeled examples m_l and given $t > 0$, Theorem 2.3.7 implies that with high probability, the function f that optimizes $\widehat{err}(f)$ subject to $\widehat{err}_{unl}(f) \leq t+\epsilon$ satisfies

$$err(f) \leq \widehat{err}(f) + \epsilon_t \leq err(f_t^*) + \epsilon_t + \sqrt{\frac{\log(4/\delta)}{2m_l}}$$

where

$$\epsilon_t = \sqrt{\frac{8}{m_l}\log\left(8C_{D,\chi}(t+2\epsilon)[2m_l, D]/\delta\right)}.$$

Note that as usual, there is an inherent tradeoff here between the quality of the comparison function f_t^*, which improves as t increases, and the estimation error ϵ_t, which gets worse as t increases. Ideally, one

would like to achieve a bound of $\min_t[err(f_t^*) + \epsilon_t] + \sqrt{\log(4/\delta)/(2m_l)}$; i.e., as if the optimal value of t were known in advance. We can perform nearly as well as this bound by (1) performing a stratification over t (so that the bound holds simultaneously for all values of t) and (2) using an estimate $\hat{\epsilon}_t$ of ϵ_t that we can calculate from the unlabeled sample and therefore use in the optimization. In particular, letting $f_t = \text{argmin}_{f' \in C}[\widehat{err}(f') : \widehat{err}_{unl}(f') \leq t]$, we will output $f = \text{argmin}_{f_t}[\widehat{err}(f_t) + \hat{\epsilon}_t]$.

Specifically, given a set S of unlabeled examples and m_l labeled examples, let

$$\hat{\epsilon}_t = \hat{\epsilon}_t(S, m_l) = \sqrt{\frac{24}{m_l} \log\left(8C_{S,\chi}(t)[m_l, S]\right)},$$

where we define $C_{S,\chi}(t)[m_l, S]$ to be the number of different partitions of the first m_l points in S using functions in $C_{S,\chi}(t)$, i.e., using functions of empirical unlabeled error at most t (we assume $|S| \geq m_l$). Then we have the following theorem.

Theorem 2.3.8 *Let $f_t^* = \text{argmin}_{f' \in C}[err(f') | err_{unl}(f') \leq t]$ and define $\hat{\epsilon}(f') = \hat{\epsilon}_{t'}$ for $t' = \widehat{err}_{unl}(f')$. Then, given m_l labeled examples, with probability at least $1 - \delta$, the function*

$$f = \text{argmin}_{f'}[\widehat{err}(f') + \hat{\epsilon}(f')]$$

satisfies the guarantee that

$$err(f) \leq \min_t[err(f_t^*) + \hat{\epsilon}(f_t^*)] + 5\sqrt{\frac{\log(8/\delta)}{m_l}}$$

Proof: First we argue that with probability at least $1 - \delta/2$, for all $f' \in C$ we have

$$err(f') \leq \widehat{err}(f') + \hat{\epsilon}(f') + 4\sqrt{\frac{\log(8/\delta)}{m_l}}.$$

In particular, define $C_0 = C_{S,\chi}(0)$ and inductively for $k > 0$ define $C_k = C_{S,\chi}(t_k)$ for t_k such that $C_k[m_l, S] = 8C_{k-1}[m_l, S]$. (If necessary, arbitrarily order the functions with empirical unlabeled error exactly t_k and choose a prefix such that the size condition holds.) Also, we may assume without loss of generality that $C_0[m_l, S] \geq 1$. Then, using bounds of [71] (see also Appendix A), we have that with probability at least $1 - \delta/2^{k+2}$, all $f' \in C_k \setminus C_{k-1}$ satisfy:

$$\begin{aligned}
err(f') &\leq \widehat{err}(f') + \sqrt{\frac{6}{m_l}\log(C_k[m_l, S])} + 4\sqrt{\frac{1}{m_l}\log(2^{k+3}/\delta)} \\
&\leq \widehat{err}(f') + \sqrt{\frac{6}{m_l}\log(C_k[m_l, S])} + 4\sqrt{\frac{1}{m_l}\log(2^k)} + 4\sqrt{\frac{1}{m_l}\log(8/\delta)} \\
&\leq \widehat{err}(f') + \sqrt{\frac{6}{m_l}\log(C_k[m_l, S])} + \sqrt{\frac{6}{m_l}\log(8^k)} + 4\sqrt{\frac{1}{m_l}\log(8/\delta)} \\
&\leq \widehat{err}(f') + 2\sqrt{\frac{6}{m_l}\log(C_k[m_l, S])} + 4\sqrt{\frac{1}{m_l}\log(8/\delta)} \\
&\leq \widehat{err}(f') + \hat{\epsilon}(f') + 4\sqrt{\frac{1}{m_l}\log(8/\delta)}.
\end{aligned}$$

Now, let $f^* = \text{argmin}_{f_t^*}[err(f_t^*) + \hat{\epsilon}(f_t^*)]$. By Hoeffding bounds, with probability at least $1 - \delta/2$ we have $\widehat{err}(f^*) \leq err(f^*) + \sqrt{\log(2/\delta)/(2m_l)}$. Also, by construction we have $\widehat{err}(f) + \hat{\epsilon}(f) \leq \widehat{err}(f^*) + \hat{\epsilon}(f^*)$.

Therefore with probability at least $1-\delta$ we have:

$$\begin{aligned} err(f) &\leq \widehat{err}(f) + \hat{\epsilon}(f) + 4\sqrt{\log(8/\delta)/m_l} \\ &\leq \widehat{err}(f^*) + \hat{\epsilon}(f^*) + 4\sqrt{\log(8/\delta)/m_l} \\ &\leq err(f^*) + \hat{\epsilon}(f^*) + 5\sqrt{\log(8/\delta)/m_l} \end{aligned}$$

as desired. ■

The above result bounds the error of the function f produced in terms of the quantity $\hat{\epsilon}(f^*)$ which depends on the *empirical* unlabeled error rate of f^*. If our unlabeled sample S is sufficiently large to estimate all unlabeled error rates to $\pm\epsilon$, then with high probability we have $\widehat{err}(f_t^*) \leq t+\epsilon$, so $\hat{\epsilon}(f_t^*) \leq \hat{\epsilon}_{t+\epsilon}$, and moreover $C_{S,\chi}(t+\epsilon) \subseteq C_{D,\chi}(t+2\epsilon)$. So, our error term $\hat{\epsilon}(f_t^*)$ is at most $\sqrt{\frac{24}{m_l}\log\left(8C_{D,\chi}(t+2\epsilon)[m_l,S]\right)}$. Recall that our ideal error term ϵ_t for the case that t was given to the algorithm in advance, factoring out the dependence on δ, was $\sqrt{\frac{8}{m_l}\log\left(8C_{D,\chi}(t+2\epsilon)[2m_l,D]\right)}$. [71] show that for any class C, the quantity $\log(C[m,S])$ is tightly concentrated about $\log(C[m,D])$ (see also Theorem A.1.6 in the Appendix A.1.2), so up to multiplicative constants, these two bounds are quite close.

Interpretation and use of unlabeled error rate as a regularizer: The above theorem suggests to optimize the sum of the empirical labeled error rate and an estimation-error bound based on the unlabeled error rate. A common related approach used in practice in machine learning (e.g., [82]) is to just directly optimize the sum of the two kinds of error: i.e., to find $\mathrm{argmin}_f[\widehat{err}(f) + \widehat{err}_{unl}(f)]$. However, this is not generically justified in our framework, because the labeled and unlabeled error rates are really of different "types". In particular, depending on the concept class and notion of compatibility, a small change in unlabeled error rate could substantially change the size of the compatible set.[3] For example, suppose all functions in C have unlabeled error rate 0.6, except for two: function f_0 has unlabeled error rate 0 and labeled error rate $1/2$, and function $f_{0.5}$ has unlabeled error rate 0.5 and labeled error rate $1/10$. Suppose also that C is sufficiently large that with high probability it contains some functions f that drastically overfit, giving $\widehat{err}(f) = 0$ even though their true error is close to $1/2$. In this case, we would like our algorithm to pick out $f_{0.5}$ (since its labeled error rate is fairly low, and we cannot trust the functions of unlabeled error 0.6). However, even if we use a regularization parameter λ, there is no way to make $f_{0.5} = \mathrm{argmin}_f[\widehat{err}(f) + \lambda err_{unl}(f)]$: in particular, one cannot have $1/10 + 0.5\lambda \leq \min[1/2 + 0\lambda, 0 + 0.6\lambda]$. So, in this case, this approach will not have the desired behavior.

Note: One could further derive tighter bounds, both in terms of labeled and unlabeled examples, that are based on other distribution dependent complexity measures and using stronger concentration results (see e.g. [72]).

2.3.2 ϵ-Cover-based Bounds

The results in the previous section are uniform convergence bounds: they provide guarantees for *any* algorithm that optimizes over the observed data. In this section, we consider stronger bounds based on ϵ-covers that apply to algorithms that behave in a specific way: they first use the unlabeled examples to choose a "representative" set of compatible hypotheses, and then use the labeled sample to choose among these. Bounds based on ϵ-covers exist in the classical PAC setting, but in our framework these bounds and algorithms of this type are especially natural, and the bounds are often much lower than what can be achieved via uniform convergence. For simplicity, we restrict ourselves in this section to the realizable

[3] On the other hand, for certain compatibility notions and under certain natural assumptions, one can use unlabeled error rate directly, e.g., see e.g., [196].

case. However one can combine ideas in Section 2.3.1 with ideas in this section in order to derive bounds in the agnostic case as well. We first present our generic bounds. In Section 2.3.2 we discuss natural settings in which they can be especially useful, and in then Section 2.3.2 we present even tighter bounds for co-training.

Recall that a set $C_\epsilon \subseteq 2^X$ is an ϵ-cover for C with respect to D if for every $f \in C$ there is a $f' \in C_\epsilon$ which is ϵ-close to f. That is, $\Pr_{x\sim D}(f(x) \neq f'(x)) \leq \epsilon$.

We start with a theorem that relies on knowing a good upper bound on the unlabeled error rate of the target function $err_{unl}(c^*)$.

Theorem 2.3.9 *Assume $c^* \in C$ and let p be the size of a minimum ϵ-cover for $C_{D,\chi}(err_{unl}(c^*) + 2\epsilon)$. Then using m_u unlabeled examples and m_l labeled examples for*

$$m_u = O\left(\frac{\max[VCdim(C), VCdim(\chi(C))]}{\epsilon^2}\log\frac{1}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{2}{\delta}\right) \text{ and } m_l = O\left(\frac{1}{\epsilon}\ln\frac{p}{\delta}\right),$$

we can with probability $1-\delta$ identify a hypothesis $f \in C$ with $err(f) \leq 6\epsilon$.

Proof: Let $t = err_{unl}(c^*)$. Now, given the unlabeled sample S_U, define $C' \subseteq C$ as follows: for every labeling of S_U that is consistent with some f in C, choose a hypothesis in C for which $\widehat{err}_{unl}(f)$ is smallest among all the hypotheses corresponding to that labeling. Next, we obtain C_ϵ by eliminating from C' those hypotheses f with the property that $\widehat{err}_{unl}(f) > t + \epsilon$. We then apply a greedy procedure on C_ϵ to obtain $G_\epsilon = \{g_1, \cdots, g_s\}$, as follows:

Initialize $C_\epsilon^1 = C_\epsilon$ and $i = 1$.

1. Let $g_i = \operatorname{argmin}_{f\in C_\epsilon^i} \widehat{err}_{unl}(f)$.
2. Using the unlabeled sample S_U, determine C_ϵ^{i+1} by deleting from C_ϵ^i those hypotheses f with the property that $\hat{d}(g_i, f) < 3\epsilon$.
3. If $C_\epsilon^{i+1} = \emptyset$ then set $s = i$ and stop; else, increase i by 1 and goto 1.

We now show that with high probability, G_ϵ is a 5ϵ-cover of $C_{D,\chi}(t)$ with respect to D and has size at most p. First, our bound on m_u is sufficient to ensure that with probability $\geq 1 - \frac{\delta}{2}$, we have (a) $|\hat{d}(f,g) - d(f,g)| \leq \epsilon$ for all $f, g \in C$ and (b) $|\widehat{err}_{unl}(f) - err_{unl}(f)| \leq \epsilon$ for all $f \in C$. Let us assume in the remainder that this (a) and (b) are indeed satisfied. Now, (a) implies that any two functions in C that agree on S_U have distance at most ϵ, and therefore C' is an ϵ-cover of C. Using (b), this in turn implies that C_ϵ is an ϵ-cover for $C_{D,\chi}(t)$. By construction, G_ϵ is a 3ϵ-cover of C_ϵ with respect to distribution $\widehat{S_U}$, and thus (using (a)) G_ϵ is a 4ϵ-cover of C_ϵ with respect to D, which implies that G_ϵ is a 5ϵ-cover of $C_{D,\chi}(t)$ with respect to D.

We now argue that G_ϵ has size at most p. Fix some optimal ϵ-cover $\{f_1, \ldots, f_p\}$ of $C_{D,\chi}(err_{unl}(c^*) + 2\epsilon)$. Consider function g_i and suppose that g_i is covered by $f_{\sigma(i)}$. Then the set of functions deleted in step (2) of the procedure include those functions f satisfying $d(g_i, f) < 2\epsilon$ which by triangle inequality includes those satisfying $d(f_{\sigma(i)}, f) \leq \epsilon$. Therefore, the set of functions deleted include those covered by $f_{\sigma(i)}$ and so for all $j > i$, $\sigma(j) \neq \sigma(i)$; in particular, σ is 1-1. This implies that G_ϵ has size at most p.

Finally, to learn c^* we simply output the function $f \in G_\epsilon$ of lowest empirical error over the labeled sample. By Chernoff bounds, the number of labeled examples is enough to ensure that with probability $\geq 1 - \frac{\delta}{2}$ the empirical optimum hypothesis in G_ϵ has true error at most 6ϵ. This implies that overall, with probability $\geq 1 - \delta$, we find a hypothesis of error at most 6ϵ. ■

Note that Theorem 2.3.9 relies on knowing a good upper bound on $err_{unl}(c^*)$. If we do not have such an upper bound, then one can perform a stratification as in Sections 2.3.1 and 2.3.1. For example, if we have a desired maximum error rate ϵ and we do not know a good upper bound for $err_{unl}(c^*)$ but

we have the ability to draw additional labeled examples as needed, then we can simply run the procedure in Theorem 2.3.9 for various value of p, testing on each round to see if the hypothesis f found indeed has zero empirical labeled error rate. One can show that $m_l = O\left(\frac{1}{\epsilon}\ln\frac{p}{\delta}\right)$ labeled examples are sufficient in total for all the "validation" steps.[4] If the number of labeled examples m_l is fixed, then one can also perform a stratification over the target error ϵ.

Some illustrative examples

To illustrate the power of ϵ-cover bounds, we now present two examples where these bounds allow for learning from significantly fewer labeled examples than is possible using uniform convergence.

Graph-based learning: Consider the setting of graph-based algorithms (e.g., Example 3). In particular, the input is a graph g where each node is an example and C is the class of all boolean functions over the nodes of g. Let us define the incompatibility of a hypothesis to be the fraction of edges in g cut by it. Suppose now that the graph g consists of two cliques of $n/2$ vertices, connected together by $\epsilon n^2/4$ edges. Suppose the target function c^* labels one of the cliques as positive and one as negative, so the target function indeed has unlabeled error rate less than ϵ. Now, given any set S_L of $m_l < \epsilon n/4$ labeled examples, there is always a highly-compatible hypothesis consistent with S_L that just separates the positive points in S_L from the entire rest of the graph: the number of edges cut will be at most $nm_l < \epsilon n^2/4$. However, such a hypothesis has true error nearly $1/2$ since it has less than $\epsilon n/4$ positive examples. So, we do not yet have uniform convergence over the space of highly compatible hypotheses, since this hypothesis has zero empirical error but high true error. Indeed, this illustrates an overfitting problem that can occur with a direct minimum-cut approach to learning [59, 67, 140]. On the other hand, the set of functions of unlabeled error rate less than ϵ has a small ϵ-cover: in particular, *any* partition of g that cuts less than $\epsilon n^2/4$ edges must be ϵ-close to (a) the all-positive function, (b) the all-negative function, (c) the target function c^*, or (d) the complement of the target function $1 - c^*$. So, ϵ-cover bounds act as if the concept class had only 4 functions and so by Theorem 2.3.9 we need only $O(\frac{1}{\epsilon}\log\frac{1}{\delta})$ labeled examples to learn well.[5] (In fact, since the functions in the cover are all far from each other, we really need only $O(\log\frac{1}{\delta})$ examples. This issue is explored further in Theorem 2.3.11).

Simple co-training: For another case where ϵ-cover bounds can beat uniform-convergence bounds, imagine examples are *pairs* of points in $\{0,1\}^d$, C is the class of linear separators, and compatibility is determined by whether both points are on the same side of the separator (i.e., the case of Example 4). Now suppose for simplicity that the target function just splits the hypercube on the first coordinate, and the distribution is uniform over pairs having the same first coordinate (so the target is fully compatible). We then have the following.

Theorem 2.3.10 *Given poly(d) unlabeled examples S_U and $\frac{1}{4}\log d$ labeled examples S_L, with high probability there will exist functions of true error $1/2 - 2^{-\frac{1}{2}\sqrt{d}}$ that are consistent with S_L and compatible with S_U.*

Proof: Let V be the set of all variables (not including x_1) that (a) appear in *every* positive example of S_L and (b) appear in *no* negative example of S_L. In other words, these are variables x_i such that

[4] Specifically, note that as we increase t (our current estimate for the unlabeled error rate of the target function), the associated p (which is an integer) increases in discrete jumps, $p_1, p_2, \ldots$. We can then simply spread the "δ" parameter across the different runs, in particular run i would use $\delta/i(i+1)$. Since $p_i \geq i$, this implies that $m_l = O\left(\frac{1}{\epsilon}\ln\frac{p}{\delta}\right)$ labeled examples are sufficient for all the "validation" steps.

[5] Effectively, ϵ-cover bounds allow one to rule out a hypothesis that, say, just separates the positive points in S_L from the rest of the graph by noting that this hypothesis is very close (with respect to D) to the all-negative hypothesis, and *that* hypothesis has a high labeled-error rate.

the function $f(x) = x_i$ correctly classifies all examples in S_L. Over the draw of S_L, each variable has a $(1/2)^{2|S_L|} = 1/\sqrt{d}$ chance of belonging to V, so the expected size of V is $(d-1)/\sqrt{d}$ and so by Chernoff bounds, with high probability V has size at least $\frac{1}{2}\sqrt{d}$. Now, consider the hypothesis corresponding to the conjunction of all variables in V. This correctly classifies the examples in S_L, and with probability at least $1 - 2|S_U|2^{-|V|}$ it classifies *every* other example in S_U negative because each example in S_U has only a $1/2^{|V|}$ chance of satisfying every variable in V. Since $|S_U| = \text{poly}(d)$, this means that with high probability this conjunction is compatible with S_U and consistent with S_L, even though its true error is at least $1/2 - 2^{-\frac{1}{2}\sqrt{d}}$. ■

So, given only a set S_U of poly(d) unlabeled examples and a set S_L of $\frac{1}{4}\log d$ labeled examples we would not want to use a uniform convergence based algorithm since we do not yet have uniform convergence. In contrast, the cover-size of the set of functions compatible with S_U is constant, so ϵ-cover based bounds again allow learning from just only $O(\frac{1}{\epsilon}\log\frac{1}{\delta})$ labeled examples (Theorem 2.3.9). In fact as we show in Theorem 2.3.11 we only need $O\left(\log_{\frac{1}{\epsilon}}\frac{1}{\delta}\right)$ labeled examples in this case.

Learning from even fewer labeled examples

In some cases, unlabeled data can allow us to learn from even fewer labeled examples than given by Theorem 2.3.9. In particular, consider a co-training setting where the target c^* is fully compatible *and* D satisfies the property that the two views x_1 and x_2 are conditionally independent given the label $c^*(\langle x_1, x_2\rangle)$. As shown by [62], one can boost any weak hypothesis from unlabeled data in this setting (assuming one has enough labeled data to produce a weak hypothesis). Related sample complexity results are given in [97]. In fact, we can use the notion of ϵ-covers to show that we can learn from just a single labeled example. Specifically, for any concept classes C_1 and C_2, we have:

Theorem 2.3.11 *Assume that $err(c^*) = err_{unl}(c^*) = 0$ and D satisfies independence given the label. Then for any $\tau \leq \epsilon/4$, using m_u unlabeled examples and m_l labeled examples we can find a hypothesis that with probability $1-\delta$ has error at most ϵ, for*

$$m_u = O\left(\frac{1}{\tau}\left[(VCdim(C_1) + VCdim(C_2))\ln\frac{1}{\tau} + \ln\frac{2}{\delta}\right]\right) \text{ and } m_l = O\left(\log_{\frac{1}{\tau}}\frac{1}{\delta}\right).$$

Proof: We will assume for simplicity the setting of Example 3, where $c^* = c_1^* = c_2^*$ and also $D_1 = D_2 = \tilde{D}$ (the general case is handled similarly, but just requires more notation).

We start by characterizing the hypotheses with low unlabeled error rate. Recall that $\chi(f, D) = \Pr_{\langle x_1,x_2\rangle\sim D}[f(x_1) = f(x_2)]$, and for concreteness assume f predicts using x_1 if $f(x_1) \neq f(x_2)$. Consider $f \in C$ with $err_{unl}(f) \leq \tau$ and let's define $p_- = \Pr_{x\in\tilde{D}}[c^*(x) = 0]$, $p_+ = \Pr_{x\in\tilde{D}}[c^*(x) = 1]$ and for $i, j \in \{0, 1\}$ define $p_{ij} = \Pr_{x\in\tilde{D}}[f(x) = i, c^*(x) = j]$. We clearly have $err(f) = p_{10} + p_{01}$. From $err_{unl}(f) = \Pr_{(x_1,x_2)\sim D}[f(x_1) \neq f(x_2)] \leq \tau$, using the independence given the label of D, we get

$$\frac{2p_{10}p_{00}}{p_{10} + p_{00}} + \frac{2p_{01}p_{11}}{p_{01} + p_{11}} \leq \tau.$$

In particular, the fact that $\frac{2p_{10}p_{00}}{p_{10}+p_{00}} \leq \tau$ implies that we cannot have both $p_{10} > \tau$ and $p_{00} > \tau$, and the fact that $\frac{2p_{01}p_{11}}{p_{01}+p_{11}} \leq \tau$ implies that we cannot have both $p_{01} > \tau$ and $p_{11} > \tau$. Therefore, any hypothesis f with $err_{unl}(f) \leq \tau$ falls in one of the following categories:

1. f is "close to c^*": $p_{10} \leq \tau$ and $p_{01} \leq \tau$; so $err(f) \leq 2\tau$.
2. f is "close to $\overline{c^*}$": $p_{00} \leq \tau$ and $p_{11} \leq \tau$; so $err(f) \geq 1 - 2\tau$.

3. f "almost always predicts negative": for $p_{10} \leq \tau$ and $p_{11} \leq \tau$; so $\Pr[f(x) = 0] \geq 1 - 2\tau$.
4. f "almost always predicts positive": for $p_{00} \leq \tau$ and $p_{01} \leq \tau$; so $\Pr[f(x) = 0] \leq 2\tau$.

Let f_1 be the constant positive function and f_0 be the constant negative function. Now note that our bound on m_u is sufficient to ensure that with probability $\geq 1 - \frac{\delta}{2}$, we have (a) $|\hat{d}(f,g) - d(f,g)| \leq \tau$ for all $f, g \in C$ and (b) all $f \in C$ with $\widehat{err}_{unl}(f) = 0$ satisfy $err_{unl}(f) \leq \tau$. Let us assume in the remainder that this (a) and (b) are indeed satisfied. By our previous analysis, there are at most four kinds of hypotheses consistent with unlabeled data: those close to c^*, those close to its complement $\overline{c^*}$, those close to f_0, and those close to f_1. Furthermore, c^*, $\overline{c^*}$, f_0, and f_1 *are* compatible with the unlabeled data.

So, algorithmically, we first check to see if there exists a hypothesis $g \in C$ with $\widehat{err}_{unl}(g) = 0$ such that $\hat{d}(f_1, g) \geq 3\tau$ and $\hat{d}(f_0, g) \geq 3\tau$. If such a hypothesis g exists, then it must satisfy either case (1) or (2) above. Therefore, we know that one of $\{g, \overline{g}\}$ is 2τ-close to c^*. If not, we must have $p_+ \leq 4\tau$ or $p_- \leq 4\tau$, in which case we know that one of $\{f_0, f_1\}$ is 4τ-close to c^*. So, either way we have a set of two functions, opposite to each other, one of which is at least 4τ-close to c^*. We finally use $O(\log_{\frac{1}{\tau}} \frac{1}{\delta})$ labeled examples to pick one of these to output, namely the one with lowest empirical labeled error. Lemma 2.3.12 below then implies that with probability $1 - \delta$ the function we output has error at most $4\tau \leq \epsilon$. ■

Lemma 2.3.12 *Consider $\tau < \frac{1}{8}$. Let $C_\tau = \{f, \overline{f}\}$ be a subset of C containing two opposite hypotheses with the property that one of them is τ-close to c^*. Then, $m_l > 6\log_{\left(\frac{1}{\tau}\right)}\left(\frac{1}{\delta}\right)$ labeled examples are sufficient so that with probability $\geq 1 - \delta$, the concept in C_τ that is τ-close to c^* in fact has lower empirical error.*

Proof: We need to show that if $m_l > 6\log_{\frac{1}{\tau}}\left(\frac{1}{\delta}\right)$, then

$$\sum_{k=0}^{\lfloor \frac{m_l}{2} \rfloor} \binom{m_l}{k} \tau^{(m_l - k)} (1-\tau)^k \leq \delta.$$

Since $\tau < \frac{1}{8}$ we have:

$$\sum_{k=0}^{\lfloor \frac{m_l}{2} \rfloor} \binom{m_l}{k} \tau^{(m_l - k)} (1-\tau)^k \leq \sum_{k=0}^{\lfloor \frac{m_l}{2} \rfloor} \binom{m_l}{k} \tau^{(m_l - k)} = \tau^{\lfloor \frac{m_l}{2} \rfloor} \sum_{k=0}^{\lfloor \frac{m_l}{2} \rfloor} \binom{m_l}{k} \tau^{\lceil \frac{m_l}{2} \rceil - k}$$

and so $S \leq (\sqrt{\tau} \cdot 2)^{m_l}$. For $\tau < \frac{1}{8}$ and $m_l > 6\frac{\log_2\left(\frac{1}{\delta}\right)}{\log_2\left(\frac{1}{\tau}\right)} = 6\log_{\left(\frac{1}{\tau}\right)}\left(\frac{1}{\delta}\right)$ it's easy to see that $(\sqrt{\tau} \cdot 2)^{m_l} < \delta$, which implies the desired result. ■

In particular, by reducing τ to poly(δ) in Theorem 2.3.11, we can reduce the number of labeled examples needed m_l to *one*. Note however that we will need polynomially more unlabeled examples.

In fact, the result in Theorem 2.3.11 can be extended to the case that D^+ and D^- merely satisfy constant expansion rather than full independence given the label, see [28].

Note: Theorem 2.3.11 illustrates that if data is especially well behaved with respect to the compatibility notion, then our bounds on labeled data can be extremely good. In Section 2.4.2, we show for the case of linear separators and independence given the label, we can give *efficient* algorithms, achieving the bounds in Theorem 2.3.11 in terms of labeled examples by a polynomial time algorithm. Note, however, that both these bounds rely heavily on the assumption that the target is fully compatible. If the assumption is more of a "hope" than a belief, then one would need an additional sample of $1/\epsilon$ labeled examples just to validate the hypothesis produced.

2.4 Algorithmic Results

In this section we give several examples of *efficient* algorithms in our model that are able to learn using sample sizes comparable to those described in Section 2.3. Note that our focus is on achieving a low-error hypothesis (also called minimizing 0-1 loss). Another common practice in machine learning (both in the context of supervised and semi-supervised learning) is to instead try to minimize a surrogate convex loss that is easier to optimize [82]. While this does simplify the computational problem, it does not in general solve the true goal of achieving low error.

2.4.1 A simple case

We give here a simple example to illustrate the bounds in Section 2.3.1, and for which we can give a polynomial-time algorithm that takes advantage of them. Let the instance space $X = \{0,1\}^d$, and for $x \in X$, let $\mathsf{vars}(x)$ be the set of variables set to 1 in the feature vector x. Let C be the class of monotone disjunctions (e.g., $x_1 \vee x_3 \vee x_6$), and for $f \in C$, let $\mathsf{vars}(f)$ be the set of variables disjoined by f. Now, suppose we say an example x is compatible with function f if either $\mathsf{vars}(x) \subseteq \mathsf{vars}(f)$ or else $\mathsf{vars}(x) \cap \mathsf{vars}(f) = \phi$. This is a very strong notion of "margin": it says, in essence, that every variable is either a positive indicator or a negative indicator, and no example should contain both positive and negative indicators.

Given this setup, we can give a simple PAC_{unl}-learning algorithm for this pair (C, χ): that is, an algorithm with sample size bounds that are polynomial (or in this case, matching) those in Theorem 2.3.1. Specifically, we can prove the following:

Theorem 2.4.1 *The class C of monotone disjunctions is PAC_{unl}-learnable under the compatibility notion defined above.*

Proof: We begin by using our unlabeled data to construct a graph on d vertices (one per variable), putting an edge between two vertices i and j if there is any example x in our unlabeled sample with $i, j \in \mathsf{vars}(x)$. We now use our labeled data to label the components. If the target function is fully compatible, then no component will get multiple labels (if some component does get multiple labels, we halt with failure). Finally, we produce the hypothesis f such that $\mathsf{vars}(f)$ is the union of the positively-labeled components. This is fully compatible with the unlabeled data and has zero error on the labeled data, so by Theorem 2.3.1, if the sizes of the data sets are as given in the bounds, with high probability the hypothesis produced will have error at most ϵ. ■

Notice that if we want to view the algorithm as "purchasing" labeled data, then we can simply examine the graph, count the number of connected components k, and then request $\frac{1}{\epsilon}[k \ln 2 + \ln \frac{2}{\delta}]$ labeled examples. (Here, $2^k = |C_{S,\chi}(0)|$.) By the proof of Theorem 2.3.1, with high probability $2^k \leq |C_{D,\chi}(\epsilon)|$, so we are purchasing no more than the number of labeled examples in the theorem statement.

Also, it is interesting to see the difference between a "helpful" and "non-helpful" distribution for this problem. An especially *non*-helpful distribution would be the uniform distribution over all examples x with $|\mathsf{vars}(x)| = 1$, in which there are d components. In this case, unlabeled data does not help at all, and one still needs $\Omega(d)$ labeled examples (or, even $\Omega\left(\frac{d}{\epsilon}\right)$ if the distribution is non-uniform as in the lower bounds of [105]). On the other hand, a helpful distribution is one such that with high probability the number of components is small, such as the case of features appearing independently given the label.

2.4.2 Co-training with linear separators

We now consider the case of co-training where the hypothesis class C is the class of linear separators. For simplicity we focus first on the case of Example 4: the target function is a linear separator in R^d and each

example is a *pair* of points, both of which are assumed to be on the same side of the separator (i.e., an example is a line-segment that does not cross the target hyperplane). We then show how our results can be extended to the more general setting.

As in the previous example, a natural approach is to try to solve the "consistency" problem: given a set of labeled and unlabeled data, our goal is to find a separator that is consistent with the labeled examples and compatible with the unlabeled ones (i.e., it gets the labeled data correct and doesn't cut too many edges). Unfortunately, this consistency problem is NP-hard: given a graph g embedded in R^d with two distinguished points s and t, it is NP-hard to find the linear separator with s on one side and t on the other that cuts the minimum number of edges, *even if the minimum is zero* [108]. For this reason, we will make an additional assumption, that the two points in an example are each drawn *independently given the label*. That is, there is a single distribution $\tilde{D}$ over R^d, and with some probability p_+, two points are drawn i.i.d. from $\tilde{D}_+$ ($\tilde{D}$ restricted to the positive side of the target function) and with probability $1 - p_+$, the two are drawn i.i.d from $\tilde{D}_-$ ($\tilde{D}$ restricted to the negative side of the target function). Note that our sample complexity results in Section 2.3.2 extend to weaker assumptions such as distributional expansion introduced by [28], but we need true independence for our algorithmic results. [62] also give positive algorithmic results for co-training when (a) the two views of an example are drawn independently given the label (which we are assuming now), (b) the underlying function is learnable via Statistical Query algorithms[6] (which is true for linear separators [64]), and (c) we have enough labeled data to produce a weakly-useful hypothesis (defined below) on one of the views to begin with. We give here an improvement over that result by showing how we can run the algorithm in [62] with only *a single* labeled example, thus obtaining an efficient algorithm in our model. It is worth noticing that in the process, we also somewhat simplify the results of [64] on efficiently learning linear separators with noise without a margin assumption.

For the analysis below, we need the following definition. A *weakly-useful* predictor is a function f such that for some β that is at least inverse polynomial in the input size we have:

$$\Pr[f(x) = 1|c^*(x) = 1] > \Pr[f(x) = 1|c^*(x) = 0] + \beta. \tag{2.2}$$

It is equivalent to the usual notion of a "weak hypothesis" [149] when the target function is balanced, but requires the hypothesis give more information when the target function is unbalanced [62]. Also, we will assume for convenience that the target separator passes through the origin, and let us denote the separator by $c^* \cdot x = 0$.

We now describe an efficient algorithm to learn to any desired error rate ϵ in this setting from just a single labeled example. For clarity, we first describe an algorithm whose running time depends polynomially on both the dimension d and $1/\gamma$, where γ is a soft *margin* of separation between positive and negative examples. Formally, in this case we assume that at least some non-negligible probability mass of examples x satisfy $\frac{|x \cdot c^*|}{|x||c^*|} \geq \gamma$; i.e., they have distance at least γ to the separating hyperplane $x \cdot c^* = 0$ after normalization. This is a common type of assumption in machine learning (in fact, often one makes the much stronger assumption that *nearly all* probability mass is on examples x satisfying this condition). We then show how one can replace the dependence on $1/\gamma$ with instead a polynomial dependence on the number of bits of precision b in the data, using the Outlier Removal Lemma of [64] and [104].

Theorem 2.4.2 *Assume that at least an α probability mass of examples x have margin $\frac{|x \cdot c^*|}{|x||c^*|} \geq \gamma$ with respect to the target separator c^*. There is a polynomial-time algorithm (polynomial in d, $1/\gamma$, $1/\alpha$, $1/\epsilon$, and $1/\delta$) to learn a linear separator under the above assumptions, from a polynomial number of unlabeled examples and a single labeled example.*

[6]For a detailed description of the Statistical Query model see [148] and [149].

Algorithm 1 Co-training with Linear Separators. The Soft Margin Case.

Input: ϵ, δ, T a set S_L of m_l labeled examples drawn i.i.d from D, a set S_U of m_u unlabeled examples drawn i.i.d from D.

Output: Hypothesis of low error.

Let h_p be the all-positive function. Let h_n be the all-negative function. Let $\tau = \epsilon/6$, $\epsilon_1 = \tau/4$.

(1) For $i = 1, \ldots, \mathrm{T}$ do

- Choose a random halfspace f_i going through the origin.
- Feed f_i, S_U and error parameters ϵ_1 and confidence parameter $\delta/6$ into the bootstrapping procedure of [62] to produce h_i.

(2) Let h be $\mathrm{argmin}_{h_i}\left\{\widehat{err}_{unl}(h_i) | \hat{d}(h, h_p) \geq 3\tau, \hat{d}(h, h_n) \geq 3\tau\right\}$.
If $\widehat{err}_{unl}(h_i) \geq 3\epsilon_1$, then let $h = h_p$.

(3) Use S_L to output either h or $\bar{h}$: output the hypothesis with lowest empirical error on the set S_L.

Proof: Let ϵ and δ be the desired accuracy and confidence parameters. Let $\mathrm{T} = O\left(\frac{1}{\alpha \cdot \gamma} \log\left(\frac{1}{\delta}\right)\right)$, $m_u = \mathrm{poly}(1/\gamma, 1/\alpha, 1/\epsilon, 1/\delta, d)$, and $m_l = 1$. We run Algorithm 1 with the inputs ϵ, δ, T S_L, S_U, and $m_l = 1$. Let $\tau = \epsilon/6$, $\epsilon_1 = \tau/4$.

In order to prove the desired result, we start with a few facts.

We first note that our bound on m_u is sufficient to ensure that with probability $\geq 1 - \frac{\delta}{3}$, we have (a) $|\hat{d}(f, g) - d(f, g)| \leq \tau$ for all $f, g \in C$ and (b) all $f \in C$ have $|\widehat{err}_{unl}(f) - err_{unl}(f)| \leq \epsilon_1$.

We now argue that if at least an α probability mass of examples x have margin $\frac{|x \cdot c^*|}{|x||c^*|} \geq \gamma$ with respect to the target separator c^*, then a *random* halfspace has at least a $\mathrm{poly}(\alpha, \gamma)$ probability of being a weakly-useful predictor. (Note that [64] uses the Perceptron algorithm to get weak learning; here, we need something simpler since we need to save our labeled example to the very end.) Specifically, consider a point x of margin $\gamma_x \geq \gamma$. By definition, the margin is the cosine of the angle between x and c^*, and therefore the angle between x and c^* is $\pi/2 - \cos^{-1}(\gamma_x) \leq \pi/2 - \gamma$. Now, imagine that we draw f at random subject to $f \cdot c^* \geq 0$ (half of the f's will have this property) and define $f(x) = \mathrm{sign}(f \cdot x)$. Then,

$$\Pr_f(f(x) \neq c^*(x) | f \cdot c^* \geq 0) \leq (\pi/2 - \gamma)/\pi = 1/2 - \gamma/\pi.$$

Moreover, if x does *not* have margin γ then at the very least we have $\Pr_f(f(x) \neq c^*(x) | f \cdot c^* \geq 0) \leq 1/2$.

Now define distribution $D^* = \frac{1}{2}D_+ + \frac{1}{2}D_-$; that is D^* is the distribution D but balanced to 50% positive and 50% negative. With respect to D^* at least an $\alpha/2$ probability mass of the examples have margin at least γ, and therefore:

$$\mathbf{E}_f[err_{D^*}(f) | f \cdot c^* \geq 0] \leq 1/2 - (\alpha/2)(\gamma/\pi).$$

Since $err(f)$ is a bounded quantity, by Markov inequality this means that at least an $\Omega(\alpha\gamma)$ probability mass of functions f must satisfy $err_{D^*}(f) \leq \frac{1}{2} - \frac{\alpha\gamma}{4\pi}$ which in turn implies that they must be useful weakly predictors with respect to D as defined in Equation (2.2) with $\beta = \frac{\alpha\gamma}{4\pi}$.

The second part of the argument is as follows. Note that in Step(1) of our algorithm we repeat the following process for T iterations: pick a random f_i, and plug it into the bootstrapping theorem of [62] (which, given a distribution over unlabeled pairs $\langle x_1^j, x_2^j \rangle$, will use $f_i(x_1^j)$ as a noisy label of x_2^j, feeding the result into a Statistical Query algorithm). Since $\mathrm{T} = O\left(\frac{1}{\alpha \cdot \gamma} \log\left(\frac{1}{\delta}\right)\right)$, using the above observation about

random halfspaces being weak predictors, we obtain that with high probability at least $1-\delta/6$, at least one of the random hypothesis f_i was a weakly-useful predictor; and since $m_u = \text{poly}(1/\gamma, 1/\alpha, 1/\epsilon, 1/\delta, d)$ we also have the associated hypothesis h_i output by the bootstrapping procedure of [62] will with probability at least $1-\delta/6$ satisfy $err(h_i) \leq \epsilon_1$. This implies that with high probability at least $1-2\delta/3$, at least one of the hypothesis h_i we find in Step 1 has true labeled error at most ϵ_1. For the rest of the hypotheses we find in Step 1, we have no guarantees.

We now observe the following. First of all, any function f with small $err(f)$ must have small $err_{unl}(f)$; in particular,

$$err_{unl}(f) = \Pr(f(x_1) \neq f(x_2)) \leq 2err(f).$$

This implies that with high probability at least $1-2\delta/3$, at least one of the hypothesis h_i we find in Step 1 has true unlabeled error at most $2\epsilon_1$, and therefore empirical unlabeled error at most $3\epsilon_1$. Secondly, because of the assumption of independence given the label, as shown in Theorem 2.3.11, with high probability the *only* functions with unlabeled error at most τ are functions 2τ-close to c^*, 2τ-close to $\neg c^*$, 2τ-close to the "all positive" function, or 2τ-close to the "all negative" function.

In Step (2) we first examine all the hypotheses produced in Step 1, and we pick the hypothesis h with the smallest empirical unlabeled error rate subject to being empirically at least 3τ-far from the "all-positive" or "all-negative" functions. If the the empirical error rate of this hypothesis h is at most $3\epsilon_1$ we know that its true unlabeled error rate is at most $4\epsilon_1 \leq \tau$, which further implies that either h or $\neg h$ is 2τ close to c^*. However, if the empirical unlabeled error rate of h is greater than $3\epsilon_1$, then we know that the target must be 4τ-close to the all-positive or all-negative function so we simply choose h = "all positive" (this is true since the unlabeled sample was large enough so that $|\hat{d}(f,g) - d(f,g)| \leq \tau$).

So, we have argued that with probability at least $1-2\delta/3$ either h or $\neg h$ is 4τ-close to c^*. We can now just use $O\left(\log_{\left(\frac{1}{\tau}\right)}\left(\frac{1}{\delta}\right)\right)$ labeled examples to determine which case is which (Lemma 2.3.12). This quantity is at most 1 and our error rate is at most ϵ if we set $\tau \leq \epsilon/4$ and τ sufficiently small compared to δ. This completes the proof. ■

The above algorithm assumes one can efficiently pick a random unit-length vector in R^d, but the argument easily goes through even if we do this to only $O(\log 1/\gamma)$ bits of precision.

We now extend the result to the case that we make no margin assumption.

Theorem 2.4.3 *There is a polynomial-time algorithm (in d, b, $1/\epsilon$, and $1/\delta$, where d is the dimension of the space and b is the number of bits per example) to learn a linear separator under the above assumptions, from a polynomial number of unlabeled examples and a single labeled example. Thus, we efficiently* PAC_{unl}*-learn the class of linear separators over $\{-2^b, \ldots, 2^b - 1, 2^b\}^d$ under the agreement notion of compatibility if the distribution D satisfies independence given the label.*

Proof: We begin by drawing a large unlabeled sample S (of size polynomial in d and b). We then compute a linear transformation T that when applied to S has the property that for any hyperplane $w \cdot x = 0$, at least a $1/poly(d,b)$ fraction of $T(S)$ has margin at least $1/poly(d,b)$. We can do this via the Outlier Removal Lemma of [64] and [104]. Specifically, the Outlier Removal Lemma states that given a set of points S, one can algorithmically remove an ϵ' fraction of S and ensure that for the remaining set S', for any vector w, $\max_{x \in S'}(w \cdot x)^2 \leq poly(d, b, 1/\epsilon')\mathbf{E}_{x \in S'}[(w \cdot x)^2]$, where b is the number of bits needed to describe the input points. Given such a set S', one can then use its eigenvectors to compute a standard linear transformation (also described in [64]) $T : R^d \to R^{d'}$, where $d' \leq d$ is the dimension of the subspace spanned by S', such that in the transformed space, for all unit-length w, we have $\mathbf{E}_{x \in T(S')}[(w \cdot x)^2] = 1$. In particular, since the maximum of $(w \cdot x)^2$ is bounded, this implies that for any vector $w \in R^{d'}$, at least an α fraction of points $x \in T(S')$ have margin at least α for some $\alpha \geq 1/poly(b, d, 1/\epsilon')$.

Now, choose $\epsilon' = \epsilon/4$, and let D' be the distribution $\tilde{D}$ restricted to the space spanned by S'. By VC-dimension bounds, $|S| = \tilde{O}(d/\alpha)$ is sufficient so that with high probability, (a) D' has probability mass at least $1 - \epsilon/2$, and (b) the vector $T(c^*)$ has at least an $\alpha/2$ probability mass of $T(D')$ at margin $\geq \alpha$. Thus, the linear transformation T converts the distribution D' into one satisfying the conditions needed for Theorem 2.4.2, and any hypothesis produced with error $\leq \epsilon/2$ on D' will have error at most ϵ on D. So, we simply apply T to D' and run the algorithm for Theorem 2.4.2 to produce a low-error linear separator. ■

Note: We can easily extend our algorithm to the standard co-training setting (where c_1^* can be different from c_2^*) as follows: we repeat the procedure in a symmetric fashion, and then just try all combinations of pairs of functions returned to find one of small unlabeled error rate, not close to "all positive", or "all negative". Finally we use $O\left(\log_{\left(\frac{1}{\epsilon}\right)}\left(\frac{1}{\delta}\right)\right)$ labeled examples to produce a low error hypothesis (and here we use only one part of the example and only one of the functions in the pair).

2.5 Related Models

In this section we discuss a transductive analog of our model, some connections with generative models and other ways of using unlabeled data in Machine Learning, and the relationship between our model and the luckiness framework of [191].

2.5.1 A Transductive Analog of our Model

In *transductive* learning, one is given a fixed set S of examples, of which some small random subset is labeled, and the goal is to predict well on the rest of S. That is, we know which examples we will be tested on up front, and in a sense this a case of learning from a known distribution (the uniform distribution over S). We can also talk about a transductive analog of our inductive model, that incorporates many of the transductive learning methods that have been developed. In order to make use of unlabeled examples, we will again express the relationship we hope the target function has with the data through a compatibility notion χ. However, since in this case the compatibility of a given hypothesis is completely determined by S (which is known), we will not need to require that compatibility be an expectation over unlabeled examples. From the sample complexity point of view we only care about how much labeled data we need, and algorithmically we need to find a highly compatible hypothesis with low error on the labeled data.

Rather than presenting general theorems, we instead focus on the modeling question, and show how a number of existing transductive graph-based learning algorithms can be modeled in our framework. In these methods one usually assumes that there is weighted graph g defined over S, which is given a-priori and encodes the prior knowledge. In the following we denote by W the weighted adjacency matrix of g and by C_S the set of all binary functions over S.

Minimum cut Suppose for $f \in C_S$ we define the incompatibility of f to be the weight of the cut in g determined by f. This is the implicit notion of compatibility considered in [59], and algorithmically the goal is to find the most compatible hypothesis that is correct on the labeled data, which can be solved efficiently using network flow. From a sample-complexity point of view, the number of labeled examples we need is proportional to the VC-dimension of the class of hypotheses that are at least as compatible as the target function. This is known to be $O\left(\frac{k}{\lambda}\right)$ [150, 152], where k is the number of edges cut by c^* and λ is the size of the global minimum cut in the graph. Also note that the Randomized Mincut algorithm (considered by [67]), which is an extension of the basic mincut approach, can be viewed as motivated by a PAC-Bayes sample complexity analysis of the problem.

Normalized Cut For $f \in C_S$ define $size(f)$ to be the weight of the cut in g determined by f, and let $neg(f)$ and $pos(f)$ be the number of points in S on which f predicts negative and positive, respectively. For the normalized cut setting of [140] we can define the incompatibility of $f \in C_S$ to be $\frac{size(f)}{neg(f) \cdot pos(f)}$. This is the penalty function used in [140], and again, algorithmically the goal would be to find a highly compatible hypothesis that is correct on the labeled data. Unfortunately, the corresponding optimization problem is in this case is NP-hard. Still, several approximate solutions have been considered, leading to different semi-supervised learning algorithms. For instance, Joachims [140] considers a spectral relaxation that leads to the "SGT algorithm"; another relaxation based on semidefinite programming is considered in [56].

Harmonic Function We can also model the algorithms introduced in [215] as follows. If we consider f to be a probabilistic prediction function defined over S, then we can define the incompatibility of f to be

$$\sum_{i,j} w_{i,j} \left(f(i) - f(j)\right)^2 = f^T L f,$$

where L is the un-normalized Laplacian of g. Similarly we can model the algorithm introduced by Zhao et al. [213] by using an incompatibility of f given by $f^T \mathcal{L} f$ where $\mathcal{L}$ is the normalized Laplacian of g. More generally, all the Graph Kernel methods can be viewed in our framework if we consider that the incompatibility of f is given by $||f||_K = f^T K f$ where K is a kernel derived from the graph (see for instance [216]).

2.5.2 Connections to Generative Models

It is also interesting to consider how generative models can be fit into our model. As mentioned in Section 2.1, a typical assumption in a generative setting is that D is a mixture with the probability density function $p(x|\theta) = p_0 \cdot p_0(x|\theta_0) + p_1 \cdot p_1(x|\theta_1)$ (see for instance [76, 77, 183]). In other words, the labeled examples are generated according to the following mechanism: a label $y \in \{0, 1\}$ is drawn according to the distribution of classes $\{p_0, p_1\}$ and then a corresponding random feature vector is drawn according to the class-conditional density p_y. The assumption typically used is that the mixture is identifiable. Identifiability ensures that the Bayes optimal decision border $\{x : p_0 \cdot p_0(x|\theta_0) = p_1 \cdot p_1(x|\theta_1)\}$ can be deduced if $p(x|\theta)$ is known, and therefore one can construct an estimate of the Bayes border by using $p(x|\hat{\theta})$ instead of $p(x|\theta)$. Essentially once the decision border is estimated, a small labeled sample suffices to learn (with high confidence and small error) the appropriate class labels associated with the two disjoint regions generated by the estimate of the Bayes decision border. To see how we can incorporate this setting in our model, consider for illustration the setting in [183]; there they assume that $p_0 = p_1$, and that the class conditional densities are d-dimensional Gaussians with unit covariance and unknown mean vectors $\theta_i \in R^d$. The algorithm used is the following: the unknown parameter vector $\theta = (\theta_0, \theta_1)$ is estimated from unlabeled data using a maximum likelihood estimate; this determines a hypothesis which is a linear separator that passes through the point $(\hat{\theta}_0 + \hat{\theta}_1)/2$ and is orthogonal to the vector $\hat{\theta}_1 - \hat{\theta}_0$; finally each of the two decision regions separated by the hyperplane is labeled according to the majority of the labeled examples in the region. Given this setting, a natural notion of compatibility we can consider is the expected log-likelihood function (where the expectation is taken with respect to the unknown distribution specified by θ). Specifically, we can identify a legal hypothesis $f_{\overline{\theta}}$ with the set of parameters $\overline{\theta} = (\overline{\theta}_0, \overline{\theta}_1)$ that determine it, and then we can define $\chi(f_{\overline{\theta}}, D) = \mathbf{E}_{x \in D}[\log(p(x|\overline{\theta}))]$. [183] show that if the unlabeled sample is large enough, then all hypotheses specified by parameters $\overline{\theta}$ which are close enough to θ, will have the property that their empirical compatibilities will be close enough to their true compatibilities. This then implies (together with other observations about Gaussian mixtures) that the maximum likelihood

estimate will be close enough to θ, up to permutations. (This actually motivates χ as a good compatibility function in our model.)

More generally, we can deal with other parametric families using the same compatibility notion; however, we will need to impose constraints on the distributions allowed in order to ensure that the compatibility is actually well defined (the expected log-likelihood is bounded).

As mentioned in Section 2.1, this kind of generative setting is really at the extreme of our model. The assumption that the distribution that generates the data is truly a mixture implies that if we knew the distribution, then there are only two possible concepts left (and this makes the unlabeled data extremely useful).

2.5.3 Connections to the Luckiness Framework

It is worth noticing that there is a strong connection between our approach and the luckiness framework [170, 191]. In both cases, the idea is to define an ordering of hypotheses that depends on the data, in the hope that we will be "lucky" and find that the target function appears early in the ordering. There are two main differences, however. The first is that the luckiness framework (because it was designed for supervised learning only) uses labeled data both for estimating compatibility and for learning: this is a more difficult task, and as a result our bounds on labeled data can be significantly better. For instance, in Example 4 described in Section 2.2, for any non-degenerate distribution, a dataset of $\frac{d}{2}$ pairs can with probability 1 be completely shattered by fully-compatible hypotheses, so the luckiness framework does not help. In contrast, with a larger (unlabeled) sample, one can potentially reduce the space of compatible functions quite significantly, and learn from $o(d)$ or even $O(1)$ labeled examples depending on the distribution – see Section 2.3.2 and Section 2.4. Secondly, the luckiness framework talks about compatibility between a hypothesis and a *sample*, whereas we define compatibility with respect to a distribution. This allows us to talk about the amount of unlabeled data needed to estimate true compatibility. There are also a number of differences at the technical level of the definitions.

2.5.4 Relationship to Other Ways of Using Unlabeled Data for Learning

It is well known that when learning under an unknown distribution, unlabeled data might help somewhat even in the standard discriminative models by allowing one to use both distribution-specific algorithms [53], [144], [194] and/or tighter data dependent sample-complexity bounds [43, 155]. However in all these methods one chooses a class of functions or a prior over functions *before* performing the inference. This does not capture the power of unlabeled data in many of the practical semi-supervised learning methods, where typically one has some idea about what structure of the data tells about the target function, and where the choice of prior can be made more precise after seeing the unlabeled data [62, 141, 158, 184]. Our focus in this chapter has been to provide a unified discriminative framework for reasoning about usefulness of unlabeled data in such settings in which one can analyze both sample complexity and algorithmic results.

Another learning setting where unlabeled data is useful and which has been increasingly popular for the past few years is *Active Learning* [30, 33, 34, 41, 86, 94]. Here, the learning algorithm has both the capability of drawing random unlabeled examples from the underlying distribution and that of asking for the labels of *any* of these examples, and the hope is that a good classifier can be learned with significantly fewer labels by *actively* directing the queries to *informative* examples. Note though that as opposed to the Semi-supervised learning setting, and similarly to the classical supervised learning settings (PAC and Statistical Learning Theory settings) the only prior belief about the learning problem in the Active Learning setting is that the target function (or a good approximation of it) belongs to a given concept

class. Luckily, it turns out that for simple concept classes such as linear separators on the line one can achieve an *exponential* improvement (over the usual supervised learning setting) in the labeled data sample complexity, under no additional assumptions about the learning problem [30, 86].[7] In general, however, for more complicated concept classes, the speed-ups achievable in the active learning setting depend on the match between the distribution over example-label pairs and the hypothesis class, and therefore on the target hypothesis in the class. We discuss all these further as well as our contribution on the topic in Chapter 5.

Finally, in this thesis, we present in the context of learning with kernels and more general similarity functions one other interesting use of unlabeled data in the learning process. While the approach of using unlabeled data in that context does have a similar flavor to the approach in this chapter, the final guarantees and learning procedures are somewhat different from those presented here. In that case the hypothesis space has an infinite capacity before performing the inference. In the training process, in a first stage, we first use unlabeled in order to extract a much smaller set of functions with the property that with high probability the target is well approximated by one the functions in the smaller class. In a second stage we then use labeled examples to learn well. We present this in more details Chapter 3 in Section 3.5.

2.6 Conclusions

Given the easy availability of unlabeled data in many settings, there has been growing interest in methods that try to use such data together with the (more expensive) labeled data for learning. Nonetheless, there has been substantial disagreement and no clear consensus about when unlabeled data helps and by how much. In our work, we have provided a PAC-style model for semi-supervised learning that captures many of the ways unlabeled data is typically used, and provides a very general framework for thinking about this issue. The high level implication of our analysis is that unlabeled data is useful if (a) we have a good notion of compatibility so that the target function indeed has a low unlabeled error rate, (b) the distribution D is *helpful* in the sense that not too many other hypotheses also have a low unlabeled error rate, and (c) we have enough *unlabeled* data to estimate unlabeled error rates well. We then make these statements precise through a series of sample-complexity results, giving bounds as well as identifying the key quantities of interest. In addition, we give several efficient algorithms for learning in this framework. One consequence of our model is that if the target function and data distribution are both well behaved with respect to the compatibility notion, then the sample-size bounds we get can substantially beat what one could hope to achieve using labeled data alone, and we have illustrated this with a number of examples throughout the chapter.

2.6.1 Subsequent Work

Following the initial publication of this work, several authors have used our framework for reasoning about semi-supervised learning, as well as for developing new algorithms and analyses of semi-supervised learning. For example [114, 184, 189] use it in the context of agreement-based multi-view learning for either classification with specific convex loss functions (e.g., hinge loss) or for regression. Sridharan and Kakade [196] use our framework in order to provide a general analysis multi-view learning for a variety of loss functions and learning tasks (classification and regression) along with characterizations of suitable notions of compatibility functions. Parts of this work appear as a book chapter in [82] and as stated in the

[7] For this simple concept class one can achieve a pure exponential improvement [86] in the realizable case, while in the agnostic case the improvement depends upon the noise rate [30].

introduction of that book, our framework can be used to obtain bounds for a number of the semi-supervised learning methods used in the other chapters.

2.6.2 Discussion

Our work brings up a number of open questions, both specific and high-level. One broad category of such questions is for what natural classes C and compatibility notions χ can one provide an efficient algorithm that PAC_{unl}-learns the pair (C, χ): i.e., an algorithm whose running time and sample sizes are polynomial in the bounds of Theorem 2.3.1? For example, a natural question of this form is: can one generalize the algorithm of Section 2.4.1 to allow for irrelevant variables that are neither positive nor negative indicators? That is, suppose we define a "two-sided disjunction" h to be a pair of disjunctions (h_+, h_-) where h is compatible with D iff for all examples x, $h_+(x) = -h_-(x)$ (and let us define $h(x) = h_+(x)$). Can we efficiently learn the class of two-sided disjunctions under this notion of compatibility?

Alternatively, as a different generalization of the problem analyzed in Section 2.4.1, suppose that again every variable is either a positive or negative indicator, but we relax the "margin" condition. In particular, suppose we require that every example x either contain at least 60% of the positive indicators and at most 40% of the negative indicators (for positive examples) or vice versa (for negative examples). Can this class be learned efficiently with bounds comparable to those from Theorem 2.3.1? Along somewhat different lines, can one generalize the algorithm given for Co-Training with linear separators, to assume some condition weaker than independence given the label, while maintaining computational efficiency?

Chapter 3

A General Theory of Learning with Similarity Functions

3.1 Learning with Kernel Functions. Introduction

Kernel functions have become an extremely popular tool in machine learning, with an attractive theory as well [1, 133, 139, 187, 190, 203]. A kernel is a function that takes in two data objects (which could be images, DNA sequences, or points in R^n) and outputs a number, with the property that the function is symmetric and positive-semidefinite. That is, for any kernel K, there must exist an (implicit) mapping ϕ, such that for all inputs x, x' we have $K(x, x') = \langle \phi(x), \phi(x') \rangle$. The kernel is then used inside a "kernelized" learning algorithm such as SVM or kernel-perceptron in place of direct access to the data. Typical kernel functions for structured data include the polynomial kernel $K(x, x') = (1 + x \cdot x')^d$ and the Gaussian kernel $K(x, x') = e^{-||x-x'||^2/2\sigma^2}$, and a number of special-purpose kernels have been developed for sequence data, image data, and other types of data as well [88, 89, 157, 173, 193].

The theory behind kernel functions is based on the fact that many standard algorithms for learning linear separators, such as SVMs [203] and the Perceptron [110] algorithm, can be written so that the only way they interact with their data is via computing dot-products on pairs of examples. Thus, by replacing each invocation of $\langle x, x' \rangle$ with a kernel computation $K(x, x')$, the algorithm behaves exactly as if we had explicitly performed the mapping $\phi(x)$, even though ϕ may be a mapping into a very high-dimensional space. Furthermore, these algorithms have learning guarantees that depend only on the *margin* of the best separator, and not on the dimension of the space in which the data resides [18, 191]. Thus, kernel functions are often viewed as providing much of the power of this implicit high-dimensional space, without paying for it either computationally (because the ϕ mapping is only implicit) or in terms of sample size (if data is indeed well-separated in that space).

While the above theory is quite elegant, it has a few limitations. When designing a kernel function for some learning problem, the intuition employed typically does not involve implicit high-dimensional spaces but rather that a good kernel would be one that serves as a good measure of similarity for the given problem [187]. So, in this sense the theory is not always helpful in providing intuition when selecting or designing a kernel function for a particular learning problem. Additionally, it may be that the most natural similarity function for a given problem is not positive-semidefinite[1], and it could require substantial work, possibly reducing the quality of the function, to coerce it into a "legal" form. Finally, it is a bit unsatisfying for the explanation of the effectiveness of some algorithm to depend on properties of an implicit high-

[1]This is very common in the context of Computational Biology where the most natural measures of alignment between sequences are not legal kernels. For more examples see Section 3.2.

dimensional mapping that one may not even be able to calculate. In particular, the standard theory at first blush has a "something for nothing" feel to it (all the power of the implicit high-dimensional space without having to pay for it) and perhaps there is a more prosaic explanation of what it is that makes a kernel useful for a given learning problem. For these reasons, it would be helpful to have a theory that was in terms of more tangible quantities.

In this chapter, we develop a theory of learning with similarity functions that addresses a number of these issues. In particular, we define a notion of what it means for a pairwise function $K(x, x')$ to be a "good similarity function" for a given learning problem that (a) does not require the notion of an implicit space and allows for functions that are not positive semi-definite, (b) we can show is sufficient to be used for learning, and (c) *strictly* generalizes the standard theory in that a good kernel in the usual sense (large margin in the implicit ϕ-space) will also satisfy our definition of a good similarity function. In this way, we provide the first theory that describes the effectiveness of a given kernel (or more general similarity function) in terms of natural similarity-based properties.

More generally, our framework provides a formal way to analyze properties of a similarity function that make it sufficient for learning, as well as what algorithms are suited for a given property. Note that while our work is motivated by extending the standard large-margin notion of a good kernel function, we expect one can use this framework to analyze other, not necessarily comparable, properties that are sufficient for learning as well. In fact, recent work along these lines is given in [208].

Structure of this chapter: We start with background and notation in Section 3.2. We the present a first notion of a good similarity function in Section 3.3 and analyze its relationship with the usual notion of a good kernel function. (These results appear in [24] and [38].) In section 3.4 we present a slightly different and broader notion that we show provides even better kernels to similarity translation; in Section 3.4.3 we give a separation result, showing that this new notion is *strictly more general* than the notion of a large margin kernel. (These results appear in [39].)

3.2 Background and Notation

We consider a learning problem specified as follows. We are given access to labeled examples (x, y) drawn from some distribution P over $X \times \{-1, 1\}$, where X is an abstract instance space. The objective of a learning algorithm is to produce a classification function $g : X \to \{-1, 1\}$ whose error rate $\Pr_{(x,y)\sim P}[g(x) \neq y]$ is low. We will consider learning algorithms that only access the points x through a pairwise similarity function $K(x, x')$ mapping pairs of points to numbers in the range $[-1, 1]$. Specifically,

Definition 3.2.1 *A* similarity function *over X is any pairwise function $K : X \times X \to [-1, 1]$. We say that K is a symmetric similarity function if $K(x, x') = K(x', x)$ for all x, x'.*

A similarity function K is a valid (or legal) kernel function if it is positive-semidefinite, i.e. there exists a function ϕ from the instance space X into some (implicit) Hilbert "ϕ-space" such that

$$K(x, x') = \langle \phi(x), \phi(x') \rangle.$$

See, e.g., Smola and Schölkopf [186] for a discussion on conditions for a mapping being a kernel function. Throughout this chapter, and without loss of generality, we will only consider kernels such that $K(x, x) \leq 1$ for all $x \in \mathcal{X}$. Any kernel K can be converted into this form by, for instance, defining

$$\tilde{K}(x, x') = K(x, x')/\sqrt{K(x, x)K(x', x')}.$$

We say that K is (ϵ, γ)-*kernel good* for a given learning problem P if there exists a vector β in the ϕ-space that has error ϵ at margin γ; for simplicity we consider only separators through the origin. Specifically:[2]

Definition 3.2.2 *K is* (ϵ, γ)-kernel good *if there exists a vector β, $||\beta|| \leq 1$ such that*

$$\Pr_{(x,y)\sim P}[y\langle\phi(x), \beta\rangle \geq \gamma] \geq 1 - \epsilon.$$

We say that K is γ-*kernel good* if it is (ϵ, γ)-*kernel good* for $\epsilon = 0$; i.e., it has zero error at margin γ.

Given a kernel that is (ϵ, γ)-kernel-good for some learning problem P, a predictor with error rate at most $\epsilon + \epsilon_{\text{acc}}$ can be learned (with high probability) from a sample of[3] $\tilde{\mathcal{O}}\big((\epsilon + \epsilon_{\text{acc}})/(\gamma^2\epsilon_{\text{acc}}^2)\big)$ examples (drawn independently from the source distribution) by minimizing the number of margin γ violations on the sample [168]. However, minimizing the number of margin violations on the sample is a difficult optimization problem [18, 20]. Instead, it is common to minimize the so-called *hinge loss* relative to a margin.

Definition 3.2.3 *We say that K is* (ϵ, γ)-*kernel good* in hinge-loss *if there exists a vector β, $||\beta|| \leq 1$ such that*

$$\mathbf{E}_{(x,y)\sim P}[[1 - y\langle\beta, \phi(x)\rangle/\gamma]_+] \leq \epsilon,$$

where $[1 - z]_+ = \max(1 - z, 0)$ is the hinge loss.

Given a kernel that is (ϵ, γ)-kernel-good in hinge-loss, a predictor with error rate at most $\epsilon + \epsilon_{\text{acc}}$ can be efficiently learned (with high probability) from a sample of $\mathcal{O}\big(1/(\gamma^2\epsilon_{\text{acc}}^2)\big)$ examples by minimizing the average hinge loss relative to margin γ on the sample [43].

We end this section by noting that a general similarity function might not be a legal (valid) kernel. To illustrate this we provide a few examples in the following.

Examples of similarity functions which are not legal kernel functions. As a simple example, let us consider a document classification task and let us assume we have a similarity function K such that two documents have similarity 1 if they have either an author in common or a keyword in common, and similarity 0 otherwise. Then we could have three documents A, B, and C, such that $K(A, B) = 1$ because A and B have an author in common, $K(B, C) = 1$ because B and C have a keyword in common, but $K(A, C) = 0$ because A and C have neither an author nor a keyword in common (and $K(A, A) = K(B, B) = K(C, C) = 1$). On the other hand, a kernel requires that if $\phi(A)$ and $\phi(B)$ are of unit length and $\langle\phi(A), \phi(B)\rangle = 1$, then $\phi(A) = \phi(B)$, so this could not happen if K was a valid kernel.

Similarity functions that are not legal kernels are common in the context of computational biology [160]; standard examples include various measures of alignment between sequences such as BLAST scores for protein sequences or for DNA. Finally, one other natural example of a similarity function that might not be a legal kernel (and which might not be even symmetric) is the following: consider a transductive setting (where we have all the points we want to classify in advance) and assume we have a base distance function $d(x, x')$. Let us define $K(x, x')$ as the percentile rank of x' in distance to x (i.e., $K(x, x') = \Pr[d(x, x') \leq d(x, x'')]$; then clearly K might not be a legal kernel since in fact it might not even be a symmetric similarity function.

Of course, one could modify such a function to be positive semidefinite, e.g., by blowing up the diagonal or by using other related methods suggested in the literature [166], but none of these methods have a formal guarantee on the final generalization bound (and these methods might significantly decrease the "dynamic range" of K and yield a very small margin).

[2] Note that we are distinguishing between what is needed for a similarity function to be a valid or legal kernel function (symmetric and positive semidefinite) and what is needed to be a *good* kernel function for a learning problem (large margin).

[3] The $\tilde{\mathcal{O}}(\cdot)$ notations hide logarithmic factors in the arguments, and in the failure probability.

3.3 Learning with More General Similarity Functions: A First Attempt

Our goal is to describe "goodness" properties that are sufficient for a similarity function to allow one to learn well that ideally are intuitive and subsume the usual notion of good kernel function. Note that as with the theory of kernel functions [186], "goodness" is with respect to a given learning problem P, and *not* with respect to a class of target functions as in the PAC framework [149, 201].

We start by presenting here the notion of good similarity functions introduced in [24] and further analyzed in [195] and [38], which throughout the chapter we call the Balcan - Blum'06 definition. We begin with a definition (Definition 3.3.1) that is especially intuitive and allows for learning via a very simple algorithm, but is not broad enough to include all kernel functions that induce large-margin separators. We then broaden this notion to the main definition in [24] (Definition 3.3.5) that requires a more involved algorithm to learn, but is now able to capture all functions satisfying the usual notion of a good kernel function. Specifically, we show that if K is a similarity function satisfying Definition 3.3.5 then one can algorithmically perform a simple, *explicit* transformation of the data under which there is a low-error large-margin separator. We also consider variations on this definition (e.g., Definition 3.3.6) that produce better guarantees on the quality of the final hypothesis when combined with existing learning algorithms.

A similarity function K satisfying the Balcan - Blum'06 definition, but that is not positive semi-definite, is not necessarily guaranteed to work well when used directly in standard learning algorithms such as SVM or the Perceptron algorithm[4]. Instead, what we show is that such a similarity function can be employed in the following two-stage algorithm. First, re-represent that data by performing what might be called an "empirical similarity map": selecting a subset of data points as landmarks, and then representing each data point using the similarities to those landmarks. Then, use standard methods to find a large-margin linear separator in the new space. One property of this approach is that it allows for the use of a broader class of learning algorithms since one does not need the algorithm used in the second step to be "kernalizable". In fact, the work in this chapter is motivated by work on a re-representation method that algorithmically transforms a kernel-based learning problem (with a valid positive-semidefinite kernel) to an explicit low-dimensional learning problem [31]. (We present this Chapter 6.)

Deterministic Labels: For simplicity in presentation, for most of this section we will consider only learning problems where the label y is a deterministic function of x. For such learning problems, we can use $y(x)$ to denote the label of point x, and we will use $x \sim P$ as shorthand for $(x, y(x)) \sim P$. We will return to learning problems where the label y may be a probabilistic function of x in Section 3.3.5.

3.3.1 Sufficient Conditions for Learning with Similarity Functions

We now provide a series of sufficient conditions for a similarity function to be useful for learning, leading to the notions given in Definitions 3.3.5 and 3.3.6.

3.3.2 Simple Sufficient Conditions

We begin with our first and simplest notion of "good similarity function" that is intuitive and yields an immediate learning algorithm, but which is not broad enough to capture all good kernel functions. Nonetheless, it provides a convenient starting point. This definition says that K is a good similarity function for a learning problem P if most examples x (at least a $1 - \epsilon$ probability mass) are on average at least γ more similar to random examples x' of the *same* label than they are to random examples x' of the opposite label. Formally,

[4]However, as we will see in Section 3.3.5, if the function *is* positive semi-definite and if it is good in the Balcan - Blum'06 sense [24, 38], or in the Balcan - Blum - Srebro'08 sense [39], then we can show it is good as a kernel as well.

Definition 3.3.1 *K is a* **strongly (ϵ, γ)-good similarity function** *for a learning problem P if at least a $1 - \epsilon$ probability mass of examples x satisfy:*

$$\mathbf{E}_{x' \sim P}[K(x, x')|y(x) = y(x')] \geq \mathbf{E}_{x' \sim P}[K(x, x')|y(x) \neq y(x')] + \gamma. \tag{3.1}$$

For example, suppose all positive examples have similarity at least 0.2 with each other, and all negative examples have similarity at least 0.2 with each other, but positive and negative examples have similarities distributed uniformly at random in $[-1, 1]$. Then, this would satisfy Definition 3.3.1 for $\gamma = 0.2$ and $\epsilon = 0$. Note that with high probability this would not be positive semidefinite.[5]

Definition 3.3.1 captures an intuitive notion of what one might want in a similarity function. In addition, if a similarity function K satisfies Definition 3.3.1 then it suggests a simple, natural learning algorithm: draw a sufficiently large set S^+ of positive examples and set S^- of negative examples, and then output the prediction rule that classifies a new example x as positive if it is on average more similar to points in S^+ than to points in S^-, and negative otherwise. Formally:

Theorem 3.3.1 *If K is strongly (ϵ, γ)-good, then a set S^+ of $(16/\gamma^2)\ln(2/\delta)$ positive examples and a set S^- of $(16/\gamma^2)\ln(2/\delta)$ negative examples are sufficient so that with probability $\geq 1 - \delta$, the above algorithm produces a classifier with error at most $\epsilon + \delta$.*

Proof: Let Good be the set of x satisfying

$$\mathbf{E}_{x' \sim P}[K(x, x')|y(x) = y(x')] \geq \mathbf{E}_{x' \sim P}[K(x, x')|y(x) \neq y(x')] + \gamma.$$

So, by assumption, $\Pr_{x \sim P}[x \in \text{Good}] \geq 1 - \epsilon$. Now, fix $x \in \text{Good}$. Since $K(x, x') \in [-1, 1]$, by Hoeffding bounds we have that over the random draw of the sample S^+,

$$\Pr\left(\left|\mathbf{E}_{x' \in S^+}[K(x, x')] - \mathbf{E}_{x' \sim P}[K(x, x')|y(x') = 1]\right| \geq \gamma/2\right) \leq 2e^{-2|S^+|\gamma^2/16},$$

and similarly for S^-. By our choice of $|S^+|$ and $|S^-|$, each of these probabilities is at most $\delta^2/2$.

So, for any given $x \in \text{Good}$, there is at most a δ^2 probability of error over the draw of S^+ and S^-. Since this is true for any $x \in \text{Good}$, it implies that the *expected* error of this procedure, over $x \in \text{Good}$, is at most δ^2, which by Markov's inequality implies that there is at most a δ probability that the error rate over Good is more than δ. Adding in the ϵ probability mass of points not in Good yields the theorem. ■

Before going to our main notion note that Definition 3.3.1 requires that almost all of the points (at least a $1 - \epsilon$ fraction) be on average more similar to random points of the same label than to random points of the other label. A weaker notion would be simply to require that two random points of the same label be on average more similar than two random points of different labels. For instance, one could consider the following generalization of Definition 3.3.1:

Definition 3.3.2 *K is a* **weakly γ-good similarity function** *for a learning problem P if:*

$$\mathbf{E}_{x,x' \sim P}[K(x, x')|y(x) = y(x')] \quad \geq \quad \mathbf{E}_{x,x' \sim P}[K(x, x')|y(x) \neq y(x')] + \gamma. \tag{3.2}$$

While Definition 3.3.2 still captures a natural intuitive notion of what one might want in a similarity function, it is not powerful enough to imply *strong* learning unless γ is quite large. For example, suppose the instance space is R^2 and that the similarity measure K we are considering is just the product of the first coordinates (i.e., dot-product but ignoring the second coordinate). Assume the distribution is half positive

[5] In particular, if the domain is large enough, then with high probability there would exist negative example A and positive examples B, C such that $K(A, B)$ is close to 1 (so they are nearly identical as vectors), $K(A, C)$ is close to -1 (so they are nearly opposite as vectors), and yet $K(B, C) \geq 0.2$ (their vectors form an acute angle).

and half negative, and that 75% of the positive examples are at position $(1,1)$ and 25% are at position $(-1,1)$, and 75% of the negative examples are at position $(-1,-1)$ and 25% are at position $(1,-1)$. Then K is a weakly γ-good similarity function for $\gamma = 1/2$, but the best accuracy one can hope for using K is 75% because that is the accuracy of the Bayes-optimal predictor given only the first coordinate.

We can however show that for any $\gamma > 0$, Definition 3.3.2 is enough to imply weak learning [188]. In particular, the following simple algorithm is sufficient to weak learn. First, determine if the distribution is noticeably skewed towards positive or negative examples: if so, weak-learning is immediate (output all-positive or all-negative respectively). Otherwise, draw a sufficiently large set S^+ of positive examples and set S^- of negative examples. Then, for each x, consider $\tilde{\gamma}(x) = \frac{1}{2}\left[\mathbf{E}_{x' \in S^+}[K(x,x')] - \mathbf{E}_{x' \in S^-}[K(x,x')]\right]$. Finally, to classify x, use the following probabilistic prediction rule: classify x as positive with probability $\frac{1+\tilde{\gamma}(x)}{2}$ and as negative with probability $\frac{1-\tilde{\gamma}(x)}{2}$. (Notice that $\tilde{\gamma}(x) \in [-1,1]$ and so our algorithm is well defined.) We can then prove the following result:

Theorem 3.3.2 *If K is a weakly γ-good similarity function, then with probability at least $1-\delta$, the above algorithm using sets S^+, S^- of size $\frac{64}{\gamma^2}\ln\left(\frac{64}{\gamma\delta}\right)$ yields a classifier with error at most $\frac{1}{2} - \frac{3\gamma}{128}$.*

Proof: First, we assume the algorithm initially draws a sufficiently large sample such that if the distribution is skewed with probability mass greater than $\frac{1}{2} + \alpha$ on positives or negatives for $\alpha = \frac{\gamma}{32}$, then with probability at least $1 - \delta/2$ the algorithm notices the bias and weak-learns immediately (and if the distribution is less skewed than $\frac{1}{2} \pm \frac{3\gamma}{128}$, with probability $1 - \delta/2$ it does not incorrectly halt in this step). In the following, then, we may assume the distribution P is less than $(\frac{1}{2} + \alpha)$-skewed, and let us define P' to be P reweighted to have probability mass exactly $1/2$ on positive and negative examples. Thus, Definition 3.3.2 is satisfied for P' with margin at least $\gamma - 4\alpha$.

For each x define $\gamma(x)$ as $\frac{1}{2}\mathbf{E}_{x'}[K(x,x')|y(x') = 1] - \frac{1}{2}\mathbf{E}_{x'}[K(x,x')|y(x') = -1]$ and notice that Definition 3.3.2 implies that $\mathbf{E}_{x \sim P'}\left[y(x)\gamma(x)\right] \geq \gamma/2 - 2\alpha$. Consider now the probabilistic prediction function g defined as $g(x) = 1$ with probability $\frac{1+\gamma(x)}{2}$ and $g(x) = -1$ with probability $\frac{1-\gamma(x)}{2}$. We clearly have that for a fixed x,

$$\Pr_g(g(x) \neq y(x)) = \frac{y(x)(y(x) - \gamma(x))}{2},$$

which then implies that $\Pr_{x \sim P', g}(g(x) \neq y(x)) \leq \frac{1}{2} - \frac{1}{4}\gamma - \alpha$. Now notice that in our algorithm we do not use $\gamma(x)$ but an estimate of it $\tilde{\gamma}(x)$, and so the last step of the proof is to argue that this is good enough. To see this, notice first that d is large enough so that for any fixed x we have

$$\Pr_{S^+, S^-}\left(|\gamma(x) - \tilde{\gamma}(x)| \geq \frac{\gamma}{4} - 2\alpha\right) \leq \frac{\gamma\delta}{32}.$$

This implies

$$\Pr_{x \sim P'}\left(\Pr_{S^+, S^-}\left(|\gamma(x) - \tilde{\gamma}(x)| \geq \frac{\gamma}{4} - 2\alpha\right)\right) \leq \frac{\gamma\delta}{32},$$

so

$$\Pr_{S^+, S^-}\left(\Pr_{x \sim P}\left(|\gamma(x) - \tilde{\gamma}(x)| \geq \frac{\gamma}{4} - 2\alpha\right) \geq \frac{\gamma}{16}\right) \leq \delta/2.$$

This further implies that with probability at least $1 - \delta/2$ we have $\mathbf{E}_{x \sim P'}\left[y(x)\tilde{\gamma}(x)\right] \geq \left(1 - \frac{\gamma}{16}\right)\frac{\gamma}{4} - 2\frac{\gamma}{16} \geq \frac{7\gamma}{64}$. Finally using a reasoning similar to the one above (concerning the probabilistic prediction function based on $\gamma(x)$), we obtain that with probability at least $1 - \delta/2$ the error of the probabilistic classifier based on $\tilde{\gamma}(x)$ is at most $\frac{1}{2} - \frac{7\gamma}{128}$ on P', which implies the error over P is at most $\frac{1}{2} - \frac{7\gamma}{128} + \alpha = \frac{1}{2} - \frac{3\gamma}{128}$.

■

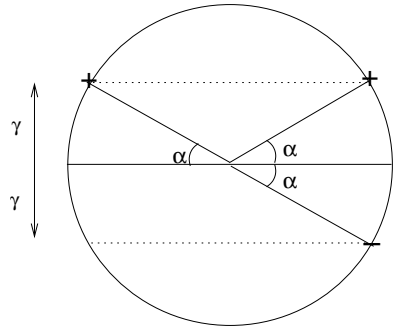

Figure 3.1: Positives are split equally among upper-left and upper-right. Negatives are all in the lower-right. For $\alpha = 30^o$ (so $\gamma = 1/2$) a large fraction of the positive examples (namely the 50% in the upper-right) have a higher dot-product with negative examples ($\frac{1}{2}$) than with a random positive example ($\frac{1}{2} \cdot 1 + \frac{1}{2}(-\frac{1}{2}) = \frac{1}{4}$). However, if we assign the positives in the upper-left a weight of 0, those in the upper-right a weight of 1, and assign negatives a weight of $\frac{1}{2}$, then all examples have higher average *weighted* similarity to those of the same label than to those of the opposite label, by a gap of $\frac{1}{4}$.

Returning to Definition 3.3.1, Theorem 3.3.1 implies that if K is a strongly (ϵ, γ)-good similarity function for small ϵ and not-too-small γ, then it can be used in a natural way for learning. However, Definition 3.3.1 is not sufficient to capture all good kernel functions. In particular, Figure 3.1 gives a simple example in $\mathcal{R}^2$ where the standard kernel $K(x, x') = \langle x, x' \rangle$ has a large margin separator (margin of $1/2$) and yet does not satisfy Definition 3.3.1, even for $\gamma = 0$ and $\epsilon = 0.24$.

Notice, however, that if in Figure 3.1 we simply ignored the positive examples in the upper-left when choosing x', and down-weighted the negative examples a bit, then we would be fine. This then motivates the following intermediate notion of a similarity function K being good under a weighting function w over the input space that can downweight certain portions of that space.

Definition 3.3.3 *A similarity function K together with a bounded weighting function w over X (specifically, $w(x') \in [0, 1]$ for all $x' \in X$) is a* **strongly (ϵ, γ)-good weighted similarity function** *for a learning problem P if at least a $1 - \epsilon$ probability mass of examples x satisfy:*

$$\mathbf{E}_{x' \sim P}[w(x')K(x, x')|y(x) = y(x')] \quad \geq \quad \mathbf{E}_{x' \sim P}[w(x')K(x, x')|y(x) \neq y(x')] + \gamma. \tag{3.3}$$

We can view Definition 3.3.3 intuitively as saying that we only require most examples be substantially more similar on average to *representative* points of the same class than to *representative* points of the opposite class, where "representativeness" is a score in $[0, 1]$ given by the weighting function w. A pair (K, w) satisfying Definition 3.3.3 can be used in exactly the same way as a similarity function K satisfying Definition 3.3.1, with the exact same proof used in Theorem 3.3.1 (except now we view $w(y)K(x, x')$ as the bounded random variable we plug into Hoeffding bounds).

3.3.3 Main Balcan - Blum'06 Conditions

Unfortunately, Definition 3.3.3 requires the designer to construct both K and w, rather than just K. We now weaken the requirement to ask only that such a w *exist*, in Definition 3.3.4 below:

Definition 3.3.4 (Main Balcan - Blum'06 Definition, Balanced Version) *A similarity function K is an (ϵ, γ)-good similarity function for a learning problem P if there* exists *a bounded weighting function w over X ($w(x') \in [0, 1]$ for all $x' \in X$) such that at least a $1 - \epsilon$ probability mass of examples x satisfy:*

$$\mathbf{E}_{x' \sim P}[w(x')K(x, x')|y(x) = y(x')] \quad \geq \quad \mathbf{E}_{x' \sim P}[w(x')K(x, x')|y(x) \neq y(x')] + \gamma. \tag{3.4}$$

As mentioned above, the key difference is that whereas in Definition 3.3.3 one needs the designer to construct both the similarity function K *and* the weighting function w, in Definition 3.3.4 we only require that such a w *exist*, but it need not be known a-priori. That is, we ask only that there exist a large probability mass of "representative" points (a weighting scheme) satisfying Definition 3.3.3, but the designer need not know in advance what that weighting scheme should be.

Definition 3.3.4 can also be stated as requiring that, for at least $1-\epsilon$ of the examples, the *classification margin*

$$\begin{aligned}\mathbf{E}_{x'\sim P}\left[w(x')K(x,x')|y(x)=y(x')\right] - \mathbf{E}_{x'\sim P}\left[w(x')K(x,x')|y(x)\neq y(x')\right] \\ = y(x)\mathbf{E}_{x'\sim P}\left[w(x')y(x')K(x,x')/P(y(x'))\right]\end{aligned} \tag{3.5}$$

be at least γ, where $P(y(x'))$ is the marginal probability under P, i.e. the prior, of the label associated with x'. We will find it more convenient in the following to analyze instead a slight variant, dropping the factor $1/P(y(x'))$ from the classification margin (3.5)—see Definition 3.3.5 in the next Section. Any similarity function satisfying Definition 3.3.5 also satisfies Definition 3.3.4 (by simply multiplying $w(x')$ by $P(y(x'))$). However, the learning algorithm using Definition 3.3.5 is slightly simpler, and the connection to kernels is a bit more direct.

We are now ready to present the main sufficient condition for learning with similarity functions in [24]. This is essentially a restatement of Definition 3.3.4, dropping the normalization by the label "priors" as discussed at the end of the preceding Section.

Definition 3.3.5 (Main Balcan - Blum'06 Definition, Margin Violations) *A similarity function K is an* (ϵ,γ)**-good similarity function** *for a learning problem P if there* exists *a bounded weighting function w over X ($w(x')\in[0,1]$ for all $x'\in X$) such that at least a $1-\epsilon$ probability mass of examples x satisfy:*

$$\mathbf{E}_{x'\sim P}[y(x)y(x')w(x')K(x,x')] \quad \geq \quad \gamma. \tag{3.6}$$

We would like to establish that the above condition is indeed sufficient for learning. I.e. that given an (ϵ,γ)-good similarity function K for some learning problem P, and a sufficiently large labeled sample drawn from P, one can obtain (with high probability) a predictor with error rate arbitrarily close to ϵ. To do so, we will show how to use an (ϵ,γ)-good similarity function K, and a sample S drawn from P, in order to construct (with high probability) an explicit mapping $\phi^S: X\to R^d$ for all points in X (not only points in the sample S), such that the mapped data $(\phi^S(x),y(x))$, where $x\sim P$, is separated with error close to ϵ (and in fact also with large margin) in the low-dimensional linear space R^d (Theorem 3.3.3 below). We thereby convert the learning problem into a standard problem of learning a linear separator, and can use standard results on learnability of linear separators to establish learnability of our original learning problem, and even provide learning guarantees.

What we are doing is actually showing how to use a good similarity function K (that is not necessarily a valid kernel) and a sample S drawn from P to construct a valid kernel $\tilde{K}^S$, given by $\tilde{K}^S(x,x') = \left\langle \phi^S(x), \phi^S(x')\right\rangle$, that is kernel-good and can thus be used for learning (In Section 3.3.5 we show that if K is already a valid kernel, a transformation is not necessary as K itself is kernel-good). We are therefore leveraging here the established theory of linear, or kernel, learning in order to obtain learning guarantees for similarity measures that are not valid kernels.

Interestingly, in Section 3.3.5 we also show that any kernel that is kernel-good is also a good similarity function (though with some degradation of parameters). The suggested notion of "goodness" (Definition 3.3.5) thus encompasses the standard notion of kernel-goodness, and extends it also to non-positive-definite similarity functions.

Theorem 3.3.3 *Let K be an (ϵ,γ)-good similarity function for a learning problem P. For any $\delta>0$, let $S=\{\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_d\}$ be a sample of size $d=8\log(1/\delta)/\gamma^2$ drawn from P. Consider the mapping*

$\phi^S : X \to R^d$ *defined as follows:* ${\phi^S}_i(x) = \frac{K(x,\tilde{x}_i)}{\sqrt{d}}$, $i \in \{1,\ldots,d\}$. *With probability at least* $1-\delta$ *over the random sample* S*, the induced distribution* $\phi^S(P)$ *in* R^d *has a separator of error at most* $\epsilon+\delta$ *at margin at least* $\gamma/2$.

Proof: Let $w : X \to [0,1]$ be the weighting function achieving (3.6) of Definition 3.3.5. Consider the linear separator $\beta \in R^d$, given by $\beta_i = \frac{y(\tilde{x}_i)w(\tilde{x}_i)}{\sqrt{d}}$; note that $||\beta|| \leq 1$. We have, for any $x, y(x)$:

$$y(x)\big\langle \beta, \phi^S(x)\big\rangle = \frac{1}{d}\sum_{i=1}^{d} y(x)y(\tilde{x}_i)w(\tilde{x}_i)K(x,\tilde{x}_i) \tag{3.7}$$

The right hand side of the (3.7) is an empirical average of $-1 \leq y(x)y(x')w(x')K(x,x') \leq 1$, and so by Hoeffding's inequality, for any x, and with probability at least $1-\delta^2$ over the choice of S, we have:

$$\frac{1}{d}\sum_{i=1}^{d} y(x)y(\tilde{x}_i)w(\tilde{x}_i)K(x,\tilde{x}_i) \geq \mathbf{E}_{x'\sim P}\big[y(x)y(x')w(x')K(x,x')\big] - \sqrt{\frac{2\log(\frac{1}{\delta^2})}{d}} \tag{3.8}$$

Since the above holds for any x with probability at least $1-\delta^2$ over the choice of S, it also holds with probability at least $1-\delta^2$ over the choice of x and S. We can write this as:

$$\mathbf{E}_{S\sim P^d}\left[\Pr_{x\sim P}(\text{ violation })\right] \leq \delta^2 \tag{3.9}$$

where "violation" refers to violating (3.8). Applying Markov's inequality we get that with probability at least $1-\delta$ over the choice of S, at most δ fraction of points violate (3.8). Recalling Definition 3.3.5, at most an additional ϵ fraction of the points violate (3.6). But for the remaining $1-\epsilon-\delta$ fraction of the points, for which both (3.8) and (3.6) hold, we have: $y(x)\big\langle \beta, \phi^S(x)\big\rangle \geq \gamma - \sqrt{\frac{2\log(\frac{1}{\delta^2})}{d}} = \gamma/2$, where to get the last inequality we use $d = 8\log(1/\delta)/\gamma^2$. ■

We can learn a predictor with error rate at most $\epsilon+\epsilon_{\text{acc}}$ using an (ϵ,γ)-good similarity function K as follows. We first draw from P a sample $S = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_d\}$ of size $d = (4/\gamma)^2\ln(4/\delta\epsilon_{\text{acc}})$ and construct the mapping $\phi^S : X \to R^d$ defined as follows: ${\phi^S}_i(x) = \frac{K(x,\tilde{x}_i)}{\sqrt{d}}$, $i \in \{1,\ldots,d\}$. The guarantee we have is that with probability at least $1-\delta$ over the random sample S, the induced distribution $\phi^S(P)$ in R^d, has a separator of error at most $\epsilon+\epsilon_{\text{acc}}/2$ at margin at least $\gamma/2$. So, to learn well, we then draw a new, fresh sample, map it into the transformed space using ϕ^S, and then learn a linear separator in transformed space using ϕ^S, the new space. The number of landmarks is dominated by the $\tilde{\mathcal{O}}\big((\epsilon+\epsilon_{\text{acc}})d/\epsilon_{\text{acc}}^2\big) = \tilde{\mathcal{O}}\big((\epsilon+\epsilon_{\text{acc}})/(\gamma^2\epsilon_{\text{acc}}^2)\big)$ sample complexity of the linear learning, yielding the same order sample complexity as in the kernel-case for achieving error at most $\epsilon+\epsilon_{\text{acc}}$: $\tilde{\mathcal{O}}\big((\epsilon+\epsilon_{\text{acc}})/(\gamma^2\epsilon_{\text{acc}}^2)\big)$.

Unfortunately, the above sample complexity refers to learning by finding a linear separator minimizing the error over the training sample. This minimization problem is NP-hard [18], and even NP-hard to approximate [20]. In certain special cases, such as if the induced distribution $\phi^S(P)$ happens to be log-concave, efficient learning algorithms exist [145]. However, as discussed earlier, in the more typical case, one minimizes the *hinge-loss* instead of the number of errors. We therefore consider also a modification of Definition 3.3.5 that captures the notion of good similarity functions for the SVM and Perceptron algorithms as follows:

Definition 3.3.6 (Main Balcan - Blum'06 Definition, Hinge Loss) *A similarity function K is an (ϵ, γ)-good similarity function in hinge loss for a learning problem P if there exists a weighting function $w(x') \in [0,1]$ for all $x' \in X$ such that*

$$\mathbf{E}_x\Big[\,[1 - y(x)g(x)/\gamma]_+\,\Big] \leq \epsilon, \tag{3.10}$$

where $g(x) = \mathbf{E}_{x' \sim P}[y(x')w(x')K(x,x')]$ is the similarity-based prediction made using $w()$, and recall that $[1-z]_+ = \max(0, 1-z)$ is the hinge-loss.

In other words, we are asking: on average, by how much, in units of γ, would a random example x fail to satisfy the desired γ separation between the weighted similarity to examples of its own label and the weighted similarity to examples of the other label.

Similarly to Theorem 3.3.3, we have:

Theorem 3.3.4 *Let K be an (ϵ, γ)-good similarity function in hinge loss for a learning problem P. For any $\epsilon_1 > 0$ and $0 < \delta < \gamma\epsilon_1/4$ let $S = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_d\}$ be a sample of size $d = 16\log(1/\delta)/(\epsilon_1\gamma)^2$ drawn from P. With probability at least $1-\delta$ over the random sample S, the induced distribution $\phi^S(P)$ in R^d, for ϕ^S as defined in Theorem 3.3.3, has a separator achieving hinge-loss at most $\epsilon + \epsilon_1$ at margin at least γ.*

Proof: Let $w : X \to [0,1]$ be the weighting function achieving an expected hinge loss of at most ϵ at margin γ, and denote $g(x) = \mathbf{E}_{x' \sim P}[y(x')w(x')K(x,x')]$. Defining β as in Theorem 3.3.3 and following the same arguments we have that with probability at least $1-\delta$ over the choice of S, at most δ fraction of the points x violate 3.8. We will only consider such samples S. For those points that do not violate (3.8) we have:

$$[1 - y(x)\big\langle \beta, \phi^S(x)\big\rangle/\gamma]_+ \leq [1 - y(x)g(x)/\gamma]_+ \frac{1}{\gamma}\sqrt{\frac{2\log(\frac{1}{\delta^2})}{d}} \leq [1 - y(x)g(x)/\gamma]_+ + \epsilon_1/2 \tag{3.11}$$

For points that do violate (3.8), we will just bound the hinge loss by the maximum possible hinge-loss:

$$[1 - y(x)\big\langle \beta, \phi^S(x)\big\rangle/\gamma]_+ \leq 1 + \max_x \big|y(x)||\beta||||\phi^S(x)|||\,/\gamma \leq 1 + 1/\gamma \leq 2/\gamma \tag{3.12}$$

Combining these two cases we can bound the expected hinge-loss at margin γ:

$$\begin{aligned}
\mathbf{E}_{x\sim P}\big[[1 - y(x)\big\langle \beta, \phi^S(x)\big\rangle/\gamma]_+\big] &\leq \mathbf{E}_{x\sim P}[[1 - y(x)g(x)/\gamma]_+] + \epsilon_1/2 + \Pr(\text{violation}\,) \cdot (2/\gamma) \\
&\leq \mathbf{E}_{x\sim P}[[1 - y(x)g(x)/\gamma]_+] + \epsilon_1/2 + 2\delta/\gamma \\
&\leq \mathbf{E}_{x\sim P}[[1 - y(x)g(x)/\gamma]_+] + \epsilon_1,
\end{aligned} \tag{3.13}$$

where the last inequality follows from $\delta < \epsilon_1\gamma/4$. ■

We can learn a predictor with error rate at most $\epsilon + \epsilon_{\text{acc}}$ using an (ϵ, γ)-good similarity function K as follows. We first draw from P a sample $S = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_d\}$ of size $d = 16\log(2/\delta)/(\epsilon_{\text{acc}}\gamma)^2$ and construct the mapping $\phi^S : X \to R^d$ defined as follows: ${\phi^S}_i(x) = \frac{K(x,\tilde{x}_i)}{\sqrt{d}}$, $i \in \{1, \ldots, d\}$. The guarantee we have is that with probability at least $1-\delta$ over the random sample S, the induced distribution $\phi^S(P)$ in R^d, has a separator achieving hinge-loss at most $\epsilon + \epsilon_{\text{acc}}/2$ at margin γ. So, to learn well, we can then use an SVM solver in the ϕ^S-space to obtain (with probability at least $1-2\delta$) a predictor with error rate $\epsilon + \epsilon_{\text{acc}}$ using $\tilde{\mathcal{O}}\big(1/(\gamma^2\epsilon_{\text{acc}}^2)\big)$ examples, and time polynomial in $1/\gamma$,$1/\epsilon_{\text{acc}}$ and $\log(1/\delta)$.

3.3.4 Extensions

We present here a few extensions of our basic setting in Section 3.3.3. For simplicity, we only consider the margin-violation version of our definitions, but all the results here can be easily extended to the hinge loss case as well.

Combining Multiple Similarity Functions

Suppose that rather than having a single similarity function, we were instead given n functions $K_1, \ldots, K_n$, and our hope is that some convex combination of them will satisfy Definition 3.3.5. Is this sufficient to be able to learn well? (Note that a convex combination of similarity functions is guaranteed to have range $[-1, 1]$ and so be a legal similarity function.) The following generalization of Theorem 3.3.3 shows that this is indeed the case, though the margin parameter drops by a factor of $\sqrt{n}$. This result can be viewed as analogous to the idea of learning a kernel matrix studied by [157] except that rather than explicitly learning the best convex combination, we are simply folding the learning process into the second stage of the algorithm.

Theorem 3.3.5 *Suppose $K_1, \ldots, K_n$ are similarity functions such that some (unknown) convex combination of them is (ϵ, γ)-good. If one draws a set $S = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_d\}$ from P containing $d = 8\log(1/\delta)/\gamma^2$ examples, then with probability at least $1 - \delta$, the mapping $\phi^S : X \to R^{nd}$ defined as $\phi^S(x) = \frac{\rho^S(x)}{\sqrt{nd}}$,*

$$\rho^S(x) = (K_1(x, \tilde{x}_1), \ldots, K_1(x, \tilde{x}_d), ..., K_n(x, \tilde{x}_1), \ldots, K_n(x, y_d))$$

has the property that the induced distribution $\phi^S(P)$ in R^{nd} has a separator of error at most $\epsilon + \delta$ at margin at least $\gamma/(2\sqrt{n})$.

Proof: Let $K = \alpha_1 K_1 + \ldots + \alpha_n K_n$ be an (ϵ, γ)-good convex-combination of the K_i. By Theorem 3.3.3, had we instead performed the mapping: $\hat{\phi}^S : X \to R^d$ defined as $\hat{\phi}^S(x) = \frac{\hat{\rho}^S(x)}{\sqrt{d}}$,

$$\hat{\rho}^S(x) = (K(x, \tilde{x}_1), \ldots, K(x, \tilde{x}_d))$$

then with probability $1 - \delta$, the induced distribution $\hat{\phi}^S(P)$ in R^d would have a separator of error at most $\epsilon + \delta$ at margin at least $\gamma/2$. Let $\hat{\beta}$ be the vector corresponding to such a separator in that space. Now, let us convert $\hat{\beta}$ into a vector in R^{nd} by replacing each coordinate $\hat{\beta}_j$ with the n values $(\alpha_1\hat{\beta}_j, \ldots, \alpha_n\hat{\beta}_j)$. Call the resulting vector $\tilde{\beta}$. Notice that by design, for any x we have $\left\langle \tilde{\beta}, \phi^S(x) \right\rangle = \frac{1}{\sqrt{n}} \left\langle \hat{\beta}, \hat{\phi}^S(x) \right\rangle$. Furthermore, $||\tilde{\beta}|| \leq ||\hat{\beta}|| \leq 1$ (the worst case is when exactly one of the α_i is equal to 1 and the rest are 0). Thus, the vector $\tilde{\beta}$ under distribution $\phi^S(P)$ has the similar properties as the vector $\hat{\beta}$ under $\hat{\phi}^S(P)$; so, using the proof of Theorem 3.3.3 we obtain that that the induced distribution $\phi^S(P)$ in R^{nd} has a separator of error at most $\epsilon + \delta$ at margin at least $\gamma/(2\sqrt{n})$. ■

Note that the above argument actually shows something a bit stronger than Theorem 3.3.5. In particular, if we define $\alpha = (\alpha_1, \ldots, \alpha_n)$ to be the mixture vector for the optimal K, then we can replace the margin bound $\gamma/(2\sqrt{n})$ with $\gamma/(2||\alpha||\sqrt{n})$. For example, if α is the uniform mixture, then we just get the bound in Theorem 3.3.3 of $\gamma/2$.

Also note that if we are in fact using an L_1-based learning algorithm then we could do much better — for details on such an approach see Section 3.4.6.

Multi-class Classification

We can naturally extend all our results to multi-class classification. Assume for concreteness that there are r possible labels, and denote the space of possible labels by $Y = \{1, \cdots, r\}$; thus, by a *multi-class learning problem* we mean a distribution P over labeled examples $(x, y(x))$, where $x \in X$ and $y(x) \in Y$.

For this multi-class setting, Definition 3.3.4 seems most natural to extend. Specifically:

Definition 3.3.7 (main, multi-class) *A similarity function K is an (ϵ, γ)-**good similarity function** for a multi-class learning problem P if there exists a bounded weighting function w over X ($w(x') \in [0, 1]$ for all $x' \in X$) such that at least a $1 - \epsilon$ probability mass of examples x satisfy:*

$$\mathbf{E}_{x' \sim P}[w(x')K(x, x')|y(x) = y(x')] \quad \geq \quad \mathbf{E}_{x' \sim P}[w(x')K(x, x')|y(x) = i] + \gamma \text{ for all } i \in Y, i \neq y(x)$$

We can then extend the argument in Theorem 3.3.3 and learn using standard adaptations of linear-separator algorithms to the multiclass case (e.g., see [110]).

3.3.5 Relationship Between Good Kernels and Good Similarity Measures

As discussed earlier, the similarity-based theory of learning is more general than the traditional kernel-based theory, since a good similarity function need not be a valid kernel. However, for a similarity function K that is a valid kernel, it is interesting to understand the relationship between the learning results guaranteed by the two theories. Similar learning guarantees and sample complexity bounds can be obtained if K is either an (ϵ, γ)-good similarity function, or a valid kernel and (ϵ, γ)-kernel-good. In fact, as we saw in Section 3.3.3, the similarity-based guarantees are obtained by transforming (using a sample) the problem of learning with an (ϵ, γ)-good similarity function to learning with a kernel with essentially the same goodness parameters. This is made more explicit in Corollary 3.3.11.

In this section we study the relationship between a kernel function being good in the similarity sense of Definitions 3.3.5 and 3.3.6 and good in the kernel sense. We show that a valid kernel function that is good for one notion, is in fact good also for the other notion. The qualitative notions of being "good" are therefore equivalent for valid kernels, and so in this sense the more general similarity-based notion subsumes the familiar kernel-based notion.

However, as we will see, the similarity-based margin of a valid kernel might be lower than the kernel-based margin, yielding a possible increase in the sample complexity guarantees if a kernel is used as a similarity measure. We also show that for a valid kernel, the kernel-based margin is never smaller than the similarity-based margin. We provide a tight bound on this possible deterioration of the margin when switching to the similarity-based notion given by definitions 3.3.5 and 3.3.6. (Note also that in the following section 3.4 we provide an even better notion of a good similarity function that provides a better kernels to similarity translations.)

Specifically, we show that if a valid kernel function is good in the similarity sense, it is also good in the standard kernel sense, both for the margin violation error rate and for the hinge loss:

Theorem 3.3.6 (A kernel good as a similarity function is also good as a kernel) *If K is a valid kernel function, and is (ϵ, γ)-good similarity for some learning problem, then it is also (ϵ, γ)-kernel-good for the learning problem. If K is (ϵ, γ)-good similarity in hinge loss, then it is also (ϵ, γ)-kernel-good in hinge loss.*

We also show the converse—If a kernel function is good in the kernel sense, it is also good in the similarity sense, though with some degradation of the margin:

Theorem 3.3.7 (A good kernel is also a good similarity function—Margin violations) *If K is (ϵ_0, γ)-kernel-good for some learning problem (with deterministic labels), then it is also $(\epsilon_0 + \epsilon_1, \frac{1}{2}(1-\epsilon_0)\epsilon_1\gamma^2)$-good similarity for the learning problem, for any $\epsilon_1 > 0$.*

Note that in any useful situation $\epsilon_0 < \frac{1}{2}$, and so the guaranteed margin is at least $\frac{1}{4}\epsilon_1\gamma^2$. A similar guarantee holds also for the hinge loss:

Theorem 3.3.8 (A good kernel is also a good similarity function—Hinge loss) *If K is (ϵ_0, γ)-kernel-good in hinge loss for learning problem (with deterministic labels), then it is also $(\epsilon_0 + \epsilon_1, 2\epsilon_1\gamma^2)$-good similarity in hinge loss for the learning problem, for any $\epsilon_1 > 0$.*

These results establish that treating a kernel as a similarity function would still enable learning, although with a somewhat increased sample complexity. As we show, the deterioration of the margin in the above results, which yields an increase in the sample complexity guarantees, is unavoidable:

Theorem 3.3.9 (Tightness, Margin Violations) *For any $0 < \gamma < \sqrt{\frac{1}{2}}$ and any $0 < \epsilon_1 < \frac{1}{2}$, there exists a learning problem and a kernel function K, which is $(0, \gamma)$-kernel-good for the learning problem, but which is only $(\epsilon_1, 4\epsilon_1\gamma^2)$-good similarity. That is, it is not (ϵ_1, γ')-good similarity for any $\gamma' > 4\epsilon_1\gamma^2$.*

Theorem 3.3.10 (Tightness, Hinge Loss) *For any $0 < \gamma < \sqrt{\frac{1}{2}}$ and any $0 < \epsilon_1 < \frac{1}{2}$, there exists a learning problem and a kernel function K, which is $(0, \gamma)$-kernel-good in hinge loss for the learning problem, but which is only $(\epsilon_1, 32\epsilon_1\gamma^2)$-good similarity in hinge loss.*

To prove Theorem 3.3.6 we will show, for any weight function, an explicit low-norm linear predictor β (in the implied Hilbert space), with equivalent behavior. To prove Theorems 3.3.7 and 3.3.8, we will consider a kernel function that is (ϵ_0, γ)-kernel-good and show that it is also good as a similarity function. We will first treat goodness in hinge-loss and prove Theorem 3.3.8, which can be viewed as a more general result. This will be done using the representation of the optimal SVM solution in terms of the dual optimal solution. We then prove Theorem 3.3.7 in terms of the margin violation error rate, by using the hinge-loss as a bound on the error rate. To prove Theorems 3.3.9 and 3.3.10, we present an explicit learning problem and kernel.

Transforming a Good Similarity Function to a Good Kernel

Before proving the above Theorems, we briefly return to the mapping of Theorem 3.3.3 and explicitly present it as a mapping between a good similarity function and a good kernel:

Corollary 3.3.11 (A good similarity function can be transformed to a good kernel) *If K is an (ϵ, γ)-good similarity function for some learning problem P, then for any $0 < \delta < 1$, given a sample of S size $(8/\gamma^2)\log(1/\delta)$ drawn from P, we can construct, with probability at least $1-\delta$ over the draw of S, a valid kernel $\tilde{K}^S$ that is $(\epsilon + \delta, \gamma/2)$-kernel good for P.*

If K is a (ϵ, γ)-good similarity function in hinge-loss for some learning problem P, then for any $\epsilon_1 > 0$ and $0 < \delta < \gamma\epsilon_1/4$, given a sample of S size $16\log(1/\delta)/(\epsilon_1\gamma)^2$ drawn from P, we can construct, with probability at least $1-\delta$ over the draw of S, a valid kernel $\tilde{K}^S$ that is $(\epsilon + \epsilon_1, \gamma)$-kernel good for P.

Proof: Let $\tilde{K}^S(x, x') = \left\langle \phi^S(x), \phi^S(x') \right\rangle$ where ϕ^S is the transformation of Theorems 3.3.3 and 3.3.4. ■

From this statement, it is clear that kernel-based learning guarantees apply also to learning with a good similarity function, essentially with the same parameters.

It is important to understand that the result of Corollary 3.3.11 is of a very different nature than the results of Theorems 3.3.6– 3.3.10. The claim here is not that a good similarity function *is* a good kernel — it can't be if it is not positive semi-definite. But, given a good similarity function we can create a good kernel. This transformation is *distribution-dependent*, and can be calculated using a sample S.

Proof of Theorem 3.3.6

Consider a similarity function K that is a valid kernel, i.e. $K(x,x') = \langle \phi(x), \phi(x') \rangle$ for some mapping ϕ of x to a Hilbert space $\mathcal{H}$. For any input distribution and any valid weighting $w(x)$ of the inputs (i.e. $0 \leq w(x) \leq 1$), we will construct a linear predictor $\beta_w \in \mathcal{H}$, with $||\beta_w|| \leq 1$, such that similarity-based predictions using w are the same as the linear predictions made with β_w

Define the following linear predictor $\beta_w \in \mathcal{H}$:

$$\beta_w = \mathbf{E}_{x'}\big[y(x')w(x')\phi(x')\big].$$

The predictor β_w has norm at most:

$$\begin{aligned} ||\beta_w|| &= ||\mathbf{E}_{x'}\big[y(x')w(x')\phi(x')\big]|| \leq \max_{x'} ||y(x')w(x')\phi(x')|| \\ &\leq \max ||\phi(x')|| = \max \sqrt{K(x',x')} \leq 1 \end{aligned}$$

where the second inequality follows from $|w(x')|, |y(x')| \leq 1$.

The predictions made by β_w are:

$$\begin{aligned} \langle \beta_w, \phi(x) \rangle &= \big\langle \mathbf{E}_{x'}\big[y(x')w(x')\phi(x')\big], \phi(x) \big\rangle \\ &= \mathbf{E}_{x'}\big[y(x')w(x')\big\langle \phi(x'), \phi(x) \big\rangle\big] = \mathbf{E}_{x'}\big[y(x')w(x')K(x,x')\big] \end{aligned}$$

That is, using β_w is the same as using similarity-based prediction with w. In particular, if the margin violation rate, as well as the hinge loss, with respect to any margin γ, is the same for predictions made using either w or β_w. This is enough to establish Theorem 3.3.6: If K is (ϵ, γ)-good (perhaps for to the hinge-loss), there exists some valid weighting w the yields margin violation error rate (resp. hinge loss) at most ϵ with respect to margin γ, and so β_w yields the same margin violation (resp. hinge loss) with respect to the same margin, establishing K is (ϵ, γ)-kernel-good (resp. for the hinge loss).

Proof of Theorem 3.3.8: Guarantee on the Hinge Loss

Recall that we are considering only learning problems where the label y is a deterministic function of x. For simplicity of presentation, we first consider finite discrete distributions, where:

$$\Pr(\, x_i, y_i \,) = p_i \tag{3.14}$$

for $i = 1 \ldots n$, with $\sum_{i=1}^{n} p_i = 1$ and $x_i \neq x_j$ for $i \neq j$.

Let K be any kernel function that is (ϵ_0, γ)-kernel good in hinge loss.Let ϕ be the implied feature mapping and denote $\phi_i = \phi(x_i)$. Consider the following weighted-SVM quadratic optimization problem with regularization parameter C:

$$\text{minimize } \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n} p_i[1 - y_i\langle \beta, \phi_i \rangle]_+ \tag{3.15}$$

The dual of this problem, with dual variables α_i, is:

$$\begin{aligned} &\text{maximize } \sum_i \alpha_i - \frac{1}{2}\sum_{ij} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ &\text{subject to } \quad 0 \le \alpha_i \le C p_i \end{aligned} \tag{3.16}$$

There is no duality gap, and furthermore the primal optimum β^* can be expressed in terms of the dual optimum α^*: $\beta^* = \sum_i \alpha_i^* y_i \phi_i$.

Since K is (ϵ_0, γ)-kernel-good in hinge-loss, there exists a predictor $||\beta_0|| = 1$ with average-hinge loss ϵ_0 relative to margin γ. The primal optimum β^* of (3.15), being the optimum solution, then satisfies:

$$\begin{aligned} \frac{1}{2}||\beta^*||^2 + C\sum_i p_i[1 - y_i\langle\beta^*, \phi_i\rangle]_+ &\le \frac{1}{2}||\frac{1}{\gamma}\beta_0||^2 + C\sum_i p_i[1 - y_i\left\langle\frac{1}{\gamma}\beta_0, \phi_i\right\rangle]_+ \\ &= \frac{1}{2\gamma^2} + C\mathbf{E}\left[[1 - y\left\langle\frac{1}{\gamma}\beta_0, \phi(x)\right\rangle]_+\right] = \frac{1}{2\gamma^2} + C\epsilon_0 \end{aligned} \tag{3.17}$$

Since both terms on the left hand side are non-negative, each of them is bounded by the right hand side, and in particular:

$$C\sum_i p_i[1 - y_i\langle\beta^*, \phi_i\rangle]_+ \le \frac{1}{2\gamma^2} + C\epsilon_0 \tag{3.18}$$

Dividing by C we get a bound on the average hinge-loss of the predictor β^*, relative to a margin of one:

$$\mathbf{E}[[1 - y\langle\beta^*, \phi(x)\rangle]_+] \le \frac{1}{2C\gamma^2} + \epsilon_0 \tag{3.19}$$

We now use the fact that β^* can be written as $\beta^* = \sum_i \alpha_i^* y_i \phi_i$ with $0 \le \alpha_i^* \le C p_i$. Using the weights

$$w_i = w(x_i) = \alpha_i^*/(Cp_i) \le 1 \tag{3.20}$$

we have for every x, y:

$$\begin{aligned} y\mathbf{E}_{x',y'}\left[w(x')y'K(x,x')\right] &= y\sum_i p_i w(x_i) y_i K(x, x_i) \\ &= y\sum_i p_i \alpha_i^* y_i K(x, x_i)/(Cp_i) \\ &= y\sum_i \alpha_i^* y_i \langle\phi_i, \phi(x)\rangle/C = y\langle\beta^*, \phi(x)\rangle/C \end{aligned} \tag{3.21}$$

Multiplying by C and using (3.19):

$$\mathbf{E}_{x,y}\left[\,[\,1 - Cy\mathbf{E}_{x',y'}\left[w(x')y'K(x,x')\right]\,]_+\,\right] = \mathbf{E}_{x,y}[\,[\,1 - y\langle\beta^*, \phi(x)\rangle\,]_+\,] \le \frac{1}{2C\gamma^2} + \epsilon_0 \tag{3.22}$$

This holds for any C, and describes the average hinge-loss relative to margin $1/C$. To get an average hinge-loss of $\epsilon_0 + \epsilon_1$, we set $C = 1/(2\epsilon_1\gamma^2)$ and get:

$$\mathbf{E}_{x,y}\left[\,[\,1 - y\mathbf{E}_{x',y'}\left[w(x')y'K(x,x')\right]/(2\epsilon_1\gamma^2)\,]_+\,\right] \le \epsilon_0 + \epsilon_1 \tag{3.23}$$

This establishes that K is $(\epsilon_0 + \epsilon_1, 2\epsilon_1\gamma^2)$-good similarity in hinge-loss.

Non-discrete distributions

The same arguments apply also in the general (not necessarily discrete) case, except that this time, instead of a fairly standard (weighted) SVM problem, we must deal with a variational optimization problem, where the optimization variable is a random variable (a function from the sample space to the reals). We will present the dualization in detail.

We consider the primal objective

$$\text{minimize } \frac{1}{2}||\beta||^2 + C\mathbf{E}_{y,\phi}[[1 - y\langle\beta,\phi\rangle]_+] \tag{3.24}$$

where the expectation is w.r.t. the distribution P, with $\phi = \phi(x)$ here and throughout the rest of this section. We will rewrite this objective using explicit slack, in the form of a random variable ξ, which will be a variational optimization variable:

$$\begin{gathered}\text{minimize } \frac{1}{2}||\beta||^2 + C\mathbf{E}[\xi] \\ \text{subject to } \Pr(\,1 - y\langle\beta,\phi\rangle - \xi \leq 0\,) = 1 \\ \Pr(\,\xi \geq 0\,) = 1\end{gathered} \tag{3.25}$$

In the rest of this section all our constraints will implicitly be required to hold with probability one. We will now introduce the dual variational optimization variable α, also a random variable over the same sample space, and write the problem as a saddle problem:

$$\begin{gathered}\min_{\beta,\xi} \max_{\alpha} \frac{1}{2}||\beta||^2 + C\mathbf{E}[\xi] + \mathbf{E}[\alpha(1 - y\langle\beta,\phi\rangle - \xi)] \\ \text{subject to } \quad \xi \geq 0 \quad \alpha \geq 0\end{gathered} \tag{3.26}$$

Note that this choice of Lagrangian is a bit different than the more standard Lagrangian leading to (3.16). Convexity and the existence of a feasible point in the dual interior allows us to change the order of maximization and minimization without changing the value of the problem, even in the infinite case [134]. Rearranging terms we obtaining the equivalent problem:

$$\begin{gathered}\max_{\alpha} \min_{\beta,\xi} \frac{1}{2}||\beta||^2 - \langle\mathbf{E}[\alpha y\phi],\beta\rangle + \mathbf{E}[\xi(C - \alpha)] + \mathbf{E}[\alpha] \\ \text{subject to } \quad \xi \geq 0, \quad \alpha \geq 0\end{gathered} \tag{3.27}$$

Similarly to the finite case, we see that the minimum of the minimization problem is obtained when $\beta = \mathbf{E}[\alpha y\phi]$ and that it is finite when $\alpha \leq C$ almost surely, yielding the dual:

$$\begin{gathered}\text{maximize } \mathbf{E}[\alpha] - \frac{1}{2}\mathbf{E}\big[\alpha y\alpha' y K(x,x')\big] \\ \text{subject to } \quad 0 \leq \alpha \leq C\end{gathered} \tag{3.28}$$

where (x, y, α) and (x', y', α') are two independent draws from the same distribution. The primal optimum can be expressed as $\beta^* = \mathbf{E}[\alpha^* y\phi]$, where α^* is the dual optimum. We can now apply the same arguments as in (3.17),(3.18) to get (3.19). Using the weight mapping

$$w(x) = \mathbf{E}[\alpha^*|x] \,/\, C \leq 1 \tag{3.29}$$

we have for every x, y:

$$y\mathbf{E}_{x',y'}\big[w(x')y'K(x,x')\big] = y\big\langle\mathbf{E}_{x',y',\alpha'}\big[\alpha' y' x'\big], x\big\rangle/C = y\langle\beta^*,\phi(x)\rangle/C. \tag{3.30}$$

From here we can already get (3.22) and setting $C = 1/(2\epsilon_1\gamma^2)$ we get (3.23), which establishes Theorem 3.3.8 for any learning problem (with deterministic labels).

Proof of Theorem 3.3.7: Guarantee on Margin Violations

We will now turn to guarantees on similarity-goodness with respect to the margin violation error-rate. We base these on the results for goodness in hinge loss, using the hinge loss as a bound on the margin violation error-rate. In particular, a violation of margin $\gamma/2$ implies a hinge-loss at margin γ of at least $\frac{1}{2}$. Therefore, twice the average hinge-loss at margin γ is an upper bound on the margin violation error rate at margin $\gamma/2$.

The kernel-separable case, i.e. $\epsilon_0 = 0$, is simpler, and we consider it first. Having no margin violations implies zero hinge loss. And so if a kernel K is $(0, \gamma)$-kernel-good, it is also $(0, \gamma)$-kernel-good in hinge loss, and by Theorem 3.3.8 it is $(\epsilon_1/2, 2(\epsilon_1/2)\gamma^2)$-good similarity in hinge loss. Now, for any $\epsilon_1 > 0$, by bounding the margin $\frac{1}{2}\epsilon_1\gamma^2$ error-rate by the $\epsilon_1\gamma^2$ average hinge loss, K is $(\epsilon_1, \frac{1}{2}\epsilon_1\gamma^2)$-good similarity, establishing Theorem 3.3.7 for the case $\epsilon_0 = 0$.

We now return to the non-separable case, and consider a kernel K that is (ϵ_0, γ)-kernel-good, with some non-zero error-rate ϵ_0. Since we cannot bound the hinge loss in terms of the margin-violations, we will instead consider a modified distribution where the margin-violations are removed.

Let β^* be the linear classifier achieving ϵ_0 margin violation error-rate with respect to margin γ, i.e. such that $\Pr(\, y\langle\beta^*, x\rangle \geq \gamma\,) > 1-\epsilon_0$. We will consider a distribution which is conditioned on $y\langle\beta^*, x\rangle \geq \gamma$. We denote this event as $\textsc{ok}(x)$ (recall that y is a deterministic function of x). The kernel K is obviously $(0, \gamma)$-kernel-good, and so by the arguments above also $(\epsilon_1, \frac{1}{2}\epsilon_1\gamma^2)$-good similarity, on the conditional distribution. Let w be the weight mapping achieving

$$\Pr_{x,y}\Big(\, y\mathbf{E}_{x',y'}\big[w(x')y'K(x,x')|\textsc{ok}(x')\big] < \gamma_1|\textsc{ok}(x)\,\Big) \leq \epsilon_1, \tag{3.31}$$

where $\gamma_1 = \frac{1}{2}\epsilon_1\gamma^2$, and set $w(x) = 0$ when $\textsc{ok}(x)$ does not hold. We have:

$$\begin{aligned}
&\Pr_{x,y}\Big(\, y\mathbf{E}_{x',y'}\big[w(x')y'K(x,x')\big] < (1-\epsilon_0)\gamma_1\,\Big) \\
&\quad\leq \Pr(\text{ not }\textsc{ok}(x)\,) + \Pr(\,\textsc{ok}(x)\,)\Pr_{x,y}\Big(\; y\mathbf{E}_{x',y'}\big[w(x')y'K(x,x')\big] < (1-\epsilon_0)\gamma_1 \mid \textsc{ok}(x)\;\Big) \\
&\quad= \epsilon_0 + (1-\epsilon_0)\Pr_{x,y}\Big(\, y(1-\epsilon_0)\mathbf{E}_{x',y'}\big[w(x')y'K(x,x')|\textsc{ok}(x)\big] < (1-\epsilon_0)\gamma_1|\textsc{ok}(x)\,\Big) \\
&\quad= \epsilon_0 + (1-\epsilon_0)\Pr_{x,y}\Big(\, y\mathbf{E}_{x',y'}\big[w(x')y'K(x,x')|\textsc{ok}(x)\big] < \gamma_1|\textsc{ok}(x)\,\Big) \\
&\quad\leq \epsilon_0 + (1-\epsilon_0)\epsilon_1 \leq \epsilon_0 + \epsilon_1
\end{aligned} \tag{3.32}$$

establishing that K is $(\epsilon_0 + \epsilon_1, \gamma_1)$-good similarity for the original (unconditioned) distribution, thus yielding Theorem 3.3.7.

Tightness

We now turn to proving of Theorems 3.3.9 and 3.3.10. This is done by presenting a specific distribution P and kernel in which the guarantees hold tightly.

Consider the standard Euclidean inner-product and a distribution on four labeled points in R^3, given by:

$$\begin{aligned}
x_1 &= (\gamma, \gamma, \sqrt{1-2\gamma^2}), & y_1 &= 1, & p_1 &= \frac{1}{2} - \epsilon \\
x_2 &= (\gamma, -\gamma, \sqrt{1-2\gamma^2}), & y_2 &= 1, & p_2 &= \epsilon \\
x_3 &= (-\gamma, \gamma, \sqrt{1-2\gamma^2}), & y_3 &= -1, & p_3 &= \epsilon \\
x_4 &= (-\gamma, -\gamma, \sqrt{1-2\gamma^2}), & y_4 &= -1, & p_4 &= \frac{1}{2} - \epsilon
\end{aligned}$$

for some (small) $0 < \gamma < \sqrt{\frac{1}{2}}$ and (small) probability $0 < \epsilon < \frac{1}{2}$. The four points are all on the unit sphere (i.e. $||x_i|| = 1$ and so $K(x_i, x_j) = \langle x_i, x_j \rangle \leq 1$), and are clearly separated by $\beta = (1, 0, 0)$ with a margin of γ. The standard inner-product kernel is therefore $(0, \gamma)$-kernel-good on this distribution.

Proof of Theorem 3.3.9: Tightness for Margin-Violations

We will show that when this kernel (the standard inner product kernel in R^3) is used as a similarity function, the best margin that can be obtained on all four points, i.e. on at least $1 - \epsilon$ probability mass of examples, is $8\epsilon\gamma^2$.

Consider the classification margin on point x_2 with weights w (denote $w_i = w(x_i)$):

$$\begin{aligned}
&\mathbf{E}[w(x)yK(x_2, x)] \\
&= (\frac{1}{2} - \epsilon)w_1(\gamma^2 - \gamma^2 + (1 - 2\gamma^2)) + \epsilon w_2(2\gamma^2 + (1 - 2\gamma^2)) \\
&\quad -\epsilon w_3(-2\gamma^2 + (1 - 2\gamma^2)) - (\frac{1}{2} - \epsilon)w_4(-\gamma^2 + \gamma^2 + (1 - 2\gamma^2)) \\
&= \left((\frac{1}{2} - \epsilon)(w_1 - w_4) + \epsilon(w_2 - w_3)\right)(1 - 2\gamma^2) \; + \; 2\epsilon(w_2 + w_3)\gamma^2 \qquad (3.33)
\end{aligned}$$

If the first term is positive, we can consider the symmetric calculation

$$-\mathbf{E}[w(x)yK(x_3, x)] = -\left((\frac{1}{2} - \epsilon)(w_1 - w_4) + \epsilon(w_2 - w_3)\right)(1 - 2\gamma^2) \; + \; 2\epsilon(w_2 + w_3)\gamma^2$$

in which the first term is negated. One of the above margins must therefore be at most

$$2\epsilon(w_2 + w_3)\gamma^2 \leq 4\epsilon\gamma^2 \qquad (3.34)$$

This establishes Theorem 3.3.9.

Proof of Theorem 3.3.10: Tightness for the Hinge Loss

In the above example, suppose we would like to get an average hinge-loss relative to margin γ_1 of at most ϵ_1:

$$\mathbf{E}_{x,y}\left[\,[\,1 - y\mathbf{E}_{x',y'}\left[w(x')y'K(x, x')\right]/\gamma_1\,]_+\,\right] \leq \epsilon_1 \qquad (3.35)$$

Following the arguments above, equation (3.34) can be used to bound the hinge-loss on at least one of the points x_2 or x_3, which, multiplied by the probability ϵ of the point, is a bound on the average hinge loss:

$$\mathbf{E}_{x,y}\left[\,[\,1 - y\mathbf{E}_{x',y'}\left[w(x')y'K(x, x')\right]/\gamma_1\,]_+\,\right] \geq \epsilon(1 - 4\epsilon\gamma^2/\gamma_1) \qquad (3.36)$$

and so to get an an average hinge-loss of at most ϵ_1 we must have:

$$\gamma_1 \leq \frac{4\epsilon\gamma^2}{1 - \epsilon_1/\epsilon} \qquad (3.37)$$

For any target hinge-loss ϵ_1, consider a distribution with $\epsilon = 2\epsilon_1$, in which case we get that the maximum margin attaining average hinge-loss ϵ_1 is $\gamma_1 = 16\epsilon_1\gamma^2$, even though we can get a hinge loss of zero at margin γ using a kernel. This establishes Theorem 3.3.10.

Note: One might object that the example used in Theorems 3.3.9 and 3.3.10 is a bit artificial, since K has margin $O(\gamma^2)$ in the similarity sense just because $1 - 4\gamma^2 \leq K(x_i, x_j) \leq 1$. Normalizing K to $[-1, 1]$ we would obtain a similarity function that has margin $O(1)$. However, this "problem" can be simply fixed by adding the symmetric points on the lower semi-sphere:

$$\begin{aligned}
x_5 &= (\gamma, \gamma, -\sqrt{1-2\gamma^2}), & y_5 &= 1, & p_5 &= \frac{1}{4} - \epsilon \\
x_6 &= (\gamma, -\gamma, -\sqrt{1-2\gamma^2}), & y_6 &= 1, & p_6 &= \epsilon \\
x_7 &= (-\gamma, \gamma, -\sqrt{1-2\gamma^2}), & y_7 &= -1, & p_7 &= \epsilon \\
x_8 &= (-\gamma, -\gamma, -\sqrt{1-2\gamma^2}), & y_8 &= -1, & p_8 &= \frac{1}{4} - \epsilon
\end{aligned}$$

and by changing $p_1 = \frac{1}{4} - \epsilon$ and $p_4 = \frac{1}{4} - \epsilon$. The classification margins on x_2 and x_3 are now (compare with (3.33)):

$$\begin{aligned}
\mathbf{E}[w(x)yK(x_2, x)] &= \left((\frac{1}{4} - \epsilon)(w_1 - w_4 - w_5 + w_8) + \epsilon(w_2 - w_3 - w_6 + w_7)\right)(1 - 2\gamma^2) \\
&\quad + 2\epsilon(w_2 + w_3 + w_6 + w_7)\gamma^2 \\
-\mathbf{E}[w(x)yK(x_3, x)] &= -\left((\frac{1}{4} - \epsilon)(w_1 - w_4 - w_5 + w_8) + \epsilon(w_2 - w_3 - w_6 + w_7)\right)(1 - 2\gamma^2) \\
&\quad + 2\epsilon(w_2 + w_3 + w_6 + w_7)\gamma^2
\end{aligned}$$

One of the above classification margins must therefore be at most $2\epsilon(w_2 + w_3 + w_6 + w_7)\gamma^2 \leq 8\epsilon\gamma^2$. And so, even though the similarity is "normalized", and is $(0, \gamma)$-kernel-good, it is only $(\epsilon, 8\epsilon\gamma^2)$-good as a similarity function. Proceeding as in the proof of Theorem 3.3.10 establishes the modified example is also only $(\epsilon, 64\epsilon\gamma^2)$-good in hinge loss.

Probabilistic Labels

So far, we have considered only learning problems where the label y is a deterministic function of x. Here, we discuss the necessary modifications to extend our theory also to noisy learning problems, where the same point x might be associated with both positive and negative labels with positive probabilities.

Although the learning guarantees are valid also for noisy learning problems, a kernel that is kernel-good for a noisy learning problem might not be good as a similarity function for this learning problem. To amend this, the definition of a good similarity function must be corrected, allowing the weights to depend not only on the point x but also on the label y:

Definition 3.3.8 (Main, Margin Violations, Corrected for Noisy Problems) *A similarity function K is an (ϵ, γ)-**good similarity function** for a learning problem P if there* exists *a bounded weighting function w over $X \times \{-1, +1\}$ ($w(x', y') \in [0, 1]$ for all $x' \in X, y' \in \{-1, +1\}$) such that at least a $1 - \epsilon$ probability mass of examples x, y satisfy:*

$$\mathbf{E}_{x',y' \sim P}[yy'w(x', y')K(x, x')] \geq \gamma. \tag{3.38}$$

It is easy to verify that Theorem 3.3.3 can be extended also to this corrected definition. The same mapping ϕ^S can be used, with $\beta_i = \tilde{y}_i w(\tilde{x}_i, \tilde{y}_i)$, where $\tilde{y}_i$ is the training label of example i. Definition 3.3.6 and Theorem 3.3.4 can be extended in a similar way.

With these modified definitions, Theorems 3.3.7 and 3.3.8 extend also to noisy learning problems. In the proof of Theorem 3.3.8, two of the points x_i, x_j might be identical, but have different labels $y_i =$

$1, y_j = -1$ associated with them. This might lead to two different weights w_i, w_j for the same point. But since w is now allowed to depend also on the label, this does not pose a problem. In the non-discrete case, this corresponds to defining the weight as:

$$w(x, y) = \mathbf{E}[\alpha^* | x, y] \,/\, C. \tag{3.39}$$

3.4 Learning with More General Similarity Functions: A Better Definition

We develop here a new notion of a good similarity function that broadens the Balcan - Blum'06 notion [24] presented in Section 3.3 while still guaranteeing learnability. As with the Balcan - Blum'06 notion, this new definition talks in terms of natural similarity-based properties and does not require positive semi-definiteness or reference to implicit spaces. However, this new notion improves on the previous Balcan - Blum'06 definition in two important respects.

First, this new notion provides a better kernel-to-similarity translation. Any large-margin kernel function is a good similarity function under the new definition, and while we still incur some loss in the parameters, this loss is much smaller than under the prior definition, especially in terms of the final labeled sample-complexity bounds. In particular, when using a valid kernel function as a similarity function, a substantial portion of the previous sample-complexity bound can be transferred over to merely a need for *un*labeled examples.

Second, we show that the new definition allows for good similarity functions to exist for concept classes for which there is *no* good kernel. In particular, for any concept class C and sufficiently unconcentrated distribution D, we show there exists a similarity function under our definition with parameters yielding a labeled sample complexity bound of $O(\frac{1}{\epsilon} \log |C|)$ to achieve error ϵ, matching the ideal sample complexity for a generic hypothesis class. In fact, we also extend this result to classes of finite VC-dimension rather than finite cardinality. In contrast, we show there exist classes C such that under the uniform distribution over the instance space, there is no kernel with margin $8/\sqrt{|C|}$ for all $f \in C$ even if one allows 0.5 average hinge-loss. Thus, the margin-based guarantee on sample complexity for learning such classes with kernels is $\Omega(|C|)$. This extends work of [50] and [109] who give hardness results with comparable margin bounds, but at much lower error rates. [209] provide lower bounds for kernels with similar error rates, but their results hold only for regression (not hinge loss). Note that given access to unlabeled data, any similarity function under the Balcan - Blum'06 definition [24] can be converted to a kernel function with approximately the same parameters. Thus, our lower bound for kernel functions applies to that definition as well. These results establish a gap in the representational power of similarity functions under our new definition relative to the representational power of either kernels or similarity functions under the old definition.

Both this new definition and the Balcan - Blum'06 definition are based on the idea of a similarity function being good for a learning problem if there exists a non-negligible subset R of "representative points" such that most examples x are on average more similar to the representative points of their own label than to the representative points of the other label. (Formally, the "representativeness" of an example may be given by a weight between 0 and 1 and viewed as probabilistic or fractional.) However, the previous Balcan - Blum'06 definition combined the two quantities of interest—the probability mass of representative points and the gap in average similarity to representative points of each label—into a single margin parameter. The new notion keeps these quantities distinct, which turns out to make a substantial difference both in terms of broadness of applicability and in terms of the labeled sample complexity bounds that result.

Note that we distinguish between labeled and unlabeled sample complexities: while the total number of examples needed depends polynomially on the two quantities of interest, the number of labeled

examples will turn out to depend only logarithmically on the probability mass of the representative set and therefore may be much smaller under the new definition. This is especially beneficial in situations as described in Chapter 2 in which unlabeled data is plentiful but labeled data is scarce, or the distribution is known and so unlabeled data is free. We discuss in detail the relation to the model in Chapter 2 in Section 3.5.

Another way to view the distinction between the two notions of similarity is that we now require good predictions using a weight function with expectation bounded by 1, rather than supremum bounded by 1: compare the old Definition 3.3.5 and the variant of the new definition given as Definition 3.4.4. (We do in fact still have a bound on the supremum which is much larger, but this bound only affects the labeled sampled complexity logarithmically.) In Theorem 3.4.13 we make the connection between the two versions of the new definition explicit.

Conditioning on a subset of representative points, or equivalently bounding the expectation of the weight function, allows us to base our learnability results on L_1-regularized linear learning. The actual learning rule we get, given in Equation (3.49), is very similar, and even identical, to learning rules suggested by various authors and commonly used in practice as an alternative to Support Vector Machines [54, 127, 185, 192, 198]. Here we give a firm theoretical basis to this learning rule, with explicit learning guarantees, and relate it to simple and intuitive properties of the similarity function or kernel used (see the discussion at the end of Section 3.4.2).

3.4.1 New Notions of Good Similarity Functions

In this section we provide new notions of good similarity functions generalizing the main definitions in Section 3.3 (Definitions 3.3.5 and 3.3.6) that we prove have a number of important advantages. For simplicity in presentation, for most of this section we will consider only learning problems where the label y is a deterministic function of x. For such learning problems, we can use $y(x)$ to denote the label of point x.

In the Definitions 3.3.5 and 3.3.6 in section 3.3, a weight $w(x') \in [0,1]$ was used in defining the quantity of interest, namely $\mathbf{E}_{(x',y')\sim P}[y'w(x')K(x,x')]$. Here, it will instead be more convenient to think of $w(x)$ as the expected value of an indicator random variable $R(x) \in \{0,1\}$ where we will view the (probabilistic) set $\{x : R(x) = 1\}$ as a set of "representative points". Formally, for each $x \in X$, $R(x)$ is a discrete random variable over $\{0,1\}$ and we will then be sampling from the joint distribution of the form

$$\Pr(x,y,r) = \Pr(x,y)\Pr(R(x) = r) \tag{3.40}$$

in the discrete case or

$$p(x,y,r) = p(x,y)\Pr(R(x) = r) \tag{3.41}$$

in the continuous case, where p is a probability density function of P.

Our new definition is now as follows.

Definition 3.4.1 (Main, Margin Violations) *A similarity function K is an (ϵ,γ,τ)-**good similarity function** for a learning problem P if there* exists *an extended distribution $P(x,y,r)$ defined as in 3.40 or 3.41 such that the following conditions hold:*

1. *A $1-\epsilon$ probability mass of examples $(x,y) \sim P$ satisfy*

$$\mathbf{E}_{(x',y',r')\sim P}[yy'K(x,x') \mid r' = 1] \geq \gamma \tag{3.42}$$

2. $\Pr_{(x',y',r')}[r' = 1] \geq \tau$.

If the representative set R is $50/50$ positive and negative (i.e., $\Pr_{(x',y',r')}[y'=1|r'=1]=1/2$), we can interpret the condition as stating that most examples x are on average 2γ more similar to random representative examples x' of their own label than to random representative examples x' of the other label. The second condition is that at least a τ fraction of the points should be representative.

We also consider a hinge-loss version of the definition:

Definition 3.4.2 (Main, Hinge Loss) *A similarity function K is an (ϵ,γ,τ)-**good similarity function in hinge loss** for a learning problem P if there* exists *an extended distribution $P(x,y,r)$ defined as in 3.40 or 3.41 such that the following conditions hold:*

1. *We have*
$$\mathbf{E}_{(x,y)\sim P}\Big[\,[1-yg(x)/\gamma]_+\,\Big]\leq\epsilon, \tag{3.43}$$
where $g(x)=\mathbf{E}_{(x',y',r')}[y'K(x,x')\mid r'=1]$.
2. $\Pr_{(x',y',r')}[r'=1]\geq\tau$.

It is not hard to see that an (ϵ,γ)-good similarity function under Definitions 3.3.5 and 3.3.6 is also an (ϵ,γ,γ)-good similarity function under Definitions 3.4.1 and 3.4.2, respectively. In the reverse direction, an (ϵ,γ,τ)-good similarity function under Definitions 3.4.1 and 3.4.2 is an $(\epsilon,\gamma\tau)$-good similarity function under Definitions 3.3.5 and 3.3.6 (respectively). Specifically:

Theorem 3.4.1 *If K is an (ϵ,γ)-good similarity function under Definitions 3.3.5 and 3.3.6, then K is also an (ϵ,γ,γ)-good similarity function under Definitions 3.4.1 and 3.4.2, respectively.*

Proof: If we set $\Pr_{(x',y',r')}(r'=1\mid x')=w(x')$, we get that in order for any point x to fulfill equation (3.6), we must have

$$\Pr_{(x',y',r')}(r'=1)=\mathbf{E}_{x'}[w(x')]\geq\mathbf{E}_{(x',y')}[yy'w(x')K(x,x')]\geq\gamma.$$

Furthermore, for any x,y for which (3.6) is satisfied, we have

$$\begin{aligned}\mathbf{E}_{(x',y',r')}[yy'K(x,x')|r'=1] &= \mathbf{E}_{(x',y')}[yy'K(x,x')w(x')]/\Pr_{(x',y',r')}(r'=1)\\ &\geq \mathbf{E}_{(x',y')}[yy'K(x,x')w(x')]\geq\gamma.\end{aligned}$$

■

Theorem 3.4.2 *If K is an (ϵ,γ,τ)-good similarity function under Definitions 3.4.1 and 3.4.2, then K is an $(\epsilon,\gamma\tau)$-good similarity function under Definitions 3.3.5 and 3.3.6 (respectively).*

Proof: Setting $w(x')=\Pr_{(x',y',r')}(r'=1\mid x')$ we have for any x,y satisfying (3.42) that

$$\begin{aligned}\mathbf{E}_{(x',y')}[yy'K(x,x')w(x')] &= \mathbf{E}_{(x',y',r')}[yy'K(x,x')r'=1]\\ &= \mathbf{E}_{(x',y',r's)}[yy'K(x,x')|r'=1]\Pr_{(x',y',r')}(r'=1)\geq\gamma\tau.\end{aligned}$$

A similar calculation establishes the correspondence for the hinge loss. ■

As we will see, under both old and new definitions, the number of labeled samples required for learning grows as $1/\gamma^2$. The key distinction between them is that we introduce a new parameter, τ, that primarily affects the number of *unlabeled* examples required. This decoupling of the number of labeled and unlabeled examples enables us to handle a wider variety of situations with an improved labeled sample complexity. In particular, in translating from a kernel to a similarity function, we will find that much of the loss can now be placed into the τ parameter.

In the following we prove three types of results about this new notion of similarity. The first is that similarity functions satisfying these conditions are sufficient for learning (in polynomial time in the case of Definition 3.4.2), with a sample size of $O(\frac{1}{\gamma^2}\ln(\frac{1}{\gamma\tau}))$ labeled examples and $O(\frac{1}{\tau\gamma^2})$ unlabeled examples. This is particularly useful in settings where unlabeled data is plentiful and cheap—such settings are increasingly common in learning applications [82, 172]—or for distribution-specific learning where unlabeled data may be viewed as free.

The second main theorem we prove is that *any* class C, over a sufficiently unconcentrated distribution on examples, has a $(0, 1, 1/(2|C|))$-good similarity function (under either definition 3.4.1 or 3.4.2), whereas there exist classes C that have no $(0.5, 8/\sqrt{|C|})$-good kernel functions in hinge loss. This provides a clear separation between the similarity and kernel notions in terms of the parameters controlling labeled sample complexity. The final main theorem we prove is that any large-margin kernel function also satisfies our similarity definitions, with substantially less loss in the parameters controlling labeled sample complexity compared to the Balcan - Blum'06 definitions. For example, if K is a $(0, \gamma)$-good kernel, then it is an $(\epsilon', \epsilon'\gamma^2)$-good similarity function under Definitions 3.3.5 and 3.3.6, and this is tight [195], resulting in a sample complexity of $\tilde{\mathcal{O}}\big(1/(\gamma^4\epsilon^3)\big)$ to achieve error ϵ. However, we can show K is an $(\epsilon', \gamma^2, \epsilon')$-good similarity function under the new definition,[6] resulting in a sample complexity of only $\tilde{\mathcal{O}}\big(1/(\gamma^4\epsilon)\big)$.

3.4.2 Good Similarity Functions Allow Learning

The basic approach proposed for learning using a similarity function is similar to that in Section 3.3 and in [24]. First, a feature space is constructed, consisting of similarities to randomly chosen landmarks. Then, a linear predictor is sought in this feature space. However, for the previous Balcan - Blum'06 definitions (Definitions 3.3.5 and 3.3.6 in Section 3.3), we used guarantees for large L_2-margin in this feature space, whereas under the new definitions we will be using guarantees about large L_1-margin in the feature space.[7]

After recalling the notion of an L_1-margin and its associated learning guarantee, we first establish that, for an (ϵ, γ, τ)-good similarity function, the feature map constructed using $\tilde{\mathcal{O}}\big(1/(\tau\gamma^2)\big)$ landmarks indeed has (with high probability) a large L_1-margin separator. Using this result, we then obtain a learning guarantee by following the strategy outlined above.

In speaking of L_1-margin γ, we refer to separation with a margin γ by a unit-L_1-norm linear separator, in a unit-L_∞-bounded feature space. Formally, let $\phi : x \mapsto \phi(x)$, $\phi(x) \in R^d$, with $||\phi(x)||_\infty \leq 1$ be a mapping of the data to a d-dimensional feature space. We say that a linear predictor $\alpha \in R^d$, achieves error ϵ relative to L_1-margin γ if $\Pr_{(x,y(x))}(y(x)\langle\alpha, \phi(x)\rangle \geq \gamma) \geq 1 - \epsilon$ (this is the standard margin constraint) and $||\alpha||_1 = 1$.

Given a d-dimensional feature map under which there exists some (unknown) zero-error linear separator with L_1-margin γ, we can with high probability $1 - \delta$ efficiently learn a predictor with error at most ϵ_{acc} using $O\left(\frac{\log(d/\delta)}{\epsilon_{\text{acc}}\gamma^2}\right)$ examples. This can be done using the Winnow algorithm with a standard online-to-batch conversion [163]. If we can only guarantee the existence of a separator with error $\epsilon > 0$ relative to L_1-margin γ, then a predictor with error $\epsilon + \epsilon_{\text{acc}}$ can be theoretically learned (with high probability $1 - \delta$) from a sample of $\tilde{\mathcal{O}}\big((\log(d/\delta))/(\gamma^2\epsilon_{\text{acc}}^2)\big)$ examples by minimizing the number of L_1-margin γ violations on the sample [211].

We are now ready to state the main result enabling learning using good similarity functions:

[6] Formally, the translation produces an $(\epsilon', \gamma^2/c, \epsilon' c)$-good similarity function for some $c \leq 1$. However, smaller values of c only improve the bounds.

[7] Note that in fact even for the previous Balcan - Blum'06 definitions we could have used guarantees for large L_1-margin in this feature space; however for the new definitions we cannot necessarily use guarantees about large L_2-margin in the feature space.

Theorem 3.4.3 *Let K be an (ϵ, γ, τ)-good similarity function for a learning problem P. Let $S = \{x'_1, x'_2, \ldots, x'_d\}$ be a (potentially unlabeled) sample of*

$$d = \frac{2}{\tau}\left(\log(2/\delta) + 8\frac{\log(2/\delta)}{\gamma^2}\right)$$

landmarks drawn from P. Consider the mapping $\phi^S : X \to R^d$ defined as follows: ${\phi^S}_i(x) = K(x, x'_i)$, $i \in \{1, \ldots, d\}$. Then, with probability at least $1 - \delta$ over the random sample S, the induced distribution $\phi^S(P)$ in R^d has a separator of error at most $\epsilon + \delta$ relative to L_1 margin at least $\gamma/2$.

Proof: First, note that since $|K(x, x)| \leq 1$ for all x, we have $||\phi^S(x)||_\infty \leq 1$.

For each landmark x'_i, let r'_i be a draw from the distribution given by $R(x'_i)$. Consider the linear separator $\alpha \in R^d$, given by $\alpha_i = y(x'_i)r'_i/d_1$ where $d_1 = \sum_i r'_i$ is the number of landmarks with $R(\tilde{x}) = 1$. This normalization ensures $||\alpha||_1 = 1$.

We have, for any $x, y(x)$:

$$y(x)\langle \alpha, \phi^S(x)\rangle = \frac{\sum_{i=1}^d y(x)y(x'_i)r'_i K(x, x'_i)}{d_1} \tag{3.44}$$

This is an empirical average of d_1 terms

$$-1 \leq y(x)y(x')K(x, x') \leq 1$$

for which $R(x') = 1$. For any x we can apply Hoeffding's inequality, and obtain that with probability at least $1 - \delta^2/2$ over the choice of S, we have:

$$y(x)\langle \alpha, \phi^S(x)\rangle \geq \mathbf{E}_{x'}[K(x, x')y(x')y(x)|R(x')] - \sqrt{\frac{2\log(\frac{2}{\delta^2})}{d_1}} \tag{3.45}$$

Since the above holds for any x with probability at least $1 - \delta^2/2$ over S, it also holds with probability at least $1 - \delta^2/2$ over the choice of x and S. We can write this as:

$$\mathbf{E}_{S\sim P^d}\left[\Pr_{x\sim P}(\text{ violation })\right] \leq \delta^2/2 \tag{3.46}$$

where "violation" refers to violating (3.45). Applying Markov's inequality we get that with probability at least $1 - \delta/2$ over the choice of S, at most δ fraction of points violate (3.45). Recalling Definition 3.4.1, at most an additional ϵ fraction of the points violate (3.42). But for the remaining $1 - \epsilon - \delta$ fraction of the points, for which both (3.45) and (3.42) hold, we have:

$$y(x)\langle \alpha, \phi^S(x)\rangle \geq \gamma - \sqrt{\frac{2\log(\frac{1}{\delta^2})}{d_1}}. \tag{3.47}$$

To bound the second term we need an upper bound on d_1, the number of representative landmarks. The probability of each of the d landmarks being representative is at least τ and so the number of representative landmarks follows a Binomial distribution, ensuring $d_1 \geq 8\log(1/\delta)/\gamma^2$ with probability at least $1 - \delta/2$. When this happens, we have $\sqrt{\frac{2\log(\frac{1}{\delta^2})}{d_1}} \leq \gamma/2$. We get then, that with probability at least $1 - \delta$, for at least $1 - \epsilon - \delta$ of the points:

$$y(x)\langle \alpha, \phi^S(x)\rangle \geq \gamma/2. \tag{3.48}$$

■

For the realizable ($\epsilon = 0$) case, we obtain:

Corollary 3.4.4 *If K is a $(0, \gamma, \tau)$-good similarity function then with high probability we can efficiently find a predictor with error at most ϵ_{acc} from an unlabeled sample of size $d_u = \tilde{\mathcal{O}}\left(\frac{1}{\gamma^2\tau}\right)$ and from a labeled sample of size $d_l = \tilde{\mathcal{O}}\left(\frac{\log d_u}{\gamma^2\epsilon_{acc}}\right)$.*

Proof: We have proved in Theorem 3.4.3 that if K is$(0, \gamma, \tau)$-good similarity function, then with high probability there exists a low-error large-margin (at least $\frac{\gamma}{2}$) separator in the transformed space under mapping ϕ^S. Thus, all we need now to learn well is to draw a new fresh sample $\tilde{S}$, map it into the transformed space using ϕ^S, and then apply a good algorithm for learning linear separators in the new space that produces a hypothesis of error at most ϵ_{acc} with high probability. In particular, remember that the vector α has error at most δ at L_1 margin $\gamma/2$ over $\phi^S(P)$, where the mapping ϕ^S produces examples of L_∞ norm at most 1. In order to enjoy the better learning guarantees of the separable case, we will set δ_u small enough so that no bad points appear in the sample. Specifically, if we draw

$$d_u = d(\gamma, \delta_u, \tau) = \frac{2}{\tau}\left(\log(2/\delta_u) + 8\frac{\log(2/\delta_u)}{\gamma^2}\right)$$

unlabeled examples then with probability at least $1-\delta_u$ over the random sample S, the induced distribution $\phi^S(P)$ in R^{d_u} has a separator of error at most δ_u relative to L_1 margin at least $\gamma/2$. So, if we draw $O\left(\frac{1}{\gamma^2\epsilon_{\text{acc}}}\ln\left(d_u/\delta\right)\right)$ new labeled examples then with high probability $1-\delta_{final}$ these points are linearly separable at margin $\gamma/2$, where $\delta_{final} = c_1\delta_u\frac{1}{\epsilon_{\text{acc}}\gamma^2}\ln\left(d_u/\delta\right)$, where c_1 is a constant.

Setting $\delta_u = \epsilon_{\text{acc}}\gamma^2\tau\delta/(c_2\ln\left(1/(\epsilon_{\text{acc}}\gamma\tau\delta)\right))$ (where c_2 is a constant) we get that high probability $1-\delta/2$ these points are linearly separable at margin $\gamma/2$ in the new feature space. The Corollary now follows from the L_1-margin learning guarantee in the separable case, discussed earlier in the section. ■

For the most general $(\epsilon > 0)$ case, Theorem 3.4.3 implies that by following our two-stage approach, first using $d_u = \tilde{\mathcal{O}}\left(\frac{1}{\gamma^2\tau}\right)$ unlabeled examples as landmarks in order to construct $\phi^S(\cdot)$, and then using a fresh sample of size $d_l = \tilde{\mathcal{O}}\left(\frac{1}{\gamma^2\epsilon_{\text{acc}}^2}\ln d_u\right)$ to learn a low-error L_1-margin γ separator in $\phi^S(\cdot)$, we have:

Corollary 3.4.5 *If K is a (ϵ, γ, τ)-good similarity function then by minimizing L_1 margin violations we can find a predictor with error at most $\epsilon + \epsilon_{acc}$ from an unlabeled sample of size $d_u = \tilde{\mathcal{O}}\left(\frac{1}{\gamma^2\tau}\right)$ and from a labeled sample of size $d_l = \tilde{\mathcal{O}}\left(\frac{\log d_u}{\gamma^2\epsilon_{acc}^2}\right)$.*

The procedure described above, although well defined, involves a difficult optimization problem: minimizing the number of L_1-margin violations. In order to obtain a computationally tractable procedure, we consider the hinge-loss instead of the margin error. In a feature space with $||\phi(x)||_\infty \leq 1$ as above, we say that a unit-L_1-norm predictor α, $||\alpha||_1 = 1$, has expected hinge-loss $\mathbf{E}\left[[1 - y(x)\langle\alpha, \phi(x)\rangle/\gamma]_+\right]$ relative to L_1-margin γ. Now, if we know there is some (unknown) predictor with hinge-loss ϵ relative L_1-margin γ, than a predictor with error $\epsilon + \epsilon_{\text{acc}}$ can be learned (with high probability) from a sample of $\tilde{\mathcal{O}}\left(\log d/(\gamma^2\epsilon_{\text{acc}}^2)\right)$ examples by minimizing the empirical average hinge-loss relative to L_1-margin γ on the sample [211].

Before proceeding to discussing the optimization problem of minimizing the average hinge-loss relative to a fixed L_1-margin, let us establish the analogue of Theorem 3.4.3 for the hinge-loss:

Theorem 3.4.6 *Assume that K is an (ϵ, γ, τ)-good similarity function in hinge-loss for a learning problem P. For any $\epsilon_1 > 0$ and $0 < \lambda < \gamma\epsilon_1/4$ let $S = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_d\}$ be a sample of size $d = \frac{2}{\tau}\left(\log(2/\delta) + 16\log(2/\delta)/(\epsilon_1\gamma)^2\right)$ drawn from P. With probability at least $1-\delta$ over the random sample S, the induced distribution $\phi^S(P)$ in R^d, for ϕ^S as defined in Theorem 3.4.3, has a separator achieving hinge-loss at most $\epsilon + \epsilon_1$ at margin γ.*

Proof: We use the same construction as in Theorem 3.4.3. ■

Corollary 3.4.7 *K is an (ϵ, γ, τ)-good similarity function in hinge loss then we can efficiently find a predictor with error at most $\epsilon + \epsilon_{acc}$ from an unlabeled sample of size $d_u = \tilde{\mathcal{O}}\left(\frac{1}{\gamma^2 \epsilon_{acc}^2 \tau}\right)$ and from a labeled sample of size $d_l = \tilde{\mathcal{O}}\left(\frac{\log d_u}{\gamma^2 \epsilon_{acc}^2}\right)$.*

For the hinge-loss, our two stage procedure boils down to solving the following optimization problem w.r.t. α:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{d_l} \left[1 - \sum_{j=1}^{d_u} \alpha_j y(x_i) K(x_i, \tilde{x}_j) \right]_+ \\ \text{s.t.} \quad & \sum_{j=1}^{d_u} |\alpha_j| \leq 1/\gamma \end{aligned} \tag{3.49}$$

This is a linear program and can thus be solved in polynomial time, establishing the efficiency in Corollary 3.4.7.

We can in fact use results in [115] to extend Corollary 3.4.7 a bit and get a better bound as follows:

Corollary 3.4.8 *If K is a $(\epsilon_{acc}/8, \gamma, \tau)$-good similarity function then with high probability we can efficiently find a predictor with error at most ϵ_{acc} from an unlabeled sample of size $d_u = \tilde{\mathcal{O}}\left(\frac{1}{\gamma^2 \tau}\right)$ and from a labeled sample of size $d_l = \tilde{\mathcal{O}}\left(\frac{\log d_u}{\gamma^2 \epsilon_{acc}}\right)$.*

An optimization problem similar to (3.49), though usually with the same set of points used both as landmarks and as training examples, is actually fairly commonly used as a learning rule in practice [54, 127, 185]. Such a learning rule is typically discussed as an alternative to SVMs. In fact, [198] suggest the Relevance Vector Machine (RVM) as a Bayesian alternative to SVMs. The MAP estimate of the RVM is given by an optimization problem similar to (3.49), though with a loss function different from the hinge loss (the hinge-loss cannot be obtained as a log-likelihood). Similarly, [192] suggests Norm-Penalized Leveraging Procedures as a boosting-like approach that mimics SVMs. Again, although the specific loss functions studied by [192] are different from the hinge-loss, the method (with a norm exponent of 1, as in [192]'s experiments) otherwise corresponds to a coordinate-descent minimization of (3.49). In both cases, no learning guarantees are provided.

The motivation for using (3.49) as an alternative to SVMs is usually that the L_1-regularization on α leads to sparsity, and hence to "few support vectors" (although [207], who also discuss (3.49), argue for more direct ways of obtaining such sparsity), and also that the linear program (3.49) might be easier to solve than the SVM quadratic program. However, we are not aware of a previous discussion on how learning using (3.49) relates to learning using a SVM, or on learning guarantees using (3.49) in terms of properties of the similarity function K. Guarantees solely in terms of the feature space in which we seek low L_1-margin (ϕ^S in our notation) are problematic, as this feature space is generated randomly from data.

In fact, in order to enjoy the SVM guarantees while using L_1 regularization to obtain sparsity, some authors suggest regularizing both the L_1 norm $||\alpha||_1$ of the coefficient vector α (as in (3.49)), and the norm $||\beta||$ of the corresponding predictor $\beta = \sum_j \alpha_j \phi(\tilde{x}_j)$ in the Hilbert space implied by K, where $K(x, x') = \langle \phi(x), \phi(x') \rangle$, as when using a SVM with K as a kernel [128, 179].

Here, we provide a natural condition on the similarity function K (Definition 3.4.2), that justifies the learning rule (3.49). Furthermore, we show (in Section 3.4.4) than any similarity function that is good as a kernel, and can ensure SVM learning, is also good as a similarity function and can thus also

ensure learning using the learning rule (3.49) (though possibly with some deterioration of the learning guarantees). These arguments can be used to justify (3.49) as an alternative to SVMs.

Before concluding this discussion, we would like to mention that [118] previously established a rather different connection between regularizing the L_1 norm $||\alpha||_1$ and regularizing the norm of the corresponding predictor β in the implied Hilbert space. [118] considered a hard-margin SVR (Support Vector Regression Machine, i.e. requiring each prediction to be within $(y(x) - \epsilon, y(x) + \epsilon)$), in the noiseless case where the mapping $x \mapsto y(x)$ is in the Hilbert space. In this setting, [118] showed that a hard-margin SVR is equivalent to minimizing the distance *in the implied Hilbert space* between the correct mapping $x \mapsto y(x)$ and the predictions $x \mapsto \sum_j \alpha_j K(x, \tilde{x}_j)$, with an L_1 regularization term $||\alpha||_1$. However, this distance between prediction functions is very different than the objective in (3.49), and again refers back to the implied feature space which we are trying to avoid.

3.4.3 Separation Results

In this Section, we show an example of a finite concept class for which no kernel yields good learning guarantees when used as a kernel, but for which there does exist a good similarity function yielding the optimal sample complexity. That is, we show that some concept classes cannot be reasonably represented by kernels, but can be reasonably represented by similarity functions.

Specifically, we consider a class C of n pairwise uncorrelated functions. This is a finite class of cardinality $|C| = n$, and so if the target belongs to C then $O(\frac{1}{\epsilon} \log n)$ samples are enough for learning a predictor with error ϵ.

Indeed, we show here that for *any* concept class C, so long as the distribution D is sufficiently unconcentrated, there exists a similarity function that is $(0, 1, \frac{1}{2|C|})$-good under our definition for every $f \in C$. This yields a (labeled) sample complexity $O(\frac{1}{\epsilon} \log |C|)$ to achieve error ϵ, matching the ideal sample complexity. In other words, for distribution-specific learning (where unlabeled data may be viewed as free) and finite classes, there is no *intrinsic* loss in sample-complexity incurred by choosing to learn via similarity functions. In fact, we also extend this result to classes of bounded VC-dimension rather than bounded cardinality.

In contrast, we show that if C is a class of n functions that are pairwise uncorrelated with respect to distribution D, then *no* kernel is (ϵ, γ)-good in hinge-loss for all $f \in C$ even for $\epsilon = 0.5$ and $\gamma = 8/\sqrt{n}$. This extends work of [50, 109] who give hardness results with comparable margin bounds, but at a much lower error rate. Thus, this shows there *is* an intrinsic loss incurred by using kernels together with margin bounds, since this results in a sample complexity bound of at least $\Omega(|C|)$, rather than the ideal $O(\log |C|)$.

We thus demonstrate a gap between the kind of prior knowledge can be represented with kernels as opposed to general similarity functions and demonstrate that similarity functions are strictly more expressive (up to the degradation in parameters discussed earlier).

Definition 3.4.3 *We say that a distribution D over X is* α-unconcentrated *if the probability mass on any given $x \in X$ is at most α.*

Theorem 3.4.9 *For any class finite class of functions C and for any $1/|C|$-unconcentrated distribution D over the instance space X, there exists a similarity function K that is a $(0, 1, \frac{1}{2|C|})$-good similarity function for all $f \in C$.*

Proof: Let $C = \{f_1, \ldots, f_n\}$. Now, let us partition X into n regions R_i of at least $1/(2n)$ probability mass each, which we can do since D is $1/n$-unconcentrated. Finally, define $K(x, x')$ for x' in R_i to be $f_i(x)f_i(x')$. We claim that for this similarity function, R_i is a set of "representative points" establishing margin $\gamma = 1$ for target f_i. Specifically,

$$\mathbf{E}[K(x, x')f_i(x)f_i(x')|x' \in R_i] = \mathbf{E}[f_i(x)f_i(x')f_i(x)f_i(x')] = 1.$$

Since $\Pr(R_i) \geq \frac{1}{2n}$, this implies that under distribution D, K is a $(0, 1, \frac{1}{2n})$-good similarity function for all $f_i \in C$.

■

Note 1: We can extend this argument to any class C of small VC dimension. In particular, for any distribution D, the class C has an ϵ-cover C_ϵ of size $(1/\epsilon)^{O(d/\epsilon)}$, where d is the VC-dimension of C [52]. By Theorem 3.4.9, we can have a $(0, 1, 1/|C_\epsilon|)$-good similarity function for the cover C_ϵ, which in turn implies an $(\epsilon, 1, 1/|C_\epsilon|)$-good similarity function for the original set (even in hinge loss since $\gamma = 1$). Plugging in our bound on $|C_\epsilon|$, we get an $(\epsilon, 1, \epsilon^{O(d/\epsilon)})$-good similarity function for C. Thus, the labeled sample complexity we get for learning with similarity functions is only $\mathcal{O}((d/\epsilon)\log(1/\epsilon))$, and again there is no *intrinsic* loss in sample complexity bounds due to learning with similarity functions.

Note 2: The need for the underlying distribution to be unconcentrated stems from our use of this distribution for both labeled and unlabeled data. We could further extend our definition of "good similarity function" to allow for the unlabeled points x' to come from some other distribution D' *given apriori*, such as the uniform distribution over the instance space X. Now, the expectation over x' and the probability mass of R would both be with respect to D', and the generic learning algorithm would draw points x'_i from D' rather than D. In this case, we would only need D' to be unconcentrated, rather than D.

We now prove our lower bound for margin-based learning with kernels.

Theorem 3.4.10 *Let C be a class of n pairwise uncorrelated functions over distribution D. Then, there is no kernel that for all $f \in C$ is (ϵ, γ)-good in hinge-loss even for $\epsilon = 0.5$ and $\gamma = 8/\sqrt{n}$.*

Proof: Let $C = \{f_1, \ldots, f_n\}$. We begin with the basic fourier setup [162, 167]. Given two functions f and g, define $\langle f, g\rangle = \mathbf{E}_x[f(x)g(x)]$ to be their correlation with respect to distribution D. (This is their inner-product if we view f as a vector whose jth coordinate is $f(x_j)[D(x_j)]^{1/2}$). Because the functions $f_i \in C$ are pairwise uncorrelated, we have $\langle f_i, f_j\rangle = 0$ for all $i \neq j$, and because the f_i are boolean functions we have $\langle f_i, f_i\rangle = 1$ for all i. Thus they form at least part of an orthonormal basis, and for any hypothesis h (i.e. any mapping $X \to \{\pm 1\}$) we have

$$\sum_{f_i \in C} \langle h, f_i\rangle^2 \leq 1.$$

So, this implies

$$\sum_{f_i \in C} |\langle h, f_i\rangle| \leq \sqrt{n}.$$

or equivalently

$$\mathbf{E}_{f_i \in C}|\langle h, f_i\rangle| \leq 1/\sqrt{n}. \tag{3.50}$$

In other words, for any hypothesis h, if we pick the target at random from C, the expected magnitude of the correlation between h and the target is at most $1/\sqrt{n}$.

We now consider the implications of having a good kernel. Suppose for contradiction that there exists a kernel K that is $(0.5, \gamma)$-good in hinge loss for every $f_i \in C$. What we will show is this implies that for any $f_i \in C$, the expected value of $|\langle h, f_i\rangle|$ for a *random* linear separator h in the ϕ-space is greater than $\gamma/8$. If we can prove this, then we are done because this implies there must *exist* an h that has $\mathbf{E}_{f_i \in C}|\langle h, f\rangle| > \gamma/8$, which contradicts equation (3.50) for $\gamma = 8/\sqrt{n}$.

So, we just have to prove the statement about random linear separators. Let w^* denote the vector in the ϕ-space that has hinge-loss at most 0.5 at margin γ for target function f_i. For any example x, define γ_x to be the margin of $\phi(x)$ with respect to w^*, and define $\alpha_x = \sin^{-1}(\gamma_x)$ to be the angular

margin of $\phi(x)$ with respect to w^*.[8] Now, consider choosing a random vector h in the ϕ-space, where we associate $h(x) = \text{sign}(\text{h} \cdot \phi(\text{x}))$. Since we only care about the absolute value $|\langle h, f_i\rangle|$, and since $\langle -h, f_i\rangle = -\langle h, f_i\rangle$, it suffices to show that $\mathbf{E}_h[\langle h, f_i\rangle \mid h \cdot w^* \geq 0] > \gamma/8$. We do this as follows.

First, for any example x, we claim that:

$$\Pr_h[(h(x) \neq f_i(x) | h \cdot w^* \geq 0] = 1/2 - \alpha_x/\pi. \tag{3.51}$$

This is because we look at the 2-dimensional plane defined by $\phi(x)$ and w^*, and consider the half-circle of $||h|| = 1$ such that $h \cdot w^* \geq 0$, then (3.51) is the portion of the half-circle that labels $\phi(x)$ incorrectly. Thus, we have:

$$\mathbf{E}_h[err(h) | h \cdot w^* \geq 0] \;=\; \mathbf{E}_x[1/2 - \alpha_x/\pi],$$

and so, using $\langle h, f_i\rangle = 1 - 2\ err(h)$, we have:

$$\mathbf{E}_h[\langle h, f_i\rangle \mid h \cdot w^* \geq 0] \;=\; 2\mathbf{E}_x[\alpha_x]/\pi.$$

Finally, we just need to relate angular margin and hinge loss: if L_x is the hinge-loss of $\phi(x)$, then a crude bound on α_x is

$$\alpha_x \geq \gamma(1 - (\pi/2)L_x).$$

Since we assumed that $\mathbf{E}_x[L_x] \leq 0.5$, we have:

$$\mathbf{E}_x[\alpha_x] \geq \gamma(1 - \pi/4).$$

Putting this together we get expected magnitude of correlation of a random halfspace is at least $2\gamma(1 - \pi/4)/\pi > \gamma/8$ as desired, proving the theorem. ■

An example of a class C satisfying the above conditions is the class of parity functions over $\{0,1\}^{\lg n}$, which are pairwise uncorrelated with respect to the uniform distribution. Note that the uniform distribution is $1/|C|$-unconcentrated, and thus there *is* a good similarity function. (In particular, one could use $K(x_i, x_j) = f_j(x_i)f_j(x_j)$, where f_j is the parity function associated with indicator vector x_j.)

We can extend Theorem 3.4.10 to classes of large Statistical Query dimension as well. In particular, the SQ-dimension of a class C with respect to distribution D is the size d of the largest set of functions $\{f_1, f_2, \ldots, f_d\} \subseteq C$ such that $|\langle f_i, f_j\rangle| \leq 1/d^3$ for all $i \neq j$ [63]. In this case, we just need to adjust the Fourier analysis part of the argument to handle the fact that the functions may not be completely uncorrelated.

Theorem 3.4.11 *Let C be a class of functions of SQ-dimension d with respect to distribution D. Then, there is no kernel that for all $f \in C$ is (ϵ, γ)-good in hinge-loss even for $\epsilon = 0.5$ and $\gamma = 16/\sqrt{d}$.*

Proof: Let $f_1, \ldots, f_d$ be d functions in C such that $|\langle f_i, f_j\rangle| \leq 1/d^3$ for all $i \neq j$. We can define an orthogonal set of functions $f_1', f_2', \ldots, f_d'$ as follows: let $f_1' = f_1$, $f_2' = f_2 - f_1\langle f_2, f_1\rangle$, and in general let f_i' be the portion of f_i orthogonal to the space spanned by $f_1, \ldots, f_{i-1}$. (That is, $f_i' = f_i - \text{proj}(f_i, \text{span}(f_1, \ldots, f_{i-1}))$, where "proj" is orthogonal projection.) Since the f_i' are orthogonal and have length at most 1, for any boolean function h we have $\sum_i \langle h, f_i'\rangle^2 \leq 1$ and therefore $\mathbf{E}_i|\langle h, f_i'\rangle| \leq 1/\sqrt{d}$. Finally, since $\langle f_i, f_j\rangle \leq 1/d^3$ for all $i \neq j$, one can show this implies that $|f_i - f_i'| \leq 1/d$ for all i. So, $\mathbf{E}_i|\langle h, f_i\rangle| \leq 1/\sqrt{d} + 1/d \leq 2/\sqrt{d}$. The rest of the argument in the proof of Theorem 3.4.10 now applies with $\gamma = 16/\sqrt{d}$. ■

[8] So, α_x is a bit larger in magnitude than γ_x. This works in our favor when the margin is positive, and we just need to be careful when the margin in negative.

For example, the class of size-n decision trees over $\{0,1\}^n$ has $n^{\Omega(\log n)}$ pairwise uncorrelated functions over the uniform distribution (in particular, any parity of $\log n$ variables can be written as an n-node decision tree). So, this means we cannot have a kernel with margin $1/poly(n)$ for all size-n decision trees over $\{0,1\}^n$. However, we *can* have a similarity function with margin 1, though the τ parameter (which controls running time) will be exponentially small.

3.4.4 Relation Between Good Kernels and Good Similarity Functions

We start by showing that a kernel good as a similarity function is also good as a kernel. Specifically, if a similarity function K is indeed a kernel, and it is (ϵ, γ, τ)-good as a similarity function (possibly in hinge-loss), than it is also (ϵ, γ)-good as a kernel (respectively, in hinge loss). That is, although the notion of a good similarity function is more widely applicable, for those similarity functions that are positive semidefinite, a good similarity function is also a good kernel.

Theorem 3.4.12 *If K is a valid kernel function, and is (ϵ, γ, τ)-good similarity for some learning problem, then it is also (ϵ, γ)-kernel-good for the learning problem. If K is (ϵ, γ, τ)-good similarity in hinge loss, then it is also (ϵ, γ)-kernel-good in hinge loss.*

Proof: Consider a similarity function K that is a valid kernel, i.e. $K(x, x') = \langle \phi(x), \phi(x') \rangle$ for some mapping ϕ of x to a Hilbert space $\mathcal{H}$. For any input distribution and any probabilistic set of representative points R of the input we will construct a linear predictor $\beta_R \in \mathcal{H}$, with $||\beta_R|| \leq 1$, such that similarity-based predictions using R are the same as the linear predictions made with β_R.

Define the following linear predictor $\beta_R \in \mathcal{H}$:

$$\beta_R = \mathbf{E}_{(x',y',r')}\left[y'\phi(x')|r'=1\right].$$

The predictor β_R has norm at most:

$$\begin{aligned} ||\beta_R|| &= ||\mathbf{E}_{(x',y',r')}\left[y'\phi(x')|r'=1\right]|| \leq \max_{x'} ||y(x')\phi(x')|| \\ &\leq \max ||\phi(x')|| = \max \sqrt{K(x',x')} \leq 1 \end{aligned}$$

where the second inequality follows from $|y(x')| \leq 1$.

The predictions made by β_R are:

$$\begin{aligned} \langle \beta_R, \phi(x) \rangle &= \left\langle \mathbf{E}_{(x',y',r')}\left[y'\phi(x')|r'=1\right], \phi(x) \right\rangle \\ &= \mathbf{E}_{(x',y',r')}\left[y'\left\langle \phi(x'), \phi(x) \right\rangle | r'=1\right] \\ &= \mathbf{E}_{(x',y',r')}\left[y'K(x,x')|r'=1\right] \end{aligned}$$

That is, using β_R is the same as using similarity-based prediction with R. In particular, the margin violation rate, as well as the hinge loss, with respect to any margin γ, is the same for predictions made using either R or β_R. This is enough to establish Theorem 3.4.12: If K is (ϵ, γ)-good (perhaps for to the hinge-loss), there exists some valid R that yields margin violation error rate (resp. hinge loss) at most ϵ with respect to margin γ, and so β_R yields the same margin violation (resp. hinge loss) with respect to the same margin, establishing K is (ϵ, γ)-kernel-good (resp. for the hinge loss). ■

We now show the converse: if a kernel function is good in the kernel sense, it is also good in the similarity sense, though with some degradation of the margin. This degradation is much smaller than the one incurred previously by the Balcan - Blum'06 definitions (and the proofs in [24], [195], and [38]). Specifically, we can show that if K is a $(0, \gamma)$-good kernel, then K is $(\epsilon, \gamma^2, \epsilon)$-good similarity function

for any ϵ (formally, it is $(\epsilon, \gamma^2/c, \epsilon c)$-good for some $c \leq 1$). The proof is based on the following idea. Say we have a good kernel in hinge loss. Then we can choose an appropriate regularization parameter and write a "distributional SVM" such that there exists a solution vector that gets a large fraction of the distribution correct, and moreover, the fraction of support vectors is large enough. Any support vector will then be considered a representative point in our similarity view, and the probability that a point is representative is proportional to α_i, where α_i is dual variable associated with x_i.

To formally prove the desired result, we introduce an intermediate notion of a good similarity function.

Definition 3.4.4 (Intermediate, Margin Violations) *A similarity function K is a* **relaxed (ϵ, γ, M)-good similarity function** *for a learning problem P if there* exists *a bounded weighting function w over X, $w(x') \in [0, M]$ for all $x' \in X$, $\mathbf{E}_{x' \sim P}[w(x')] \leq 1$ such that at least a $1 - \epsilon$ probability mass of examples x satisfy:*

$$\mathbf{E}_{x' \sim P}[y(x)y(x')w(x')K(x, x')] \quad \geq \quad \gamma. \tag{3.52}$$

Definition 3.4.5 (Intermediate, Hinge Loss) *A similarity function K is a* **relaxed (ϵ, γ, M)-good similarity function in hinge loss** *for a learning problem P if there exists a weighting function $w(x') \in [0, M]$ for all $x' \in X$, $\mathbf{E}_{x' \sim P}[w(x')] \leq 1$ such that*

$$\mathbf{E}_x \Big[[1 - y(x)g(x)/\gamma]_+ \Big] \leq \epsilon, \tag{3.53}$$

where $g(x) = \mathbf{E}_{x' \sim P}[y(x')w(x')K(x, x')]$ is the similarity-based prediction made using $w(\cdot)$.

These intermediate definitions are closely related to our main similarity function definitions: in particular, if K is a relaxed (ϵ, γ, M)-good similarity function for a learning problem P, then it is also an $(\epsilon, \gamma/c, c/M)$-good similarity function for some $\gamma \leq c \leq 1$.

Theorem 3.4.13 *If K is a relaxed (ϵ, γ, M)-good similarity function for a learning problem P, then there exists $\gamma \leq c \leq 1$ such that K is a $(\epsilon, \gamma/c, c/M)$-good similarity function for P. If K is a relaxed (ϵ, γ, M)-good similarity function in hinge loss for P, then there exists $\gamma \leq c \leq 1$ such that K is a $(\epsilon, \gamma/c, c/M)$-good similarity function for P.*

Proof: First, divide $w(x)$ by M to scale its range to $[0, 1]$, so $\mathbf{E}[w] = c/M$ for some $c \leq 1$ and the margin is now γ/M. Define random indicator $R(x')$ to equal 1 with probability $w(x')$ and 0 with probability $1 - w(x')$, and let the extended probability P over $X \times Y \times \{0, 1\}$ be defined as in Equations 3.40 or 3.41.

We have

$$\tau = \Pr_{(x', y', r')}[r' = 1] = \mathbf{E}_{x'}[w(x')] = c/M,$$

and we can rewrite (3.52) as

$$\mathbf{E}_{(x', y', r')}[y(x)y' I(r' = 1)K(x, x')] \quad \geq \quad \gamma/M. \tag{3.54}$$

Finally, divide both sides of (3.54) by $\tau = c/M$, producing the conditional $\mathbf{E}_{(x', y', r')}[y(x)y(x')K(x, x') \mid r' = 1]$ on the LHS and a margin of γ/c on the RHS. The case of hinge-loss is identical. ■

Note that since our guarantees for (ϵ, γ, τ)-good similarity functions depend on τ only through $\gamma^2\tau$, a decrease in τ and a proportional increase in γ (as when $c < 1$ in Theorem 3.4.13) only improves the guarantees. However, allowing flexibility in this tradeoff will make the kernel-to-similarity function translation much easier.

We will now establish that a similarity function K that is good as a kernel, is also good as a similarity function in this intermediate sense, and hence, by Theorem 3.4.13, also in our original sense. We begin by considering goodness in hinge-loss, and will return to margin violations at the end of the Section.

Theorem 3.4.14 *If K is (ϵ_0, γ)-good kernel in hinge loss for learning problem (with deterministic labels), then it is also a relaxed $(\epsilon_0 + \epsilon_1, \frac{\gamma^2}{1+\epsilon_0/2\epsilon_1}, \frac{1}{2\epsilon_1+\epsilon_0})$-good similarity in hinge loss for the learning problem, for any $\epsilon_1 > 0$.*

Proof: We initially only consider finite discrete distributions, where:

$$\Pr(\, x_i, y_i \,) = p_i \tag{3.55}$$

for $i = 1 \ldots n$, with $\sum_{i=1}^{n} p_i = 1$ and $x_i \neq x_j$ for $i \neq j$.

Let K be any kernel function that is (ϵ_0, γ)-kernel good in hinge loss. Let ϕ be the implied feature mapping and denote $\phi_i = \phi(x_i)$. Consider the following weighted-SVM quadratic optimization problem with regularization parameter C:

$$\text{minimize } \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n} p_i[1 - y_i\langle\beta, \phi_i\rangle]_+ \tag{3.56}$$

The dual of this problem, with dual variables α_i, is:

$$\begin{aligned} \text{maximize } & \sum_i \alpha_i - \frac{1}{2}\sum_{ij} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \text{subject to } & \quad 0 \leq \alpha_i \leq C p_i \end{aligned} \tag{3.57}$$

There is no duality gap, and furthermore the primal optimum β^* can be expressed in terms of the dual optimum α^*: $\beta^* = \sum_i \alpha_i^* y_i \phi_i$.

Since K is (ϵ_0, γ)-kernel-good in hinge-loss, there exists a predictor $||\beta_0|| = 1$ with average-hinge loss ϵ_0 relative to margin γ. The primal optimum β^* of (3.56), being the optimum solution, then satisfies:

$$\begin{aligned} \frac{1}{2}||\beta^*||^2 + C\sum_i p_i[1 - y_i\langle\beta^*, \phi_i\rangle]_+ &\leq \frac{1}{2}||\frac{1}{\gamma}\beta_0||^2 + C\sum_i p_i[1 - y_i\left\langle\frac{1}{\gamma}\beta_0, \phi_i\right\rangle]_+ \\ &= \frac{1}{2\gamma^2} + C\mathbf{E}\left[[1 - y\left\langle\frac{1}{\gamma}\beta_0, \phi(x)\right\rangle]_+\right] \\ &= \frac{1}{2\gamma^2} + C\epsilon_0 \end{aligned}$$

Since both terms on the left hand side are non-negative, each of them is bounded by the right hand side, and in particular:

$$C\sum_i p_i[1 - y_i\langle\beta^*, \phi_i\rangle]_+ \leq \frac{1}{2\gamma^2} + C\epsilon_0$$

Dividing by C we get a bound on the average hinge-loss of the predictor β^*, relative to a margin of one:

$$\mathbf{E}[[1 - y\langle\beta^*, \phi(x)\rangle]_+] \leq \frac{1}{2C\gamma^2} + \epsilon_0 \tag{3.58}$$

We now use the fact that β^* can be written as $\beta^* = \sum_i \alpha_i^* y_i \phi_i$ with $0 \leq \alpha_i^* \leq C p_i$. Let us consider the weights

$$w_i = w(x_i) = \alpha_i^*/(A p_i) \tag{3.59}$$

So, $w_i \leq \frac{C}{A}$ and $\mathbf{E}[w] = \frac{\sum_i \alpha_i^*}{A}$. Furthermore, since we have no duality gap we also have

$$\sum_i \alpha_i^* - \frac{1}{2}||\beta^*||^2 = \frac{1}{2}||\beta^*||^2 + C\sum_i p_i[1 - y_i\langle\beta^*, \phi_i\rangle]_+,$$

so $\sum_i \alpha_i^* \leq \frac{1}{\gamma^2} + C\epsilon_0$.
So, we have for every x, y:

$$\begin{aligned} y\mathbf{E}_{x',y'}\left[w(x')y'K(x,x')\right] &= y\sum_i p_i w(x_i)y_i K(x,x_i) \\ &= y\sum_i p_i \alpha_i^* y_i K(x,x_i)/(Ap_i) \\ &= y\sum_i \alpha_i^* y_i \langle\phi_i, \phi(x)\rangle/A \\ &= y\langle\beta^*, \phi(x)\rangle/A \end{aligned}$$

Multiplying by A and using (3.58):

$$\mathbf{E}_{x,y}\left[\,[\,1 - Ay\mathbf{E}_{x',y'}\left[w(x')y'K(x,x')\right]\,]_+\,\right] = \mathbf{E}_{x,y}[\,[\,1 - y\langle\beta^*, \phi(x)\rangle\,]_+\,] \leq \frac{1}{2C\gamma^2} + \epsilon_0 \quad (3.60)$$

Since $w_i \leq \frac{C}{A}$, $\mathbf{E}[w] = \frac{\sum_i \alpha_i^*}{A}$, and $\sum_i \alpha_i^* \leq \frac{1}{\gamma^2} + C\epsilon_0$, and we want $\mathbf{E}[w] \leq 1$, we need to impose that $\left(\frac{1}{\gamma^2} + C\epsilon_0\right)\frac{1}{A} \leq 1$. We also want $w_i \in [0, M]$, so we also have the constraint $\frac{C}{A} \leq M$. Choosing $M = \frac{1}{2\epsilon_1+\epsilon_0}$, $A = \frac{1+\epsilon_0/2\epsilon_1}{\gamma^2}$, and $C = 1/(2\epsilon_1\gamma^2)$ we get an average hinge-loss of $\epsilon_0 + \epsilon_1$ at margin $1/A$

$$\mathbf{E}_{x,y}\left[\,[\,1 - y\mathbf{E}_{x',y'}\left[w(x')y'K(x,x')\right]/(1/A)\,]_+\,\right] \leq \epsilon_0 + \epsilon_1 \quad (3.61)$$

as desired. This establishes that if K is (ϵ_0, γ)-good kernel in hinge loss then it is also a relaxed $(\epsilon_0 + \epsilon_1, \frac{\gamma^2}{1+\epsilon_0/2\epsilon_1}, \frac{1}{2\epsilon_1+\epsilon_0})$-good similarity in hinge loss, for any $\epsilon_1 > 0$, at least for finite discrete distributions.

To extend the result also to non-discrete distributions, we can consider the variational "infinite SVM" problem and apply the same arguments, as in [195] and in Section 3.3. ■

Interpretation The proof of theorem 3.4.14 shows the following. Assume that K is $(0, \gamma)$-good kernel. Assume that τ is our desired error rate. Then we can choose a regularization parameter $C = 1/(2\gamma^2 \cdot \tau)$ for the "distributional SVM" (Eq. 3.56) such that there exists a solution vector that gets a $(1 - \tau)$ fraction of the distribution correct, and moreover, the number of support vectors is at least $\gamma^2 \cdot \tau$ fraction of the whole distribution; so, we do end up spread out a bit the support vectors of the SVM in Eq. 3.56. Any support vector will then be considered a representative point in our similarity view, and the probability that a point is representative is proportional to α_i/p_i.

Note however that if the K is a good kernel, then there might exist multiple different good sets of representative points ; and the argument in theorem 3.4.14 shows the existence of such a set based on an SVM argument. [9]

We can now use the hinge-loss correspondence to get a similar result for the margin-violation definitions:

[9] In fact, the original proof that a good kernel is a good similarity function in the Balcan - Blum'06 sense which appeared in [24] was based on a different Perceptron based argument.

Theorem 3.4.15 *If K is (ϵ_0, γ)-good kernel for a learning problem (with deterministic labels), then it is also a relaxed $(\epsilon_0 + \epsilon_1, \gamma^2/2, \frac{1}{(1-\epsilon_0)\epsilon_1})$-good similarity function for the learning problem, for any $\epsilon_1 > 0$.*

Proof: If K is $(0, \gamma)$-good as a kernel, it is also $(0, \gamma)$ good as a kernel in hinge loss, and we can apply Theorem 3.4.14 to obtain that K is also $(\epsilon_0/2, \gamma_1, \tau_1)$-good, where $\gamma_1 = \gamma^2$ and $\tau_1 = 1/\epsilon_1$. We can then bound the number of margin violations at $\gamma_2 = \gamma_1/2$ by half the hinge loss at margin γ_1 to obtain the desired result.

If K is only (ϵ, γ)-good as a kernel, we follow a similar procedure to that described in [195] and in Section 3.3, and consider a distribution conditioned only on those places where there is no error. Returning to the original distribution, we must scale the weights up by an amount proportional to the probability of the event we conditioned on (i.e. the probability of no margin violation). This yields the desired bound. ■

Note: We also note that if we want our Definitions 3.4.1 and Definition 3.4.2 to include the usual notions of good kernel functions, we do need to allow the set $\{x : R(x) = 1\}$ to be probabilistic. To see this, let us consider the following example.

$$\begin{aligned}
x_1 &= (\sqrt{1-\gamma^2}, \gamma), & y_1 &= 1, & p_1 &= \frac{1}{2} - \epsilon \\
x_2 &= (-\sqrt{1-\gamma^2}, \gamma,), & y_2 &= 1, & p_2 &= \epsilon \\
x_3 &= (\sqrt{1-\gamma^2}, -\gamma,), & y_3 &= -1, & p_3 &= \epsilon \\
x_4 &= (-\sqrt{1-\gamma^2}, -\gamma), & y_4 &= -1, & p_4 &= \frac{1}{2} - \epsilon
\end{aligned}$$

for some (small) $0 < \gamma < \sqrt{\frac{1}{2}}$ and (small) probability $0 < \epsilon < \frac{1}{2}$. The four points are all on the unit sphere (i.e. $||x_i|| = 1$ and so $K(x_i, x_j) = \langle x_i, x_j \rangle \leq 1$), and are clearly separated by $\beta = (0, 1)$ with a margin of γ. The standard inner-product kernel is therefore $(0, \gamma)$-kernel-good on this distribution. Note however that for any τ, in order to get K to be a $(0, \gamma, \tau)$-good similarity function we need to allow R to be probabilistic. This can be easily verified by a case analysis. Clearly we cannot have R contain just one point. Also, we cannot R be only $\{x_1, x_4\}$ since x_3 will fail to satisfy the condition. Similarly wrt $\{x_2, x_3\}$. Other cases can be easily verified as well.

One can use the same example in order to show that we need to consider $w(x) \in [0, 1]$ rather than $w \in \{0, 1\}$ in the context Definitions 3.3.5 and Definitions 3.3.6.

3.4.5 Tighteness

We show here that in fact we need to allow $O(\gamma^2)$ loss in the kernel to similarity translation. Specifically:

Theorem 3.4.16 (Tightness, Margin Violations) *For any ε, τ, and γ there exists a learning problem and a kernel function K, which is $(0, \gamma)$-kernel-good for the learning problem, but which cannot be $(\varepsilon, \tilde{\gamma}, \tau)$-good similarity for $\tilde{\gamma} \geq 2\gamma^2$.*

Proof: Assume that $X \in R^d$, for $d \geq \frac{1}{\gamma^2}$. Assume that x_i has all coordinates 0 except for coordinates 1 and i which are set to $y(x_i)\gamma$ and $\sqrt{1-\gamma^2}$, respectively. It is easy to verify that the standard inner-product kernel is a $(0, \gamma)$-kernel-good on this distribution – it is separated by $\beta = (1, 0, \ldots, 0)$ with a margin of γ. We also clearly have $|K(x_i, x_j)| \leq \gamma^2$ for all $i \neq j$ and $K(x, x) = 1$ for all x, which implies that $\mathbf{E}_{(x', y', r') \sim P}[yy' K(x, x') \mid r' = 1] \leq 2\gamma^2$ for any extended distribution $P(x, y, r)$. This then implies the desired conclusion. ■

Theorem 3.4.17 (Tightness, Hinge Loss) *For any $\varepsilon \leq \frac{1}{2}$, τ, and γ there exists a learning problem and a kernel function K, which is $(0, \gamma)$-kernel-good in hinge loss for the learning problem, but which cannot be $(\varepsilon, \tilde{\gamma}, \tau)$-good similarity in hinge loss for $\tilde{\gamma} \geq 4\gamma^2$.*

Proof: The same example as in Theorem 3.4.16 gives us the desired conclusion.

Let $g(x) = \mathbf{E}_{(x',y',r')}[y'K(x,x') \mid r' = 1]$ be defined as in Definition 3.4.2. We clearly have $g(x) \in [-2\gamma^2, 2\gamma^2]$. So, clearly for $\tilde{\gamma} \geq 4\gamma^2$ we have $[1 - y(x)g(x)/\tilde{\gamma}]_+ \geq [1 - y(x)\gamma^2/(2\gamma^2)]_+ \geq 1/2$. This then implies the desired conclusion. ■

3.4.6 Learning with Multiple Similarity Functions

We consider here as in Section 3.3.4 the case of learning with multiple similarity functions. Suppose that rather than having a single similarity function, we were instead given n functions $K_1, ..., K_n$, and our hope is that some convex combination of them will satisfy Definition 3.4.1. Is this sufficient to be able to learn well? The following generalization of Theorem 3.4.3 shows that this is indeed the case. (The analog of Theorem 3.4.6 can be derived similarly.)

Theorem 3.4.18 *Suppose $K_1, \ldots, K_n$ are similarity functions such that some (unknown) convex combination of them is (ϵ, γ, τ)-good. For any $\delta > 0$, let $S = \{x'_1, x'_2, \ldots, x'_d\}$ be a sample of size $d = 16\frac{\log(1/\delta)}{\tau\gamma^2}$ drawn from P. Consider the mapping $\phi^S : X \to R^{nd}$ defined as follows: $\phi^S{}_i(x) = (K_1(x, x'_1), \ldots, K_n(x, x'_1), \ldots, K_1(x, x'_d), \ldots, K_n(x, x'_d))$.*

With probability at least $1 - \delta$ over the random sample S, the induced distribution $\phi^S(P)$ in R^{nd} has a separator of error at most $\epsilon + \delta$ at L_1, L_∞ margin at least $\gamma/2$.

Proof: Let $K = \alpha_1 K_1 + \ldots + \alpha_n K_n$ be an (ϵ, γ, τ)-good convex-combination of the K_i. By Theorem 3.4.3, had we instead performed the mapping: $\tilde{\phi}^S : X \to R^d$ defined as

$$\tilde{\phi}^S(x) = (K(x, \tilde{x}_1), \ldots, K(x, \tilde{x}_d)),$$

then with probability $1 - \delta$, the induced distribution $\tilde{\phi}^S(P)$ in R^d would have a separator of error at most $\epsilon + \delta$ at margin at least $\gamma/2$. Let $\hat{\beta}$ be the vector corresponding to such a separator in that space. Now, let us convert $\hat{\beta}$ into a vector in R^{nd} by replacing each coordinate $\hat{\beta}_j$ with the n values $(\alpha_1\hat{\beta}_j, \ldots, \alpha_n\hat{\beta}_j)$. Call the resulting vector $\tilde{\beta}$. Notice that by design, for any x we have $\left\langle \tilde{\beta}, \phi^S(x) \right\rangle = \left\langle \hat{\beta}, \tilde{\phi}^S(x) \right\rangle$. Furthermore, $||\tilde{\beta}||_1 = ||\hat{\beta}||_1$. Thus, the vector $\tilde{\beta}$ under distribution $\phi^S(P)$ has the same properties as the vector $\hat{\beta}$ under $\tilde{\phi}^S(P)$. This implies the desired result. ■

Note that we get significantly better bounds here than in Section 3.3.4 and in [24], since the margin does not drop by a factor of $\frac{1}{\sqrt{n}}$ since we use an L_1 based learning algorithm.

3.5 Connection to the Semi-Supervised Learning Setting

We discuss here how we can connect the framework in this chapter with the Semi-Supervised Learning model in Chapter 2. The approach here does have a similar flavor to the approach in in Chapter 2, however, at a technical level, the final guarantees and learning procedures are somewhat different.

Given a similarity function K let us define C_K as the set of functions of the form

$$f_\alpha = \sum_{x_i \in X} \alpha(x_i) K(\cdot, x_i).$$

Clearly, in general C_K may have infinite capacity.[10] Our assumptions on the similarity function, e.g., the assumption in Definition 3.3.5 can be interpreted as saying that the target function has unlabeled error ϵ at margin γ, where the unlabeled error rate of a function f_α specified by coefficients $\alpha(x_i)$ is defined as

$$err_{unl}(f_\alpha) = 1 - \chi(f_\alpha, P) = \Pr\left[\left|\mathbf{E}_{x'}[K(x,x')\alpha(x')]\right| \leq \gamma\right].$$

Note that here we can define $\chi(f_\alpha, x) = 1$ if $|\mathbf{E}_{x'}[K(x,x')\alpha(x')]| \leq \gamma$ and 0 otherwise.

Let us define $P_\chi = P_{|x:\chi(f_\alpha,x)=1}$ and let $d_\chi(f,g) = \Pr_{x\sim P_\chi}[f(x) \neq g(x)]$. What we are effectively doing in Section 3.4 is the following. Given a fixed γ, we extract a $(\delta, \delta/2)$-randomized approximate cover of C_K with respect to distance d_χ. [11] In particular, the guarantee we get is that for any function f_α with probability at least $1 - \delta/2$, we can find a function $\tilde{f}_\alpha$ in the cover such that $d_\chi(f_\alpha, \tilde{f}_\alpha) \leq \delta$. Since K is (ϵ, γ)-good in the sense of Definition 3.3.5, it follows that there exist a function f_α such that $err_{unl}(f_\alpha) + err_\chi(f_\alpha) \leq \epsilon$, where

$$err_\chi(f_\alpha) = \Pr_{x\sim P_\chi}[f(x) \neq y(x)] \Pr_x[\chi(f,x) = 1].$$

Since we extract a $(\delta, \delta/2)$-randomized approximate cover of C_K, it follows that with high probability, at least $1 - \delta/2$, we can find a function $\tilde{f}_\alpha$ such that $err(\tilde{f}_\alpha) \leq err_{unl}(f_\alpha) + err_\chi(f_\alpha) + \delta$. Once we have constructed the randomized approximate cover, we then in a second stage use labeled examples to learn well.

So, in the case studied in this chapter, the hypothesis space may have an *infinite capacity* before performing the inference. In the training process, in a first stage, we first use unlabeled in order to extract a much smaller set of functions with the property that with high probability the target is well approximated by one the functions in the smaller class. In a second stage we then use labeled examples to learn well. (Note that our compatibility assumption implies an upper bound on the best labeled error we could hope for.)

For the hinge loss definition 3.3.6, we need to consider a cover according to the distance

$$d_\chi(f,g) = \mathbf{E}[|f(x) - g(x)|/\gamma.$$

3.6 Conclusions

The main contribution of this chapter is to develop a theory of learning with similarity functions—namely, of when a similarity function is good for a given learning problem—that is more general and in terms of more tangible quantities than the standard theory of kernel functions. We provide a definition that we show is both sufficient for learning and satisfied by the usual large-margin notion of a good kernel. Moreover, the similarity properties we consider do not require reference to implicit high-dimensional spaces nor do they require that the similarity function be positive semi-definite. In this way, we provide the first rigorous explanation showing why a kernel function that is good in the large-margin sense can also formally be viewed as a good similarity function, thereby giving formal justification to the standard intuition about kernels. We prove that our main notion of a "good similarity function" is strictly more powerful than the traditional notion of a large-margin kernel. This notion relies upon L_1 regularized learning, and our

[10] By capacity of a set of functions here we mean a distribution independent notion of dimension of the given set of functions, e.g., VC-dimension.

[11] Given a class of functions C, we define an (α, β)-cover of C with respect to distance d to be a probability distribution over sets of functions $\tilde{C}$ such that for any $f \in C$ with probability at least $1 - \alpha$, the randomly chosen $\tilde{C}$ from the distribution contains $\tilde{f}$ such that $d(f, \tilde{f}) \leq \beta$.

separation result is related to a separation result between what is learnable with L_1 vs. L_2 regularization. In a lower bound of independent interest, we show that if C is a class of n pairwise uncorrelated functions, then *no* kernel is (ϵ, γ)-good in hinge-loss for all $f \in C$ even for $\epsilon = 0.5$ and $\gamma = 8/\sqrt{n}$.

From a practical perspective, the results of Section 3.3 and 3.4 suggest that if K is in fact a valid kernel, we are probably better off using it as a kernel, e.g. in an SVM or Perceptron algorithm, rather than going through the transformation of Section 3.3.3. However, faced with a non-positive-semidefinite similarity function (coming from domain experts), the transformation of Theorem 3.3.3 might well be useful. In fact, Liao and Noble have used an algorithm similar to the one we propose in the context of protein classification [160]. Furthermore, a direct implication of our results is that we can indeed think (in the design process) of the usefulness of a kernel function in terms of more intuitive, direct properties of the data in the original representation, without need to refer to implicit spaces.

Finally, our algorithms (much like those of [31]) suggest a natural way to use kernels or other similarity functions in learning problems for which one also wishes to use the native features of the examples. For instance, consider the problem of classifying a stream of documents arriving one at a time. Rather than running a kernelized learning algorithm, one can simply take the native features (say the words in the document) and augment them with additional features representing the similarity of the current example with each of a pre-selected set of initial documents. One can then feed the augmented example into a standard unkernelized online learning algorithm. It would be interesting to explore this idea further.

It would be interesting to explore whether the lower bound could be extended to cover *margin violations* with a constant error rate $\epsilon > 0$ rather than only hinge-loss. In addition, it would be particularly interesting to develop even broader natural notions of good similarity functions, that allow for functions that are not positive-semidefinite and yet provide even better kernel-to-similarity translations (e.g., not squaring the margin parameter).

Subsequent Work: Inspired by our work in [24], Wang et. al [208] have recently analyzed different, alternative sufficient conditions for learning via pairwise functions. In particular, Wang et. al [208] analyze unbounded dissimilarity functions which are invariant to order preserving transformations. They provide conditions that they prove are sufficient for learning, though they may not include all good kernel functions.

On a different line of inquiry, we have used this approach [40] for analyzing similarity functions in the context of *clustering* (i.e. learning from purely *unlabeled* data). Specifically, in [40] we ask what (stronger) properties would be sufficient to allow one to produce an accurate hypothesis without any label information at all. We show that if one relaxes the objective (for example, allows the algorithm to produce a hierarchical clustering such that some pruning is close to the correct answer), then one can define a number of interesting graph-theoretic and game-theoretic properties of similarity functions that are sufficient to cluster well. We present this in detail in Chapter 4.

Chapter 4

A Discriminative Framework for Clustering via Similarity Functions

Problems of clustering data from pairwise similarity information are ubiquitous in Machine Learning and Computer Science. Theoretical treatments often view the similarity information as ground-truth and then design algorithms to (approximately) optimize various graph-based objective functions. However, in most applications, this similarity information is merely based on some heuristic; the ground truth is really the unknown correct clustering of the data points and the real goal is to achieve low error on the data. In this work, we develop a theoretical approach to clustering from this perspective. In particular, motivated by our work in Chapter 3 that asks "what natural properties of a similarity (or kernel) function are sufficient to be able to learn well?" we ask "what natural properties of a similarity function are sufficient to be able to *cluster* well?"

To study this question we develop a theoretical framework that can be viewed as an analog for clustering of the discriminative models for Supervised classification (i.e., the Statistical Learning Theory framework and the PAC learning model), where the object of study, rather than being a concept class, is a class of (concept, similarity function) pairs, or equivalently, a *property* the similarity function should satisfy with respect to the ground truth clustering. Our notion of property is similar to the large margin property for a kernel or the properties given in Definitions 3.3.1, 3.3.5, 3.3.6, 3.4.1 or 3.4.2 for supervised learning, though we will need to consider stronger conditions since we have no labeled data.

We then analyze both algorithmic and information theoretic issues in our model. While quite strong properties are needed if the goal is to produce a single approximately-correct clustering, we find that a number of reasonable properties are sufficient under two natural relaxations: (a) list clustering: analogous to the notion of list-decoding, the algorithm can produce a small list of clusterings (which a user can select from) and (b) hierarchical clustering: the algorithm's goal is to produce a hierarchy such that desired clustering is some pruning of this tree (which a user could navigate). We develop a notion of the *clustering complexity* of a given property (analogous to the notion of ϵ-cover examined in Chapter 2), that characterizes its information-theoretic usefulness for clustering. We analyze this quantity for several natural game-theoretic and learning-theoretic properties, as well as design new efficient algorithms that are able to take advantage of them. Our algorithms for hierarchical clustering combine recent learning-theoretic approaches with linkage-style methods. We also show how our algorithms can be extended to the inductive case, i.e., by using just a constant-sized sample, as in property testing. The analysis here uses regularity-type results of [113] and [14].

4.1 Introduction

Clustering is an important problem in the analysis and exploration of data. It has a wide range of applications in data mining, computer vision and graphics, and gene analysis. It has many variants and formulations and it has been extensively studied in many different communities.

In the Algorithms literature, clustering is typically studied by posing some objective function, such as k-median, min-sum or k-means, and then developing algorithms for approximately optimizing this objective given a data set represented as a weighted graph [83, 138, 146]. That is, the graph is viewed as "ground truth" and then the goal is to design algorithms to optimize various objectives over this graph. However, for most clustering problems such as clustering documents by topic or clustering web-search results by category, ground truth is really the unknown true topic or true category of each object. The construction of the weighted graph is just done using some heuristic: e.g., cosine-similarity for clustering documents or a Smith-Waterman score in computational biology. In all these settings, the goal is really to produce a clustering that is as accurate as possible on the data. Alternatively, methods developed both in the algorithms and in the machine learning literature for learning mixtures of distributions [8, 19, 91, 95, 147, 205] explicitly have a notion of ground-truth clusters which they aim to recover. However, such methods are based on very strong assumptions: they require an embedding of the objects into R^n such that the clusters can be viewed as distributions with very specific properties (e.g., Gaussian or log-concave). In many real-world situations (e.g., clustering web-search results by topic, where different users might have different notions of what a "topic" is) we can only expect a domain expert to provide a notion of similarity between objects that is related in some reasonable ways to the desired clustering goal, and not necessarily an embedding with such strong properties.

In this work, we develop a theoretical study of the clustering problem from this perspective. In particular, motivated by our work on similarity functions presented in Chapter 3 that asks "what natural properties of a given kernel (or similarity) function K are sufficient to allow one to *learn* well?" [24, 31, 133, 187, 190] we ask the question "what natural properties of a pairwise similarity function are sufficient to allow one to *cluster* well?" To study this question we develop a theoretical framework which can be thought of as a discriminative (PAC style) model for clustering, though the basic object of study, rather than a concept class, is a *property* of the similarity function K in relation to the target concept much like the types of properties stated in Chapter 3.

The main difficulty that appears when phrasing the problem in this general way is that if one defines success as outputting *a single clustering* that closely approximates the correct clustering, then one needs to assume very strong conditions on the similarity function. For example, if the function provided by our expert is extremely good, say $K(x, y) > 1/2$ for all pairs x and y that should be in the same cluster, and $K(x, y) < 1/2$ for all pairs x and y that should be in different clusters, then we could just use it to recover the clusters in a trivial way.[1] However, if we just slightly weaken this condition to simply require that all points x are more similar to all points y from their own cluster than to any points y from any other clusters, then this is no longer sufficient to uniquely identify even a good approximation to the correct answer. For instance, in the example in Figure 4.1, there are multiple clusterings consistent with this property. Even if one is told the correct clustering has 3 clusters, there is no way for an algorithm to tell which of the two (very different) possible solutions is correct. In fact, results of Kleinberg [151] can be viewed as effectively ruling out a broad class of scale-invariant properties such as this one as being sufficient for producing the correct answer.

[1] Correlation Clustering can be viewed as a relaxation that allows *some* pairs to fail to satisfy this condition, and the algorithms of [11, 66, 84, 197] show this is sufficient to cluster well if the number of pairs that fail is small. *Planted partition* models [13, 92, 169] allow for many failures so long as they occur at *random*. We will be interested in much more drastic relaxations, however.

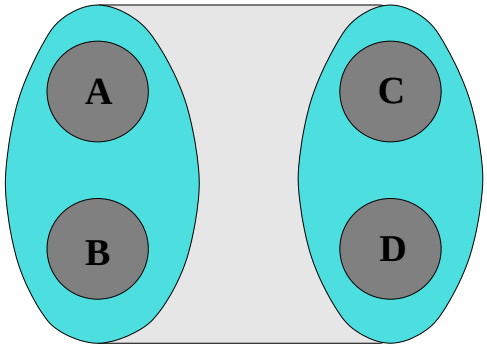

Figure 4.1: Data lies in four regions A, B, C, D (e.g., think of as documents on baseball, football, TCS, and AI). Suppose that $K(x, y) = 1$ if x and y belong to the same region, $K(x, y) = 1/2$ if $x \in A$ and $y \in B$ or if $x \in C$ and $y \in D$, and $K(x, y) = 0$ otherwise. Even assuming that all points are more similar to other points in their own cluster than to any point in any other cluster, there are still multiple consistent clusterings, including two consistent 3-clusterings (($A \cup B$, C, D) or (A, B, $C \cup D$)). However, there is a single hierarchical decomposition such that any consistent clustering is a pruning of this tree.

In our work we overcome this problem by considering two relaxations of the clustering objective that are natural for many clustering applications. The first is as in list-decoding to allow the algorithm to produce a small *list* of clusterings such that at least one of them has low error. The second is instead to allow the clustering algorithm to produce a *tree* (a hierarchical clustering) such that the correct answer is approximately some pruning of this tree. For instance, the example in Figure 4.1 has a natural hierarchical decomposition of this form. Both relaxed objectives make sense for settings in which we imagine the output being fed to a user who will then decide what she likes best. For example, with the tree relaxation, we allow the clustering algorithm to effectively say: "I wasn't sure how specific you wanted to be, so if any of these clusters are too broad, just click and I will split it for you." We then show that with these relaxations, a number of interesting, natural learning-theoretic and game-theoretic properties can be defined that each are sufficient to allow an algorithm to cluster well.

At the high level, our framework has two goals. The first is to provide advice about what type of *algorithms* to use given certain beliefs about the relation of the similarity function to the clustering task. That is, if a domain expert handed us a similarity function that they believed satisfied a certain natural property with respect to the true clustering, what algorithm would be most appropriate to use? The second goal is providing advice to the *designer* of a similarity function for a given clustering task (such as clustering web-pages by topic). That is, if a domain expert is trying up to come up with a similarity measure, what properties should they aim for?

4.1.1 Perspective

The standard approach in theoretical computer science to clustering is to choose some objective function (e.g., k-median) and then to develop algorithms that approximately optimize that objective [83, 100, 138, 146]. If the true goal is to achieve low error with respect to an underlying correct clustering (e.g., a user's desired clustering of search results by topic), however, then one can view this as implicitly making the strong assumption that not only does the correct clustering have a good objective value, but also that all clusterings that approximately optimize the objective must be close to the correct clustering as well. In this work, we instead explicitly consider the goal of producing a clustering of low error and then ask what natural properties of the similarity function in relation to the target clustering are sufficient to allow an

algorithm to do well.

In this respect we are closer to work done in the area of clustering or learning with mixture models [8, 19, 95, 147, 205]. That work, like ours, has an explicit notion of a correct ground-truth clustering of the data points and to some extent can be viewed as addressing the question of what properties of an *embedding of data into* R^n would be sufficient for an algorithm to cluster well. However, unlike our focus, the types of assumptions made are distributional and in that sense are much more stringent than the types of properties we will be considering. This is similarly the case with work on planted partitions in graphs [13, 92, 169]. Abstractly speaking, this view of clustering parallels the *generative* classification setting [103], while the framework we propose parallels the *discriminative* classification setting (i.e. the PAC model of Valiant [201] and the Statistical Learning Theory framework of Vapnik [203] and the setting used in Chapters 2, 3, 6 and 5 of this thesis).

In the PAC model for learning [201], the basic object of study is the *concept class*, and one asks what natural classes are efficiently learnable and by what algorithms. In our setting, the basic object of study is *property*, which can be viewed as a set of (concept, similarity function) pairs, i.e., the pairs for which the target concept and similarity function satisfy the desired relation. As with the PAC model for learning, we then ask what natural properties are sufficient to efficiently cluster well (in either the tree or list models) and by what algorithms. Note that an alternative approach in clustering is to pick some specific *algorithm* (e.g., k-means, EM) and analyze conditions for that algorithm to "succeed". While there is also work in classification of that type (e.g., when does some heuristic like ID3 work well), another important aspect is in understanding which classes of functions are learnable and by what algorithms. We study the analogous questions in the clustering context: what properties are sufficient for clustering, and then ideally the simplest algorithm to cluster given that property.

4.1.2 Our Results

We provide a PAC-style framework for analyzing what properties of a similarity function are sufficient to allow one to cluster well under the above two relaxations (list and tree) of the clustering objective. We analyze both algorithmic and information theoretic questions in our model and provide results for several natural game-theoretic and learning-theoretic properties. Specifically:

- We consider a family of stability-based properties, showing that a natural generalization of the "stable marriage" property is sufficient to produce a hierarchical clustering. (The property is that no two subsets $A \subset C$, $A' \subset C'$ of clusters $C \neq C'$ in the correct clustering are both more similar on average to each other than to the rest of their own clusters.) Moreover, a significantly weaker notion of stability is also sufficient to produce a hierarchical clustering, but requires a more involved algorithm.
- We show that a weaker "average-attraction" property (which is provably not enough to produce a single correct hierarchical clustering) is sufficient to produce a small list of clusterings, and give generalizations to even weaker conditions that generalize the notion of large-margin kernel functions.
- We define the *clustering complexity* of a given property (the minimum possible list length that can be guaranteed by any algorithm) and provide both upper and lower bounds for the properties we consider. This notion is analogous to notions of capacity in classification [72, 103, 203] and it provides a formal measure of the inherent usefulness of a given property.
- We also show that properties implicitly assumed by approximation algorithms for standard graph-based objective functions can be viewed as special cases of some of the properties considered above.
- We show how our methods can be extended to the *inductive* case, i.e., by using just a *constant-sized*

sample, as in property testing. While most of our algorithms extend in a natural way, for certain properties their analysis requires more involved arguments using regularity-type results of [14, 113].

More generally, our framework provides a formal way to analyze what properties of a similarity function would be sufficient to produce low-error clusterings, as well as what algorithms are suited for a given property. For some of our properties we are able to show that known algorithms succeed (e.g. variations of bottom-up hierarchical linkage based algorithms), but for the most general ones we need new algorithms that are able to take advantage of them.

4.1.3 Connections to other chapters and to other related work

Some of the questions we address can be viewed as a generalization of questions studied in Chapter 3 or in other work machine learning that asks what properties of similarity functions (especially kernel functions) are sufficient to allow one to *learn* well [24, 31, 133, 187, 190]. E.g., the usual statement is that if a kernel function satisfies the property that the target function is separable by a large margin in the implicit kernel space, then learning can be done from few labeled examples. The clustering problem is more difficult because there is no labeled data, and even in the relaxations we consider, the forms of feedback allowed are much weaker.

We note that as in learning, given an embedding of data into some metric space, the similarity function $K(x, x')$ need *not* be a direct translation of distance like $e^{-d(x,x')}$, but rather may be a derived function based on the entire dataset. For example, in the *diffusion kernel* of [156], the similarity $K(x, x')$ is related to the effective resistance between x and x' in a weighted graph defined from distances in the original metric. This would be a natural similarity function to use, for instance, if data lies in two well-separated pancakes.

In the inductive setting, where we imagine our given data is only a small random sample of the entire data set, our framework is close in spirit to recent work done on sample-based clustering (e.g., [49]) in the context of clustering algorithms designed to optimize a certain objective. Based on such a sample, these algorithms have to output a clustering of the full domain set, that is evaluated with respect to the underlying distribution.

We also note that the assumption that the similarity function satisfies a given property with respect to the target clustering is analogous to the assumption considered in Chapter 2 that the target satisfies a certain relation with respect to the underlying distribution. That is, the similarity function plays the role of the distribution in Chapter 2. At a technical level however the results are not directly comparable. In particular in Chapter 2 we focus on compatibility notions that can be estimated from a finite sample and the main angle there is understanding what is a good target for a given distribution given a compatibility relation and what is a good distribution for a given compatibility notion. Here we imagine fixing the both the target, and we are trying to understand what is a good similarity function for the given target pair.

4.2 Definitions and Preliminaries

We consider a clustering problem (S, l) specified as follows. Assume we have a data set S of n objects, where each object is an element of an abstract instance space X. Each $x \in S$ has some (unknown) "ground-truth" label $l(x)$ in $Y = \{1, \ldots, k\}$, where we will think of k as much smaller than n. The goal is to produce a hypothesis $h : X \to Y$ of low error up to isomorphism of label names. Formally, we define the error of h to be $err(h) = \min_{\sigma \in \mathcal{S}_k} [\Pr_{x \in S} [\sigma(h(x)) \neq l(x)]]$. We will assume that a target error rate ϵ, as well as k, are given as input to the algorithm.

We will be considering clustering algorithms whose only access to their data is via a pairwise similarity function $K(x, x')$ that given two examples outputs a number in the range $[-1, 1]$.[2] We will say that K is a symmetric similarity function if $K(x, x') = K(x', x)$ for all x, x'.

Our focus is to analyze natural properties that sufficient for a similarity function K to be *good* for a clustering problem (S, l) which (ideally) are intuitive, broad, and imply that such a similarity function results in the ability to *cluster well*. Formally, a property $\mathcal{P}$ is a relation $\{(l, K)\}$ and we say that K has property $\mathcal{P}$ with respect to $\mathcal{P}$ if $(l, K) \in \mathcal{P}$.

As mentioned in the introduction, however, requiring an algorithm to output a single low-error clustering rules out even quite strong properties. Instead we will consider two objectives that are natural if one assumes the ability to get some limited additional feedback from a user. Specifically, we consider the following two models:

1. **List model:** In this model, the goal of the algorithm is to propose a small number of clusterings such that at least one has error at most ϵ. As in work on property testing, the list length should depend on ϵ and k only, and be independent of n. This list would then go to a domain expert or some hypothesis-testing portion of the system which would then pick out the best clustering.
2. **Tree model:** In this model, the goal of the algorithm is to produce a hierarchical clustering: that is, a tree on subsets such that the root is the set S, and the children of any node S' in the tree form a partition of S'. The requirement is that there must exist a *pruning* h of the tree (not necessarily using nodes all at the same level) that has error at most ϵ. In many applications (e.g. document clustering) this is a significantly more user-friendly output than the list model. Note that any given tree has at most 2^{2k} prunings of size k [154], so this model is at least as strict as the list model.

Transductive vs Inductive. Clustering is typically posed as a "transductive" [203] problem in that we are asked to cluster a *given* set of points S. We can also consider an *inductive* model in which S is merely a small random subset of points from a much larger abstract instance space X, and our goal is to produce a hypothesis $h : X \to Y$ of low error on X. For a given property of our similarity function (with respect to X) we can then ask how large a set S we need to see in order for our list or tree produced with respect to S to induce a good solution with respect to X. For clarity of exposition, for most of this chapter we will focus on the transductive setting. In Section 4.6 we show how our algorithms can be adapted to the inductive setting.

Realizable vs Agnostic. For most of the properties we consider here, our assumptions are analogous to the *realizable* case in supervised learning and our goal is to get ϵ-close to the target (in a tree of list) for any desired $\epsilon > 0$. For other properties, our assumptions are more like the *agnostic* in that we will assume only that $1 - \nu$ fraction of the data satisfies a certain condition. In these cases our goal os to get $\nu + \epsilon$-close to the target.

Notation. We will denote the underlying ground-truth clusters as $C_1, \ldots, C_k$ (some of which may be empty). For $x \in X$, we use $C(x)$ to denote the cluster $C_{l(x)}$ to which point x belongs. For $A \subseteq X, B \subseteq X$, let $K(A, B) = \mathbf{E}_{x \in A, x' \in B}[K(x, x')]$. We call this the *average attraction* of A to B. Let $K_{max}(A, B) = \max_{x \in A, x' \in B} K(x, x')$; we call this *maximum attraction* of A to B. Given two clusterings g and h we define the distance $d(g, h) = \min_{\sigma \in \mathcal{S}_k} [\Pr_{x \in S} [\sigma(h(x)) \neq g(x)]]$, i.e., the fraction of points in the symmetric difference under the optimal renumbering of the clusters.

We are interested in natural *properties* that we might ask a similarity function to satisfy with respect to the ground truth clustering. For example, one (strong) property would be that all points x are more similar to all points $x' \in C(x)$ than to any $x' \notin C(x)$ – we call this the *strict separation* property. A

[2]That is, the input to the clustering algorithm is just a weighted graph. However, we still want to conceptually view K as a *function* over abstract objects.

weaker property would be to just require that points x are *on average* more similar to their own cluster than to any other cluster, that is, $K(x, C(x) - \{x\}) > K(x, C_i)$ for all $C_i \neq C(x)$. We will also consider intermediate "stability" conditions. For properties such as these we will be interested in the size of the smallest list any algorithm could hope to output that would guarantee that at least one clustering in the list has error at most ϵ. Specifically, we define the *clustering complexity* of a property as:

Definition 4.2.1 *Given a property $\mathcal{P}$ and similarity function K, define the (ϵ, k)-***clustering complexity** *of the pair $(\mathcal{P}, K)$ to be the length of the shortest list of clusterings $h_1, \ldots, h_t$ such that any consistent k-clustering is ϵ-close to some clustering in the list.*[3] *That is, at least one h_i must have error at most ϵ. The (ϵ, k)-***clustering complexity of** $\mathcal{P}$ *is the maximum of this quantity over all similarity functions K.*

The clustering complexity notion is analogous to notions of capacity in classification [72, 103, 203] and it provides a formal measure of the inherent usefulness of a given property.

Computational Complexity. In the transductive case, our goal will be to produce a list or a tree in time polynomial in n and ideally polynomial in ϵ and k as well. We will indicate when our running times involve a non-polynomial dependence on these parameters. In the inductive case, we want the running time to depend only on k and ϵ and to be independent of the size of the overall instance space X, under the assumption that we have an oracle that in constant time can sample a random point from X.

In the following sections we analyze both the clustering complexity and the computational complexity of several natural properties and provide efficient algorithms to take advantage of such functions. We start by analyzing the strict separation property as well as a natural relaxation in Section 4.3. We also give formal relationships between these properties and those considered implicitly by approximation algorithms for standard clustering objectives. We then analyze a much weaker average-attraction property in Section 4.4 that is similar to Definition 3.3.1 in Chapter 3 (and which, as we have seen, has close connections to large margin properties studied in Learning Theory [24, 31, 133, 187, 190].) This property is not sufficient to produce a hierarchical clustering, however, so we then turn to the question of how weak a property can be and still be sufficient for hierarchical clustering, which leads us to analyze properties motivated by game-theoretic notions of stability in Section 4.5.

Our framework allows one to study computational hardness results as well. While our focus is on getting positive algorithmic results, we discuss a simple few hardness examples in Section 4.8.1.

4.3 Simple Properties

We begin with the simple strict separation property mentioned above.

Property 1 *The similarity function K satisfies the* **strict separation** *property for the clustering problem (S, l) if all x are strictly more similar to any point $x' \in C(x)$ than to every $x' \notin C(x)$.*

Given a similarity function satisfying the strict separation property, we can efficiently construct a tree such that the ground-truth clustering is a pruning of this tree (Theorem 4.3.2). As mentioned above, a consequence of this fact is a $2^{O(k)}$ upper bound on the clustering complexity of this property. We begin by showing a matching $2^{\Omega(k)}$ lower bound.

Theorem 4.3.1 *For $\epsilon < \frac{1}{2k}$, the strict separation property has (ϵ, k)-clustering complexity at least $2^{k/2}$.*

Proof: The similarity function is a generalization of the similarity in the picture in Figure 4.1. Specifically, partition the n points into k subsets $\{R_1, \ldots, R_k\}$ of n/k points each. Group the subsets into pairs $\{(R_1, R_2), (R_3, R_4), \ldots\}$, and let $K(x, x') = 1$ if x and x' belong to the same R_i, $K(x, x') = 1/2$ if x and x' belong to two subsets in the same pair, and $K(x, x') = 0$ otherwise. Notice that in this setting

[3] A clustering $\mathcal{C}$ is consistent if K has property $\mathcal{P}$ with respect to $\mathcal{C}$.

there are $2^{\frac{k}{2}}$ clusterings (corresponding to whether or not to split each pair $R_i \cup R_{i+1}$) that are consistent with Property 1 and differ from each other on at least n/k points. Since $\epsilon < \frac{1}{2k}$, any given hypothesis clustering can be ϵ-close to at most one of these and so the clustering complexity is at least $2^{k/2}$. ■

We now present the upper bound.

Theorem 4.3.2 *Let K be a similarity function satisfying the strict separation property. Then we can efficiently construct a tree such that the ground-truth clustering is a pruning of this tree.*

Proof: If K is symmetric, then to produce a tree we can simply use bottom up "single linkage" (i.e., Kruskal's algorithm). That is, we begin with n clusters of size 1 and at each step we merge the two clusters C, C' maximizing $K_{max}(C, C')$. This maintains the invariant that at each step the current clustering is laminar with respect to the ground-truth: if the algorithm merges two clusters C and C', and C is strictly contained in some cluster C_r of the ground truth, then by the strict separation property we must have $C' \subset C_r$ as well. If K is not symmetric, then single linkage may fail.[4] However, in this case, the following "Boruvka-inspired" algorithm can be used. Starting with n clusters of size 1, draw a directed edge from each cluster C to the cluster C' maximizing $K_{max}(C, C')$. Then pick some cycle produced (there must be at least one cycle) and collapse it into a single cluster, and repeat. Note that if a cluster C in the cycle is strictly contained in some ground-truth cluster C_r, then by the strict separation property its out-neighbor must be as well, and so on around the cycle. So this collapsing maintains laminarity as desired. ■

Note: Even though the strict separation property is quite strong, a similarity function satisfying this property can still fool a top-down spectral clustering approach. See Figure 4.2 in Section 4.8.4.

We can also consider the agnostic version of the strict separation property, where we require that K satisfies strict separation for *most* of the data.

Property 2 *The similarity function K satisfies ν-***strict separation** *for the clustering problem (S, l) if for some $S' \subseteq S$ of size $(1-\nu)n$, K satisfies strict separation for (S', l).*

We can then show that:

Theorem 4.3.3 *If K satisfies ν-***strict separation***, then so long as the smallest correct cluster has size greater than $5\nu n$, we can produce a tree such that the ground-truth clustering is ν-close to a pruning of this tree.*

For a proof see Section 4.7, where we also show that properties implicitly assumed by approximation algorithms for standard graph-based objective functions can be viewed as special cases of the ν-strict separation property.

4.4 Weaker properties

A much weaker property to ask of a similarity function is just that most points are noticeably more similar *on average* to points in their own cluster than to points in any other cluster. This is similar to Definition 3.3.1 in Chapter 3 (and which, as we have seen, has close connections to large margin properties studied in Learning Theory [24, 31, 133, 187, 190].)

Specifically, we define:

[4]Consider 3 points x, y, z whose correct clustering is $(\{x\}, \{y, z\})$. If $K(x, y) = 1$, $K(y, z) = K(z, y) = 1/2$, and $K(y, x) = K(z, x) = 0$, then this is consistent with strict separation and yet the algorithm will incorrectly merge x and y in its first step.

Property 3 *A similarity function K satisfies the (ν, γ)-***average attraction** *property for the clustering problem (S, l) if a $1 - \nu$ fraction of examples x satisfy:*

$$K(x, C(x)) \geq K(x, C_i) + \gamma \text{ for all } i \in Y, i \neq l(x).$$

This is a fairly natural property to ask of a similarity function: if a point x is more similar on average to points in a different cluster than to those in its own, it is hard to expect an algorithm to label it correctly. The following is a simple clustering algorithm that given a similarity function K satisfying the average attraction property produces a list of clusterings of size that depends only on ϵ, k, and γ. Specifically,

Algorithm 2 Sampling Based Algorithm, List Model

Input: Data set S, similarity function K, parameters $\gamma, \epsilon > 0$, $k \in \mathbb{Z}^+$; $N(\epsilon, \gamma, k)$, $s(\epsilon, \gamma, k)$.

- Set $\mathcal{L} = \emptyset$.
- Repeat $N(\epsilon, \gamma, k)$ times

  For $k' = 1, \ldots, k$ do:
  - Pick a set $R_S^{k'}$ of $s(\epsilon, \gamma, k)$ random points from S.
  - Let h be the average-nearest neighbor hypothesis induced by the sets R_S^i, $1 \leq i \leq k'$. That is, for any point $x \in S$, define $h(x) = \text{argmax}_{i \in \{1, \ldots k'\}}[K(x, R_S^i)]$. Add h to $\mathcal{L}$.
- Output the list $\mathcal{L}$.

Theorem 4.4.1 *Let K be a similarity function satisfying the (ν, γ)-average attraction property for the clustering problem (S, l). Using Algorithm 2 with the parameters $s(\epsilon, \gamma, k) = \frac{4}{\gamma^2} \ln\left(\frac{8k}{\epsilon\delta}\right)$ and $N(\epsilon, \gamma, k) = \left(\frac{2k}{\epsilon}\right)^{\frac{4k}{\gamma^2} \ln\left(\frac{8k}{\epsilon\delta}\right)} \ln(\frac{1}{\delta})$ we can produce a list of at most $k^{O\left(\frac{k}{\gamma^2} \ln\left(\frac{1}{\epsilon}\right) \ln\left(\frac{k}{\epsilon\delta}\right)\right)}$ clusterings such that with probability $1 - \delta$ at least one of them is $(\nu + \epsilon)$-close to the ground-truth.*

Proof: We say that a ground-truth cluster is big if it has probability mass at least $\frac{\epsilon}{2k}$; otherwise, we say that the cluster is small. Let k' be the number of "big" ground-truth clusters. Clearly the probability mass in all the small clusters is at most $\epsilon/2$.

Let us arbitrarily number the big clusters $C_1, \ldots, C_{k'}$. Notice that in each round there is at least a $\left(\frac{\epsilon}{2k}\right)^{s(\epsilon,\gamma,k)}$ probability that ${R_S}^i \subseteq C_i$, and so at least a $\left(\frac{\epsilon}{2k}\right)^{ks(\epsilon,\gamma,k)}$ probability that ${R_S}^i \subseteq C_i$ for all $i \leq k'$. Thus the number of rounds $\left(\frac{2k}{\epsilon}\right)^{\frac{4k}{\gamma^2} \ln\left(\frac{8k}{\epsilon\delta}\right)} \ln(\frac{1}{\delta})$ is large enough so that with probability at least $1 - \delta/2$, in at least one of the $N(\epsilon, \gamma, k)$ rounds we have ${R_S}^i \subseteq C_i$ for all $i \leq k'$. Let us fix now one such good round. We argue next that the clustering induced by the sets picked in this round has error at most $\nu + \epsilon$ with probability at least $1 - \delta$.

Let Good be the set of x in the big clusters satisfying

$$K(x, C(x)) \geq K(x, C_j) + \gamma \text{ for all } j \in Y, j \neq l(x).$$

By assumption and from the previous observations, $\Pr_{x \sim S}[x \in \textsf{Good}] \geq 1 - \nu - \epsilon/2$. Now, fix $x \in$ Good. Since $K(x, x') \in [-1, 1]$, by Hoeffding bounds we have that over the random draw of ${R_S}^j$, conditioned on ${R_S}^j \subseteq C_j$,

$$\Pr_{{R_S}^j}\left(\left|\mathbf{E}_{x' \sim {R_S}^j}[K(x, x')] - K(x, C_j)\right| \geq \gamma/2\right) \leq 2e^{-2|{R_S}^j|\gamma^2/4},$$

for all $j \in \{1, \ldots, k'\}$. By our choice of ${R_S}^j$, each of these probabilities is at most $\epsilon\delta/4k$. So, for any given $x \in$ Good, there is at most a $\epsilon\delta/4$ probability of error over the draw of the sets ${R_S}^j$. Since this is

true for any $x \in$ Good, it implies that the *expected* error of this procedure, over $x \in$ Good, is at most $\epsilon\delta/4$, which by Markov's inequality implies that there is at most a $\delta/2$ probability that the error rate over Good is more than $\epsilon/2$. Adding in the $\nu + \epsilon/2$ probability mass of points not in Good yields the theorem. ■

Note that Theorem 4.4.1 immediately implies a corresponding upper bound on the (ϵ, k)-clustering complexity of the $(\epsilon/2, \gamma)$-average attraction property. Note that this bound however is not polynomial in k and γ. We can also give a lower bound showing that the exponential dependence on γ is necessary, and furthermore this property is not sufficient to cluster in the tree model:

Theorem 4.4.2 *For $\epsilon < \gamma/2$, the (ϵ, k)-clustering complexity of the $(0, \gamma)$-average attraction property is at least $\max_{k' \leq k} k'^{\frac{1}{\gamma}}/k'!$, and moreover this property is not sufficient to cluster in the tree model.*

Proof: Consider $\frac{1}{\gamma}$ regions $\{R_1, \ldots, R_{1/\gamma}\}$ each with γn points. Assume $K(x, x') = 1$ if x and x' belong to the same region R_i and $K(x, x') = 0$, otherwise. Notice that in this setting all the k-way partitions of the set $\{R_1, \ldots, R_{1/\gamma}\}$ are consistent with Property 3 and they are all pairwise at distance at least γn from each other. Since $\epsilon < \gamma/2$, any given hypothesis clustering can be ϵ-close to at most one of these and so the clustering complexity is at least the sum of Stirling numbers of the 2nd kind $\sum_{k'=1}^{k} S(1/\gamma, k')$ which is at least $\max_{k' \leq k} k'^{1/\gamma}/k'!$. ■

Note: In fact, the clustering complexity bound immediately implies one cannot cluster in the tree model since for $k = 2$ the bound is greater than 1.

We can further extend the lower bound in Theorem 4.4.3 to show the following:

Theorem 4.4.3 *For $\epsilon < 1/2$, the (ϵ, k)-clustering complexity of the $(0, \gamma)$-average attraction property is at least $k^{\frac{k}{8\gamma}}$.*

One can even weaken the above property to ask only that there *exists* an (unknown) weighting function over data points (thought of as a "reasonableness score"), such that most points are on average more similar to the *reasonable* points of their own cluster than to the *reasonable* points of any other cluster. This is a generalization of the notion of K being a kernel function with the large margin property [24, 191, 195, 203] as shown in Chapter 3.

Property 4 *A similarity function K satisfies the (ν, γ)-**average weighted attraction** property for the clustering problem (S, l) if there exists a weight function $w : X \to [0, 1]$ such that a $1 - \nu$ fraction of examples x satisfy:*

$$\mathbf{E}_{x' \in C(x)}[w(x')K(x, x')] \geq K_{x' \in C_r}[w(x')K(x, x')] + \gamma \text{ for all } r \in Y, r \neq l(x).$$

If we have K a similarity function satisfying the (ν, γ)-average weighted attraction property for the clustering problem (S, l), then we can again cluster well in the list model, but via a more involved clustering algorithm. Formally we can show that:

Theorem 4.4.4 *If K is a similarity function satisfying the (ν, γ)-average weighted attraction property for the clustering problem (S, l), we can produce a list of at most $k^{\tilde{O}\left(\frac{k}{\epsilon\gamma^2}\right)}$ clusterings such that with probability $1 - \delta$ at least one of them is $\epsilon + \nu$-close to the ground-truth.*

We defer the proof of Theorem 4.4.4 to Section 4.10.

A too-weak property: One could imagine further relaxing the average attraction property to simply require that for all C_i, C_j in the ground truth we have $K(C_i, C_i) \geq K(C_i, C_j) + \gamma$; that is, the average

intra-cluster similarity is larger than the average inter-cluster similarity. However, even for $k = 2$ and $\gamma = 1/4$, this is *not sufficient* to produce clustering complexity independent of (or even polynomial in) n. In particular, suppose there are two regions A, B of $n/2$ points each such that $K(x, x') = 1$ for x, x' in the same region and $K(x, x') = 0$ for x, x' in different regions. However, suppose C_1 contains 75% of A and 25% of B and C_2 contains 25% of C_1 and 75% of C_2. Then this property is satisfied for $\gamma = 1/4$ and yet by classic coding results (or Chernoff bounds), clustering complexity is clearly exponential in n for $\epsilon < 1/8$. Moreover, this implies there is no hope in the inductive (or property testing) setting.

4.5 Stability-based Properties

The properties in Section 4.4 are fairly general and allow construction of a list whose length depends only on on ϵ and k (for constant γ), but are not sufficient to produce a single tree. In this section, we show that several natural stability-based properties that lie between those considered in Sections 4.3 and 4.4 are in fact sufficient for *hierarchical* clustering.

For simplicity, we focus on symmetric similarity functions. We consider the following relaxations of Property 1 which ask that the ground truth be "stable" in the stable-marriage sense:

Property 5 *A similarity function K satisfies the* **strong stability** *property for the clustering problem (S, l) if for all clusters C_r, $C_{r'}$, $r \neq r'$ in the ground-truth, for all $A \subset C_r$, $A' \subseteq C_{r'}$ we have*

$$K(A, C_r \setminus A) > K(A, A').$$

Property 6 *A similarity function K satisfies the* **weak stability** *property for the clustering problem (S, l) if for all C_r, $C_{r'}$, $r \neq r'$, for all $A \subset C_r$, $A' \subseteq C_{r'}$, we have:*

- *If $A' \subset C_{r'}$ then either $K(A, C_r \setminus A) > K(A, A')$ or $K(A', C_{r'} \setminus A') > K(A', A)$.*
- *If $A' = C_{r'}$ then $K(A, C_r \setminus A) > K(A, A')$.*

We can interpret weak stability as saying that for any two clusters in the ground truth, there does not exist a subset A of one and subset A' of the other that are more attracted to each other than to the remainder of their true clusters (with technical conditions at the boundary cases) much as in the classic notion of stable-marriage. Strong stability asks that *both* be more attracted to their true clusters. To further motivate these properties, note that if we take the example from Figure 4.1 and set a small random fraction of the edges inside each dark-shaded region to 0, then with high probability this would still satisfy strong stability with respect to all the natural clusters even though it no longer satisfies strict separation (or even ν-strict separation for any $\nu < 1$ if we included at least one edge incident to each vertex). Nonetheless, we can show that these stability notions are sufficient to produce a hierarchical clustering. We start by proving this for strong stability here and then in Theorem 4.5.2 we also prove it for the weak stability.

Algorithm 3 Average Linkage, Tree Model

Input: Data set S, similarity function K. Output: A tree on subsets.

- Begin with n singleton clusters.
- Repeat till only one cluster remains: Find clusters C, C' in the current list which maximize $K(C, C')$ and merge them into a single cluster.
- Output the tree with single elements as leaves and internal nodes corresponding to all the merges performed.

Theorem 4.5.1 *Let K be a symmetric similarity function satisfying Property 5. Then we can efficiently construct a binary tree such that the ground-truth clustering is a pruning of this tree.*

Proof: We will show that Algorithm 3 (Average Linkage) will produce the desired result. Note that the algorithm uses $K(C, C')$ rather than $K_{max}(C, C')$ as in single linkage; in fact in Figure 4.3 (In section 4.8.4) we show an example satisfying this property where single linkage would fail.

We prove correctness by induction. In particular, assume that our current clustering is laminar with respect to the ground truth clustering (which is true at the start). That is, for each cluster C in our current clustering and each C_r in the ground truth, we have either $C \subseteq C_r$, or $\mathcal{C}_r \subseteq C$ or $C \cap C_r = \emptyset$. Now, consider a merge of two clusters C and C'. The only way that laminarity could fail to be satisfied after the merge is if one of the two clusters, say, C', is strictly contained inside some ground-truth cluster C_r (so, $C_r - C' \neq \emptyset$) and yet C is disjoint from C_r. Now, note that by Property 5, $K(C', C_r - C') > K(C', x)$ for all $x \notin C_r$, and so in particular we have $K(C', C_r - C') > K(C', C)$. Furthermore, $K(C', C_r - C')$ is a weighted average of the $K(C', C'')$ over the sets $C'' \subseteq C_r - C'$ in our current clustering and so at least one such C'' must satisfy $K(C', C'') > K(C', C)$. However, this contradicts the specification of the algorithm, since by definition it merges the pair C, C' such that $K(C', C)$ is greatest. ■

Theorem 4.5.2 *Let K be a symmetric similarity function satisfying the weak stability property. Then we can efficiently construct a binary tree such that the ground-truth clustering is a pruning of this tree.*

Proof: As in the proof of theorem 4.5.1 we show that bottom-up average-linkage will produce the desired result. Specifically, the algorithm is as follows: we begin with n clusters of size 1, and then at each step we merge the two clusters C, C' such that $K(C, C')$ is highest.

We prove correctness by induction. In particular, assume that our current clustering is laminar with respect to the ground truth clustering (which is true at the start). That is, for each cluster C in our current clustering and each C_r in the ground truth, we have either $C \subseteq C_r$, or $\mathcal{C}_r \subseteq C$ or $C \cap C_r = \emptyset$. Now, consider a merge of two clusters C and C'. The only way that laminarity could fail to be satisfied after the merge is if one of the two clusters, say, C', is strictly contained inside some ground-truth cluster $C_{r'}$ and yet C is disjoint from $C_{r'}$.

We distinguish a few cases. First, assume that C is a cluster C_r of the ground-truth. Then by definition, $K(C', C_{r'} - C') > K(C', C)$. Furthermore, $K(C', C_{r'} - C')$ is a weighted average of the $K(C', C'')$ over the sets $C'' \subseteq C_{r'} - C'$ in our current clustering and so at least one such C'' must satisfy $K(C', C'') > K(C', C)$. However, this contradicts the specification of the algorithm, since by definition it merges the pair C, C' such that $K(C', C)$ is greatest.

Second, assume that C is strictly contained in one of the ground-truth clusters C_r. Then, by the weak stability property, either $K(C, C_r - C) > K(C, C')$ or $K(C', C_{r'} - C') > K(C, C')$. This again contradicts the specification of the algorithm as in the previous case.

Finally assume that C is a union of clusters in the ground-truth $C_1, \ldots C_{k'}$. Then by definition, $K(C', C_{r'} - C') > K(C', C_i)$, for $i = 1, \ldots k'$, and so $K(C', C_{r'} - C') > K(C', C)$. This again leads to a contradiction as argued above. ■

While natural, Properties 5 and 6 are still somewhat brittle: in the example of Figure 4.1, for instance, if one adds a small number of edges with similarity 1 *between* the natural clusters, then the properties are no longer satisfied for them (because pairs of elements connected by these edges will want to defect). We can make the properties more robust by requiring that stability hold only for *large* sets. This will break the average-linkage algorithm used above, but we can show that a more involved algorithm building on the approach used in Section 4.4 will nonetheless find an approximately correct tree. For simplicity, we focus on broadening the strong stability property, as follows (one should view s as small compared to ϵ/k in this definition):

Property 7 *The similarity function K satisfies the (s, γ)-***strong stability of large subsets** *property for the clustering problem (S, l) if for all clusters C_r, $C_{r'}$, $r \neq r'$ in the ground-truth, for all $A \subset C_r$,*

$A' \subseteq C_{r'}$ *with* $|A| + |A'| \geq sn$ *we have*

$$K(A, C_r \setminus A) > K(A, A') + \gamma.$$

The idea of how we can use this property is we will first run an algorithm for the list model much like Algorithm 2, viewing its output as simply a long list of candidate clusters (rather than cluster*ings*). In particular, we will get a list $\mathcal{L}$ of $k^{O\left(\frac{k}{\gamma^2}\log\frac{1}{\epsilon}\log\frac{k}{\delta f}\right)}$ clusters such that with probability at least $1-\delta$ any cluster in the ground-truth of size at least $\frac{\epsilon}{4k}$ is close to one of the clusters in the list. We then run a second "tester" algorithm that is able to throw away candidates that are sufficiently non-laminar with respect to the correct clustering and assembles the ones that remain into a tree. We present and analyze the tester algorithm, Algorithm 4, below.

Algorithm 4 Testing Based Algorithm, Tree Model.

Input: Data set S, similarity function K, parameters $\gamma > 0$, $k \in \mathbb{Z}^+$, $f, g, s, \alpha > 0$. A list of clusters $\mathcal{L}$ with the property that any cluster C in the ground-truth is at least f-close to one of them.
Output: A tree on subsets.

1. Throw out all clusters of size at most αn. For every pair of clusters C, C' in our list $\mathcal{L}$ of clusters that are sufficiently "non-laminar" with respect to each other in that $|C \setminus C'| \geq gn$, $|C' \setminus C| \geq gn$ and $|C \cap C'| \geq gn$, compute $K(C \cap C', C \setminus C')$ and $K(C \cap C', C' \setminus C)$. Throw out whichever one does worse: i.e., throw out C if the first similarity is smaller, else throw out C'. Let $\mathcal{L}'$ be the remaining list of clusters at the end of the process.
2. Greedily sparsify the list $\mathcal{L}'$ so that no two clusters are approximately equal (that is, choose a cluster, throw out all that are approximately equal to it, and repeat). We say two clusters C, C' are approximately equal if $|C \setminus C'| \leq gn$, $|C' \setminus C| \leq gn$ and $|C' \cap C| \geq gn$. Let $\mathcal{L}''$ be the list remaining.
3. Construct a forest on the remaining list $\mathcal{L}''$. C becomes a child of C' in this forest if C' approximately contains C, i.e. $|C \setminus C'| \leq gn$, $|C' \setminus C| \geq gn$ and $|C' \cap C| \geq gn$.
4. Complete the forest arbitrarily into a tree.

Theorem 4.5.3 *Let K be a similarity function satisfying (s, γ)-strong stability of large subsets for the clustering problem (S, l). Let $\mathcal{L}$ be a list of clusters such that any cluster in the ground-truth of size at least αn is f-close to one of the clusters in the list. Then Algorithm 4 with parameters satisfying $s + f \leq g$, $f \leq g\gamma/10$ and $\alpha > 6kg$ yields a tree such that the ground-truth clustering is $2\alpha k$-close to a pruning of this tree.*

Proof: Let k' be the number of "big" ground-truth clusters: the clusters of size at least αn; without loss of generality assume that C_1, ..., $C_{k'}$ are the big clusters.

Let C'_1, ...,$C'_{k'}$ be clusters in $\mathcal{L}$ such that $d(C_i, C'_i)$ is at most f for all i. By Property 7 and Lemma 4.5.4 (stated below), we know that after Step 1 (the "testing of clusters" step) all the clusters C'_1, ...,$C'_{k'}$ survive; furthermore, we have three types of relations between the remaining clusters. Specifically, either:

(a) C and C' are approximately equal; that means $|C \setminus C'| \leq gn$, $|C' \setminus C| \leq gn$ and $|C' \cap C| \geq gn$.

(b) C and C' are approximately disjoint; that means $|C \setminus C'| \geq gn$, $|C' \setminus C| \geq gn$ and $|C' \cap C| \leq gn$.

(c) or C' approximately contains C; that means $|C \setminus C'| \leq gn$, $|C' \setminus C| \geq gn$ and $|C' \cap C| \geq gn$.

Let $\mathcal{L}''$ be the remaining list of clusters after sparsification. It's easy to show that there exists $C''_1, ..., C''_{k'}$ in $\mathcal{L}''$ such that $d(C_i, C''_i)$ is at most $(f + 2g)$, for all i. Moreover, all the elements in $\mathcal{L}''$ are either in the relation "subset" or "disjoint". Also, since all the clusters C_1, ..., $C_{k'}$ have size at least αn, we also have

that C_i'', C_j'' are in the relation "disjoint", for all i, j, $i \neq j$. That is, in the forest we construct C_i'' are not descendants of one another.

We show $C_1'', ..., C_{k'}''$ are part of a pruning of small error rate of the final tree. We do so by exhibiting a small extension to a list of clusters $\mathcal{L}'''$ that are all approximately disjoint and nothing else in $\mathcal{L}''$ is approximately disjoint from any of the clusters in $\mathcal{L}'''$ (thus $\mathcal{L}'''$ will be the desired pruning). Specifically greedily pick a cluster $\tilde{C}_1$ in $\mathcal{L}''$ that is approximately disjoint from $C_1'', ..., C_{k'}''$, and in general in step $i > 1$ greedily pick a cluster $\tilde{C}_1$ in $\mathcal{L}''$ that is approximately disjoint from $C_1'', ..., C_{k'}'', \tilde{C}_1, \ldots, \tilde{C}_{i-1}$. Let $C_1'', ..., C_{k'}'', \tilde{C}_1, \ldots, \tilde{C}_{\tilde{k}}$ be the list $\mathcal{L}'''$. By design, $\mathcal{L}'''$ will be a pruning of the final tree and we now claim its total error is at most $2\alpha kn$. In particular, note that the total number of points missing from $C_1'', ..., C_{k'}''$ is at most $k(f+2g)n + k\alpha n \leq \frac{3}{2}k\alpha n$. Also, by construction, each $\tilde{C}_i$ must contain at least $\alpha n - (k+i)gn$ new points, which together with the above implies that $\tilde{k} \leq 2k$. Thus, the total error of $\mathcal{L}'''$ overall is at most $\frac{3}{2}\alpha kn + 2kk'gn \leq 2\alpha kn$. ■

Lemma 4.5.4 *Let K be a similarity function satisfying the (s, γ)-strong stability of large subsets property for the clustering problem (S, l). Let C, C' be such that $|C \cap C'| \geq gn$, $|C \setminus C'| \geq gn$ and $|C' \setminus C| \geq gn$. Let C^* be a cluster in the underlying ground-truth such that $|C^* \setminus C| \leq fn$ and $|C \setminus C^*| \leq fn$. Let $I = C \cap C'$. If $s + f \leq g$ and $f \leq g\gamma/10$, then $K(I, C \setminus I) > K(I, C' \setminus I)$.*

Proof: Let $I^* = I \cap C^*$. So, $I^* = C \cap C' \cap C^*$. We prove first that

$$K(I, C \setminus I) > K(I^*, C^* \setminus I^*) - \gamma/2. \tag{4.1}$$

Since $K(x, x') \geq -1$, we have

$$K(I, C \setminus I) \geq (1 - p_1)K(I \cap C^*, (C \setminus I) \cap C^*) - p_1,$$

where $1 - p_1 = \frac{|I^*|}{|I|} \cdot \frac{|(C \setminus I) \cap C^*|}{|C \setminus I|}$. By assumption we have $|I| \geq gn$, and also $|I \setminus I^*| \leq fn$. That means $\frac{|I^*|}{|I|} = \frac{|I| - |I \setminus I^*|}{|I|} \geq \frac{g-f}{g}$. Similarly, $|C \setminus I| \geq gn$ and $\left|(C \setminus I) \cap \bar{C}^*\right| \leq |C \setminus C^*| \leq fn$. So,

$$\frac{|(C \setminus I) \cap C^*|}{|C \setminus I|} = \frac{|C \setminus I| - \left|(C \setminus I) \cap \bar{C}^*\right|}{|C \setminus I|} \geq \frac{g-f}{g}.$$

Let us denote by $1 - p$ the quantity $\left(\frac{g-f}{g}\right)^2$. We have:

$$K(I, C \setminus I) \geq (1 - p)K(I^*, (C \setminus I) \cap C^*) - p. \tag{4.2}$$

Let $A = (C^* \setminus I^*) \cap C$ and $B = (C^* \setminus I^*) \cap \bar{C}$. We have

$$K(I^*, C^* \setminus I^*) = (1 - \alpha)K(I^*, A) - \alpha K(I^*, B), \tag{4.3}$$

where $1 - \alpha = \frac{|A|}{|C^* \setminus I^*|}$. Note that

$$A = (C^* \setminus I^*) \cap C = (C^* \cap C) \setminus (I^* \cap C) = (C^* \cap C) \setminus I^*$$

and

$$(C \setminus I) \cap C^* = (C \cap C^*) \setminus (I \cap C^*) = (C^* \cap C) \setminus I^*,$$

so $A = (C \setminus I) \cap C^*$. Furthermore

$$|(C \setminus I) \cap C^*| = |(C \setminus C') \setminus (C \setminus (C' \cap C^*))| \geq |C \setminus C'| - |C \setminus (C' \cap C^*)| \geq |C \setminus C'| - |C \setminus C^*| \geq gn - fn.$$

We also have $|B| = |(C^* \setminus I^*) \cap \bar{C}| \geq |C^* \setminus C|$. These imply that $1 - \alpha = \frac{|A|}{|A|+|B|} = \frac{1}{1+|B|/|A|} \geq \frac{g-f}{g}$, and furthermore $\frac{\alpha}{1-\alpha} = -1 + \frac{1}{1-\alpha} \leq \frac{f}{g-f}$. Equation (4.3) implies

$$K(I^*, A) = \frac{1}{1-\alpha} K(I^*, C^* \setminus I^*) - \frac{\alpha_1}{1-\alpha_1} \alpha_1 K(I^*, B)$$

and since $K(x, x') \leq 1$, we obtain:

$$K(I^*, A) \geq K(I^*, C^* \setminus I^*) - \frac{f}{g-f}. \tag{4.4}$$

Overall, combining (4.2) and (4.4) we obtain: $K(I, C \setminus I) \geq (1-p)\left[K(I^*, C^* \setminus I^*) - \frac{f}{g-f}\right] - p$, so

$$K(I, C \setminus I) \geq K(I^*, C^* \setminus I^*) - 2p - (1-p)\frac{f}{g-f}.$$

We prove now that $2p + (1-p)\frac{f}{g-f} \leq \gamma/2$, which finally implies relation (4.1). Since $1 - p = \left(\frac{g-f}{g}\right)^2$, we have $p = \frac{2gf-f^2}{g^2}$, so $2p + (1-p)\frac{f}{g-f} = 2\frac{2gf-f^2}{g^2} + \frac{f(g-f)}{g^2} = 4\frac{f}{g} - 2\left(\frac{f}{g}\right)^2 + \frac{f}{g} - \left(\frac{f}{g}\right)^2 = 5\frac{f}{g} - 2\left(\frac{f}{g}\right)^2 \leq \gamma/2$, since by assumption $f \leq g\gamma/10$.

Our assumption that K is a similarity function satisfying the strong stability property with a threshold sn and a γ-gap for our clustering problem (S, l), together with the assumption $s + f \leq g$ implies

$$K(I^*, C^* \setminus I^*) \geq K(I^*, C' \setminus (I^* \cup C^*)) + \gamma. \tag{4.5}$$

We finally prove that

$$K(I^*, C' \setminus (I^* \cup C^*)) \geq K(I, C' \setminus I) - \gamma/2. \tag{4.6}$$

The proof is similar to the proof of statement (4.1). First note that

$$K(I, C' \setminus I) \leq (1 - p_2) K(I^*, (C' \setminus I) \cap \bar{C^*}) + p_2,$$

where $1 - p_2 = \frac{|I^*|}{|I|} \cdot \frac{|(C' \setminus I) \cap \bar{C^*}|}{|C' \setminus I|}$. We know from above that $\frac{|I^*|}{|I|} \geq \frac{g-f}{g}$, and we can also show $\frac{|(C' \setminus I) \cap \bar{C^*}|}{|C' \setminus I|} \geq \frac{g-f}{g}$. So $1 - p_2 \geq \left(\frac{g-f}{g}\right)^2$, and so $p_2 \leq 2\frac{g}{f} \leq \gamma/2$, as desired.

To complete the proof note that relations (4.1), (4.5) and (4.6) together imply the desired result, namely that $K(I, C \setminus I) > K(I, C' \setminus I)$. ■

Theorem 4.5.5 *Let K be a similarity function satisfying the (s, γ)-strong stability of large subsets property for the clustering problem (S, l). Assume that $s = O(\epsilon^2\gamma/k^2)$. Then using Algorithm 4 with parameters $\alpha = O(\epsilon/k)$, $g = O(\epsilon^2/k^2)$, $f = O(\epsilon^2\gamma/k^2)$, together with Algorithm 2 we can with probability $1 - \delta$ produce a tree with the property that the ground-truth is ϵ-close to a pruning of this tree. Moreover, the size of this tree is $O(k/\epsilon)$.*

Proof: First, we run Algorithm 2 get a list $\mathcal{L}$ of clusters such that with probability at least $1 - \delta$ any cluster in the ground-truth of size at least $\frac{\epsilon}{4k}$ is f-close to one of the clusters in the list. We can ensure that our list $\mathcal{L}$ has size at most $k^{O\left(\frac{k}{\gamma^2}\log\frac{1}{\epsilon}\log\frac{k}{\delta f}\right)}$. We then run Procedure 4 with parameters $\alpha = O(\epsilon/k)$, $g = O(\epsilon^2/k^2)$, $f = O(\epsilon^2\gamma/k^2)$. We thus obtain a tree with the guarantee that the ground-truth is ϵ-close to a pruning of this tree (see Theorem 4.5.3). To complete the proof we only need to show that this tree has $O(k/\epsilon)$ leaves. This follows from the fact that all leaves of our tree have at least αn points and the overlap between any two of them is at most gn (for a formal proof see lemma 4.5.6). ■

Lemma 4.5.6 *Let P_1, ..., P_s be a quasi-partition of S such that $|P_i| \geq n\frac{\nu}{k}$ and $|P_i \cap P_j| \leq gn$ for all $i, j \in \{1, \ldots, s\}$, $i \neq j$. If $g = \frac{\nu^2}{5k^2}$, then $s \leq 2\frac{k}{\nu}$.*

Proof: Assume for contradiction that $s > L = 2\frac{k}{\nu}$, and consider the first L parts P_1, ..., P_L. Then $\left(n\frac{\nu}{k} - 2\frac{k}{\nu}gn\right)2\frac{k}{\nu}$ is a lower bound on the number of points that belong to exactly one of the parts P_i, $i \in \{1, \ldots, L\}$. For our choice of g, $g = \frac{\nu^2}{5k^2}$, we have $\left(n\frac{\nu}{k} - 2\frac{k}{\nu}gn\right)2\frac{k}{\nu} = 2n - \frac{4}{5}n$. So $\frac{6}{5}n$ is a lower bound on the number of points that belong to exactly one of the parts P_i, $i \in \{1, \ldots, L\}$, which is impossible since $|S| = n$. So, we must have $s \leq 2\frac{k}{\nu}$. ■

To better illustrate our properties, we present a few interesting examples in Section 4.8.4.

4.6 Inductive Setting

In this section we consider an *inductive* model in which S is merely a small random subset of points from a much larger abstract instance space X, and clustering is represented *implicitly* through a hypothesis $h : X \to Y$. In the list model our goal is to produce a list of hypotheses, $\{h_1, \ldots, h_t\}$ such that at least one of them has error at most ϵ. In the tree model we assume that each node in the tree induces a cluster which is implicitly represented as a function $f : X \to \{0, 1\}$. For a fixed tree t and a point x, we define $t(x)$ as the subset of nodes in T that contain x (the subset of nodes $f \in t$ with $f(x) = 1$). We say that a tree T has error at most ϵ if $T(X)$ has a pruning $f_1, ..., f_{k'}$ of error at most ϵ.

We analyze in the following, for each of our properties, how large a set S we need to see in order for our list or tree produced with respect to S to induce a good solution with respect to X.

The average attraction property. Our algorithms for the average attraction property (Property 3) and the average weighted attraction property are already inherently inductive.

The strict separation property. We can adapt the algorithm in Theorem 4.3.2 to the inductive setting as follows. We first draw a set S of $n = O\left(\frac{k}{\epsilon}\ln\left(\frac{k}{\delta}\right)\right)$ unlabeled examples. We run the algorithm described in Theorem 4.3.2 on this set and obtain a tree t on the subsets of S. Let Q be the set of leaves of this tree. We associate each node u in t a boolean function f_u specified as follows. Consider $x \in X$, and let $q(x) \in Q$ be the leaf given by $\mathrm{argmax}_{q \in Q} K(x, q)$; if u appears on the path from $q(x)$ to the root, then set $f_u(x) = 1$, otherwise set $f_u(x) = 0$.

Note that n is large enough to ensure that with probability at least $1 - \delta$, S includes at least a point in each cluster of size at least $\frac{\epsilon}{k}$. Remember that $\mathcal{C} = \{C_1, \ldots, C_k\}$ is the correct clustering of the entire domain. Let $\mathcal{C}_S$ be the (induced) correct clustering on our sample S of size n. Since our property is hereditary, Theorem 4.3.2 implies that $\mathcal{C}_S$ is a pruning of t. It then follows from the specification of our algorithm and from the definition of the strict separation property that with probability at least $1 - \delta$ the partition induced over the whole space by this pruning is ϵ-close to $\mathcal{C}$.

The strong stability of large subsets property. We can also naturally extend the algorithm for Property 7 to the inductive setting. The main difference in the inductive setting is that we have to *estimate* (rather than *compute*) the $|C_r \setminus C_{r'}|$, $|C_{r'} \setminus C_r|$, $|C_r \cap C_{r'}|$, $K(C_r \cap C_{r'}, C_r \setminus C_{r'})$ and $K(C_r \cap C_{r'}, C_{r'} \setminus C_r)$ for any two clusters $C_r, C_{r'}$ in the list $\mathcal{L}$. We can easily do that with only $\mathrm{poly}(k, 1/\epsilon, 1/\gamma, 1/\delta)\log(|\mathcal{L}|))$ additional points, where $\mathcal{L}$ is the input list in Algorithm 4 (whose size depends on $1/\epsilon$, $1/\gamma$ and k only). Specifically, using a modification of the proof in Theorem 4.5.5 and standard concentration inequalities (e.g. the McDiarmid inequality [103]) we can show that:

Theorem 4.6.1 *Assume that K is a similarity function satisfying the (s, γ)-strong stability of large subsets property for (X, l). Assume that $s = O(\epsilon^2\gamma/k^2)$. Then using Algorithm 4 with parameters $\alpha = O(\epsilon/k)$, $g = O(\epsilon^2/k^2)$, $f = O(\epsilon^2\gamma/k^2)$, together with Algorithm 2 we can produce a tree with the property that*

the ground-truth is ϵ-close to a pruning of this tree. Moreover, the size of this tree is $O(k/\epsilon)$. We use $O\big(\frac{k}{\gamma^2}\ln\big(\frac{k}{\epsilon\delta}\big)\cdot\big(\frac{k}{\epsilon}\big)^{\frac{4k}{\gamma^2}\ln\big(\frac{k}{\epsilon\delta}\big)}\ln(\frac{1}{\delta})\big)$ points in the first phase and $O\big(\frac{1}{\gamma^2}\frac{1}{g^2}\frac{k}{\gamma^2}\log\frac{1}{\epsilon}\log\frac{k}{\delta f}\log k\big)$ points in the second phase.

Note that each cluster is represented as a nearest neighbor hypothesis over at most k sets.

The strong stability property. We first note that we need to consider a variant of our property that has a γ-gap. To see why this is necessary consider the following example. Suppose all $K(x,x')$ values are equal to $1/2$, except for a special single center point x_i in each cluster C_i with $K(x_i,x)=1$ for all x in C_i. This satisfies strong-stability since for every $A\subset C_i$ we have $K(A,C_i\setminus A)$ is strictly larger than $1/2$. Yet it is impossible to cluster in the inductive model because our sample is unlikely to contain the center points. The variant of our property that is suited to the inductive setting is the following:

Property 8 *The similarity function K satisfies the* **γ-strong stability** *property for the clustering problem (X,l) if for all clusters C_r, $C_{r'}$, $r\neq r'$ in the ground-truth, for all $A\subset C_r$, for all $A'\subseteq C_{r'}$ we have*

$$K(A,C_r\setminus A)>K(A,A')+\gamma.$$

For this property, we could always run the algorithm for Theorem 4.6.1, though running time would be exponential in k and $1/\gamma$. We show here how we can get polynomial dependence on these parameters by adapting Algorithm 3 to the inductive setting as in the case of the strict order property. Specifically, we first draw a set S of n unlabeled examples. We run the average linkage algorithm on this set and obtain a tree t on the subsets of S. We then attach each new point x to its most similar leaf in this tree as well as to the set of nodes on the path from that leaf to the root. For a formal description see Algorithm 5. While this algorithm looks natural, proving its correctness requires more involved arguments.

Algorithm 5 Inductive Average Linkage, Tree Model

Input: Similarity function K, parameters $\gamma,\epsilon>0$, $k\in\mathbb{Z}^+$; $n=n(\epsilon,\gamma,k,\delta)$;

- Pick a set $S=\{x_1,\ldots,x_n\}$ of n random examples from X
- Run the average linkage algorithm (Algorithm 3) on the set S and obtain a tree t on the subsets of S. Let Q be the set of leaves of this tree.
- Associate each node u in t a function f_u (which induces a cluster) specified as follows.

  Consider $x\in X$, and let $q(x)\in Q$ be the leaf given by $\mathrm{argmax}_{q\in Q}K(x,q)$; if u appears on the path from $q(x)$ to the root, then set $f_u(x)=1$, otherwise set $f_u(x)=0$.
- Output the tree t.

We show in the following that for $n=\mathrm{poly}(k,1/\epsilon,1/\gamma,1/\delta)$ we obtain a tree T which has a pruning $f_1,...,f_{k'}$ of error at most ϵ. Specifically:

Theorem 4.6.2 *Let K be a similarity function satisfying the strong stability property for the clustering problem (X,l). Then using Algorithm 5 with parameters $n=\mathrm{poly}(k,1/\epsilon,1/\gamma,1/\delta)$, we can produce a tree with the property that the ground-truth is ϵ-close to a pruning of this tree.*

Proof: Remember that $\mathcal{C}=\{C_1,\ldots,C_k\}$ is the ground-truth clustering of the entire domain. Let $\mathcal{C}_S=\{C_1',\ldots,C_k'\}$ be the (induced) correct clustering on our sample S of size n. As in the previous arguments we assume that a cluster is big if it has probability mass at least $\frac{\epsilon}{2k}$.

First, Theorem 4.6.3 below implies that with high probability the clusters C_i' corresponding to the large ground-truth clusters satisfy our property with a gap $\gamma/2$. (Just perform a union bound over $x\in S\setminus C_i'$.)

It may be that C_i' corresponding to the small ground-truth clusters do not satisfy the property. However, a careful analysis of the argument in Theorem 4.5.1 shows that that with high probability $\mathcal{C}_S$ is a pruning of the tree t. Furthermore since n is large enough we also have that with high probability $K(x, C(x))$ is within $\gamma/2$ of $K(x, C'(x))$ for a $1-\epsilon$ fraction of points x. This ensures that with high probability, for any such good x the leaf $q(x)$ belongs to $C(x)$. This finally implies that the partition induced over the whole space by the pruning $\mathcal{C}_S$ of the tree t is ϵ-close to $\mathcal{C}$. ■

Note that each cluster u is implicitly represented by the function f_u defined in the description of Algorithm 5.

We prove in the following that for a sufficiently large value of n sampling preserves stability. Specifically:

Theorem 4.6.3 *Let $C_1, C_2, \ldots, C_k$ be a partition of a set X such that for any $A \subseteq C_i$ and any $x \notin C_i$,*

$$K(A, C_i \setminus A) \geq K(A, x) + \gamma.$$

Let $x \notin C_i$ and let C_i' be a random subset of n' elements of C_i. Then, $n' = poly(1/\gamma, \log(1/\delta))$ is sufficient so that with probability $1-\delta$, for any $A \subset C_i'$,

$$K(A, C_i' \setminus A) \geq K(A, x) + \frac{\gamma}{2}.$$

Proof: First of all, the claim holds for singleton subsets A with high probability using a Chernoff bound. This implies the condition is also satisfied for every subset A of size at most $\gamma n'/2$. Thus, it remains to prove the claim for large subsets. We do this using the cut-decomposition of [113] and the random sampling analysis of [14].

Let $N = |C_i|$. By [113], we can decompose the similarity matrix for C_i into a sum of cut-matrices $B_1 + B_2 + \ldots + B_s$ plus a low cut-norm matrix W with the following properties. First, each B_j is a cut-matrix, meaning that for some subset S_{j1} of the rows and subset S_{j2} of the columns and some value d_j, we have: $B_j[xy] = d_j$ for $x \in S_{j1}, y \in S_{j2}$ and all $B_j[xy] = 0$ otherwise. Second, each $d_j = O(1)$. Finally, $s = 1/\epsilon^2$ cut-matrices are sufficient so that matrix W has cut-norm at most $\epsilon^2 N$: that is, for any partition of the vertices A, A', we have $|\sum_{x \in A, y \in A'} W[xy]| \leq \epsilon N^2$; moreover, $||W||_\infty \leq 1/\varepsilon$ and $||W||_F \leq N$.

We now closely follow arguments in [14]. First, let us imagine that we have exact equality $C_i = B_1 + \ldots + B_s$, and we will add in the matrix W later. We are given that for all A, $K(A, C_i \setminus A) \geq K(A, x) + \gamma$. In particular, this trivially means that for each "profile" of sizes $\{t_{jr}\}$, there is no set A satisfying

$$\begin{aligned} |A \cap S_{jr}| &\in [t_{jr} - \alpha, t_{jr} + \alpha]N \\ |A| &\geq (\gamma/4)N \end{aligned}$$

that violates our given condition. The reason for considering cut-matrices is that the values $|A \cap S_{jr}|$ completely determine the quantity $K(A, C_i \setminus A)$. We now set α so that the above constraints determine $K(A, C_i \setminus A)$ up to $\pm\gamma/4$. In particular, choosing $\alpha = o(\gamma^2/s)$ suffices. This means that fixing a profile of values $\{t_{jr}\}$, we can replace "violates our given condition" with $K(A, x) \geq c_0$ for some value c_0 depending on the profile, losing only an amount $\gamma/4$. We now apply Theorem 9 (random sub-programs of LPs) of [14]. This theorem states that with probability $1-\delta$, in the subgraph C_i', there is no set A' satisfying the above inequalities where the right-hand-sides and objective c_0 are reduced by $O(\sqrt{\log(1/\delta)}/\sqrt{n})$. Choosing $n \gg \log(1/\delta)/\alpha^2$ we get that with high probability the induced cut-matrices B_i' have the property that there is no A' satisfying

$$\begin{aligned} |A' \cap S_{jr}'| &\in [t_{jr} - \alpha/2, t_{jr} + \alpha/2]N \\ |A'| &\geq (\gamma/2)n' \end{aligned}$$

with the objective value c_0 reduced by at most $\gamma/4$. We now simply do a union-bound over all possible profiles $\{t_{jr}\}$ consisting of multiples of α to complete the argument.

Finally, we incorporate the additional matrix W using the following result from [14].

Lemma 4.6.4 *[14][Random submatrix] For $\varepsilon, \delta > 0$, and any W an $N \times N$ real matrix with cut-norm $||W||_C \leq \varepsilon N^2$, $||W||_\infty \leq 1/\varepsilon$ and $||W||_F \leq N$, let S' be a random subset of the rows of W with $n' = |S'|$ and let W' be the $n' \times n'$ submatrix of W corresponding to W. For $n' > (c_1/\varepsilon^4\delta^5)\log(2/\varepsilon)$, with probability at least $1 - \delta$,*

$$||W'||_C \leq c_2 \frac{\varepsilon}{\sqrt{\delta}} n'^2$$

where c_1, c_2 are absolute constants.

We want the addition of W' to influence the values $K(A, C_i' - A)$ by $o(\gamma)$. We now use the fact that we only care about the case that $|A| \geq \gamma n'/2$ and $|C_i' - A| \geq \gamma n'/2$, so that it suffices to affect the sum $\sum_{x \in A, y \in C_i' - A} K(x, y)$ by $o(\gamma^2 n'^2)$. In particular, this means it suffices to have $\epsilon = \tilde{o}(\gamma^2)$, or equivalently $s = \tilde{O}(1/\gamma^4)$. This in turn implies that it suffices to have $\alpha = \tilde{o}(\gamma^6)$, which implies that $n' = \tilde{O}(1/\gamma^{12})$ suffices for the theorem. ■

4.7 Approximation Assumptions

When developing a c-approximation algorithm for some clustering objective function F, if the goal is to actually get the points correct, then one is implicitly making the assumption (or hope) that any c-approximation to F must be ϵ-close in symmetric difference to the target clustering. We show here we show how assumptions of this kind can be viewed as special cases of the ν-strict separation property.

Property 9 *Given objective function F, we say that a metric d over point set S satisfies the (c, ϵ)-F property with respect to target $\mathcal{C}$ if all clusterings $\mathcal{C}'$ that are within a factor c of optimal in terms of objective F are ϵ-close to $\mathcal{C}$.*

We now consider in particular the k-median and k-center objective functions.

Theorem 4.7.1 *If metric d satisfies the $(2, \epsilon)$-k-median property for dataset S, then the similarity function $-d$ satisfies the ν-strict separation property for $\nu = 4\epsilon$.*

Proof: Let $C = C_1, C_2, ..., C_k$ be the target clustering and let $\mathrm{OPT} = \{\mathrm{OPT}_1, \mathrm{OPT}_2 ..., \mathrm{OPT}_k\}$ be the k-median optimal clustering, where $\sum_i |C_i \cap OPT_i| \geq (1 - \epsilon)n$. Let's mark the all set of points of size at most ϵn at most where C and OPT disagree.

If there exists an unmarked x_j that is more similar to some unmarked z_j in a different cluster than to some unmarked y_j in its own cluster, and if so we mark all three points. If this process halts after $\leq \epsilon n$ rounds, then we are happy: the unmarked set, which has at least $(1-4\epsilon)n$ points, satisfies strict separation. We now claim we can get a contradiction if the process lasts longer. Specifically, begin with OPT (not C) and move each x_j to the cluster containing point z_j. Call the result OPT'. Note that for all j, the pair (x_j, y_j) are in the *same* cluster in C (because we only chose from unmarked points where $\mathcal{C}$ and OPT agree) but are in *different* clusters in OPT'. So, $d(\mathrm{OPT}', C) > \epsilon n$. However, OPT' has cost at most $2\,\mathrm{OPT}$; to see this note that moving x_i into the cluster of the corresponding z_i will increase the k-median objective by at most $\mathrm{cost}'(x_j) \leq d(x_j, z_j) + \mathrm{cost}(z_j) \leq d(x_j, y_j) + \mathrm{cost}(z_j) \leq \mathrm{cost}(x_j) + \mathrm{cost}(y_j) + \mathrm{cost}(z_j)$. Thus, the k-median objective at most doubles, i.e, $\mathrm{cost}'(\mathrm{OPT}') \leq \mathrm{cost}(\mathrm{OPT})$ contradicting our initial assumption. ■

We can similarly prove:

Theorem 4.7.2 *If the metric d satisfies the (3, ϵ)-k-center property, then the similarity function $(-d)$ satisfies the ν-strict separation property for $\nu = 4\epsilon$.*

So if the metric d satisfies the (2, ϵ)-k-median or the (2, ϵ)-k-center property for dataset S, then the similarity function $-d$ satisfies the ν-strict separation property for $\nu = 4\epsilon$. Theorem 4.3.3 (in Section 4.7.1) then implies that as long as the smallest cluster in the target has size $20\epsilon n$ we can produce a tree such that the ground-truth clustering is 4ϵ-close to a pruning of this tree.

Note: In fact, the both the $(2, \epsilon)$-k-median property and the $(2, \epsilon)$-k-means property are quite a bit more restrictive than ν-strict separation. They imply, for instance, that except for an $O(\epsilon)$ fraction of "bad" points, there exists d such that all points in the same cluster have distance much less than d and all points in different clusters have distance much greater than d. In contrast, ν-strict separation would allow for different distance scales at different parts of the graph.

We have further exploited this in recent work [42]. Specifically in [42] we show that if we assume that any c-approximation to the k-median objective is ϵ-close to the target—then we can produce clusterings that are $O(\epsilon)$-close to the target, *even for values c for which obtaining a c-approximation is NP-hard.*

In particular, the main results of [42] for the are the following:

Theorem 4.7.3 *If metric d satisfies the $(1+\alpha, \epsilon)$-k-median property for dataset S and each cluster in the target clustering has size at least $(4 + 15/\alpha)\epsilon n + 2$, then we can efficiently find a clustering that is ϵ-close to the target.*

Theorem 4.7.4 *If metric d satisfies the $(1+\alpha, \epsilon)$-k-median property for dataset S, then we can efficiently find a clustering which is $O(\epsilon/\alpha)$-close to the target.*

These results also highlight a somewhat surprising conceptual difference between assuming that the *optimal* solution to the k-median objective is ϵ-close to the target, and assuming that any *approximately optimal* solution is ϵ-close to the target, even for approximation factor say $c = 1.01$. In the former case, the problem of finding a solution that is $O(\epsilon)$-close to the target remains computationally hard, and yet for the latter we have an efficient algorithm.

We also prove in [42] similar results for the k-means and min-sum properties.

4.7.1 The ν-strict separation Property

We end this section by proving theorem 4.3.3.

Theorem 4.3.3 If K satisfies **ν-strict separation**, then so long as the smallest correct cluster has size greater than $5\nu n$, we can produce a tree such that the ground-truth clustering is ν-close to a pruning of this tree.

Proof: Let $S' \subseteq S$ be the set of $(1 - \nu)n$ points such that K satisfies strict separation with respect to S'. Call the points in S' "good", and those not in S' "bad" (of course, goodness is not known to the algorithm). We first generate a list $\mathcal{L}$ of n^2 clusters such that, ignoring bad points, any cluster in the ground-truth is in the list. We can do this by for each point $x \in S$ creating a cluster of the t nearest points to it for each $4\nu n \leq t \leq n$.

We next run a procedure that removes points from clusters that are non-laminar with respect to each other without hurting any of the correct clusters, until the remaining set is fully laminar. Specifically, while there exist two clusters C and C' that are non-laminar with respect to each other, we do the following:

1. If either C or C' has size $\leq 4\nu n$, delete it from the list. (By assumption, it cannot be one of the ground-truth clusters).
2. If C and C' are "somewhat disjoint" in that $|C \setminus C'| > 2\nu n$ and $|C' \setminus C| > 2\nu n$, each point $x \in C \cap C'$ chooses one of C or C' to belong to based on whichever of $C \setminus C'$ or $C' \setminus C$ respectively

has larger *median* similarity to x. We then remove x from the cluster not chosen. Because each of $C \setminus C'$ and $C' \setminus C$ has a majority of good points, if one of C or C' is a ground-truth cluster (with respect to S'), all good points x in the intersection will make the correct choice. C and C' are now fully disjoint.

3. If C, C' are "somewhat equal" in that $|C \setminus C'| \leq 2\nu n$ and $|C' \setminus C| \leq 2\nu n$, we make them exactly equal based on the following related procedure. Each point x in the symmetric difference of C and C' decides *in* or *out* based on whether its similarity to the $(\nu n + 1)$st most-similar point in $C \cap C'$ is larger or smaller (respectively) than its similarity to the $(\nu n + 1)$st most similar point in $S \setminus (C \cup C')$. If x is a good point in $C \setminus C'$ and C is a ground-truth cluster (with respect to S'), then x will correctly choose *in*, whereas if C' is a ground-truth cluster then x will correctly choose *out*. Thus, we can replace C and C' with a single cluster consisting of their intersection plus all points x that chose *in*, without affecting the correct clusters.

4. If none of the other cases apply, it may still be there exist C, C' such that C "somewhat contains" C' in that $|C \setminus C'| > 2\nu n$ and $0 < |C' \setminus C| \leq 2\nu n$. In this case, choose the largest such C and apply the same procedure as in Step 3, but only over the points $x \in C' \setminus C$. At the end of the procedure, we have $C \supseteq C'$ and the correct clusters have not been affected with respect to the good points.

Since all clusters remaining are laminar, we can now arrange them into a forest, which we then arbitrarily complete into a tree. ■

4.8 Other Aspects and Examples

4.8.1 Computational Hardness Results

Our framework also allows us to study computational hardness results as well. We discuss here a simple example.

Property 10 *A similarity function K satisfies the* **unique best cut** *property for the clustering problem (S, l) if $r = 2$ and $\sum_{x \in C_1, x' \in C_2} K(x, x') < \sum_{x \in A, x' \in B} K(x, x')$ for all partitions $(A, B) \neq (C_1, C_2)$ of S.*

Clearly, by design the clustering complexity of Property 10 is 1. However, we have the following computational hardness result.

Theorem 4.8.1 *List-clustering under the unique best cut property is NP-hard. That is, there exists $\epsilon > 0$ such that given a dataset S and a similarity function K satisfying the unique best cut property, it is NP-hard to produce a polynomial-length list of clusterings such that at least one is ϵ-close to the ground truth.*

Proof: It is known that the MAX-CUT problem on cubic graphs is APX-hard [12] (i.e. it is hard to approximate within a constant factor $\alpha < 1$).

We create a family $((S, l), K)$ of instances for our clustering property as follows. Let $G = (V, E)$ be an instance of the MAX-CUT problem on cubic graphs, $|V| = n$. For each vertex $i \in V$ in the graph we associate a point $x_i \in S$; for each edge $(i, j) \in E$ we define $K(x_i, x_j) = -1$, and we define $K(x_i, x_j) = 0$ for each $(i, j) \notin E$. Let $S_{V'}$ denote the set $\{x_i : i \in V'\}$. Clearly for any given cut (V_1, V_2) in $G = (V, E)$, the value of the cut is exactly

$$F(S_{V_1}, S_{V_2}) = \sum_{x \in S_{V_1}, x' \in S_{V_2}} -K(x, x').$$

Let us now add tiny perturbations to the K values so that there is a unique partition $(C_1, C_2) = (S_{V_1^*}, S_{V_2^*})$ minimizing the objective function F, and this partition corresponds to some maxcut (V_1^*, V_2^*)

of G (e.g., we can do this so that this partition corresponds to the lexicographically first such cut). By design, K now satisfies the unique best cut property for the clustering problem S with target clustering (C_1, C_2).

Define ϵ such that any clustering which is ϵ-close to the correct clustering (C_1, C_2) must be at least α-close in terms of the max-cut objective. E.g., $\epsilon < \frac{1-\alpha}{4}$ suffices because the graph G is cubic. Now, suppose a polynomial time algorithm produced a polynomial-sized list of clusterings with the guarantee that at least one clustering in the list has error at most ϵ in terms of its accuracy with respect to (C_1, C_2). In this case, we could then just evaluate the cut value for all the clusterings in the list and pick the best one. Since at least one clustering is at least ϵ-close to (C_1, C_2) by assumption, we are guaranteed that at least one is within α of the optimum cut value. ■

Note that we can get a similar results for any clustering objective F that (a) is NP-hard to approximate within a constant factor, and (b) has the smoothness property that it gives approximately the same value to any two clusterings that are almost the same.

4.8.2 Other interesting properties

An interesting relaxation of the average attraction property is to ask that there exists a cluster so that most of the points are noticeably more similar on average to other points in their own cluster than to points in all the other clusters, and that once we take out the points in that cluster the property becomes true recursively [5]. Formally:

Property 11 *A similarity function K satisfies the γ-**weak average attraction** property for the clustering problem (S, l) if there exists cluster C_r such that all examples $x \in C_r$ satisfy:*

$$K(x, C(x)) \geq K(x, S \setminus C_r) + \gamma,$$

and moreover the same holds recursively on the set $S \setminus C_r$.

We can then adapt Algorithm 2 to get the following result:

Theorem 4.8.2 *Let K be a similarity function satisfying γ-weak average attraction for the clustering problem (S, l). Using Algorithm 2 with $s(\epsilon, \gamma, k) = \frac{4}{\gamma^2} \ln\left(\frac{8k}{\epsilon\delta}\right)$ and $N(\epsilon, \gamma, k) = \left(\frac{2k}{\epsilon}\right)^{\frac{4k}{\gamma^2} \ln\left(\frac{8k}{\epsilon\delta}\right)} \ln(\frac{1}{\delta})$ we can produce a list of at most $k^{O\left(\frac{k}{\gamma^2} \ln\left(\frac{1}{\epsilon}\right) \ln\left(\frac{k}{\epsilon\delta}\right)\right)}$ clusterings such that with probability $1 - \delta$ at least one of them is ϵ-close to the ground-truth.*

Strong attraction An interesting property that falls in between the weak stability property and the average attraction property is the following:

Property 12 *The similarity function K satisfies the γ-**strong attraction** property for the clustering problem (S, l) if for all clusters C_r, $C_{r'}$, $r \neq r'$ in the ground-truth, for all $A \subset C_r$ we have*

$$K(A, C_r \setminus A) > K(A, C_{r'}) + \gamma.$$

We can interpret the strong attraction property as saying that for any two clusters C_r and $C_{r'}$ in the ground truth, for any subset $A \subset C_r$, the subset A is more attracted to the rest of its own cluster than to $C_{r'}$. It is easy to see that we cannot cluster in the tree model, and moreover we can show an lower bound on the sample complexity which is exponential. Specifically:

Theorem 4.8.3 *For $\epsilon \leq \gamma/4$, the γ-strong attraction property has $(\epsilon, 2)$ clustering complexity as large as $2^{\Omega(1/\gamma)}$.*

[5] Thanks to Sanjoy Dasgupta for pointing out that this property is satisfied on real datasets, such as the MINST dataset.

Proof: Consider $N = \frac{1}{\gamma}$ blobs of equal probability mass. Let's consider a special matching of these blobs $\{(R_1, L_1), (R_2, L_2), \ldots, (R_{N/2}, L_{N/2})\}$ and let's define $K(x, x') = 0$ if $x \in R_i$ and $x' \in L_i$ for some i and $K(x, x') = 1$ otherwise. Then each partition of these blobs into *two* pieces of equal size that fully "respects" our matching (in the sense that for all i R_i, L_i are on two different parts) satisfies Property 12 with a gap $\gamma' = 2\gamma$. The desired result then follows from the fact that the number of such partitions (which split the set of blobs into two pieces of equal "size" and fully respect our matching) is $2^{\frac{1}{2\gamma}-1}$. ■

It would be interesting to see if one could develop algorithms especially designed for this property that provides better guarantees than Algorithm 2.

4.8.3 Verification

A natural question is how hard is it (computationally) to determine if a proposed clustering of a given dataset S satisfies a given property or not. It is important to note, however, that we can always in polynomial time compute the distance between two clusterings (via a weighted matching algorithm). This then ensures that the user is able to compare in polynomial time the target/built-in clustering with any proposed clustering. So, even if it is computationally difficulty to determine if a proposed clustering of a given dataset S satisfies a certain property or not, the property is still reasonable to consider. Note that computing the distance between two the target clustering and any other clustering is the analogue of computing the empirical error rate of a given hypothesis in the PAC setting [201]; furthermore, there are many learning problems in the PAC model where the consistency problem is NP-hard (e.g. 3-Term DNF), even though the corresponding classes are learnable.

4.8.4 Examples

In all the examples below we consider symmetric similarity functions.

Strict separation and Spectral partitioning Figure 4.2 shows that it is possible for a similarity function to satisfy the strict separation property for a given clustering problem for which Theorem 4.3.2 gives a good algorithm, but nonetheless to fool a straightforward spectral clustering approach.

Consider $2k$ blobs $B_1, B_2, \ldots, B_k, B'_1, B'_2, \ldots, B'_k$ of equal probability mass. Assume that $K(x, x') = 1$ if $x \in B_i$ and $x' \in B'_i$, and $K(x, x') = 1$ if $x, x' \in B_i$ or $x, x' \in B'_i$, for all $i \in \{1, \ldots, k\}$. Assume also $K(x, x') = 0.5$ if $x \in B_i$ and $x' \in B_j$ or $x \in B'_i$ and $x' \in B'_j$, for $i \neq j$; let $K(x, x') = 0$ otherwise. See Figure 4.2 (a). Let $C_i = B_i \cup B'_i$, for all $i \in \{1, \ldots, k\}$. It is easy to verify that the clustering $C_1, \ldots, C_k$ (see Figure 4.2 (b)) is consistent with Property 4.2 (a possible value for the unknown threshold is $c = 0.7$). However for k large enough the cut of min-conductance is the one shown in Figure 4.2 (c), namely the cut that splits the graph into parts $\{B_1, B_2, \ldots, B_k\}$ and $\{B'_1, B'_2, \ldots, B'_k\}$. A direct consequence of this example is that applying a spectral clustering approach could lead to a hypothesis of high error.

Linkage-based algorithms and strong stability Figure 4.3 (a) gives an example of a similarity function that does not satisfy the strict separation property, but for large enough m, w.h.p. will satisfy the strong stability property. (This is because there are at most m^k subsets A of size k, and each one has failure probability only $e^{-O(mk)}$.) However, single-linkage using $K_{max}(C, C')$ would still work well here. Figure 4.3 (b) extends this to an example where single-linkage using $K_{max}(C, C')$ fails. Figure 4.3 (c) gives an example where strong stability is not satisfied and average linkage would fail too. However notice that the average attraction property is satisfied and Algorithm 2 will succeed.

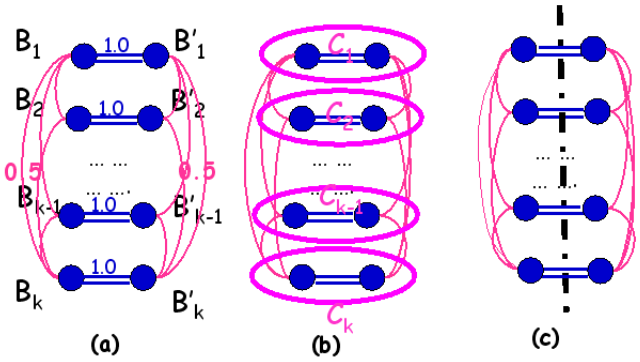

Figure 4.2: Consider $2k$ blobs $B_1, B_2, \ldots, B_k, B'_1, B'_2, \ldots, B'_k$ of equal probability mass. Points inside the same blob have similarity 1. Assume that $K(x, x') = 1$ if $x \in B_i$ and $x' \in B'_i$. Assume also $K(x, x') = 0.5$ if $x \in B_i$ and $x' \in B_j$ or $x \in B'_i$ and $x' \in B'_j$, for $i \neq j$; let $K(x, x') = 0$ otherwise. Let $C_i = B_i \cup B'_i$, for all $i \in \{1, \ldots, k\}$. It is easy to verify that the clustering $C_1, \ldots, C_k$ is consistent with Property 1 (part (b)). However, for k large enough the cut of min-conductance is the cut that splits the graph into parts $\{B_1, B_2, \ldots, B_k\}$ and $\{B'_1, B'_2, \ldots, B'_k\}$ (part (c)).

4.9 Conclusions and Discussion

In this chapter we provide a generic framework for analyzing what properties of a similarity function are sufficient to allow it to be useful for clustering, under two natural relaxations of the clustering objective. We propose a measure of the *clustering complexity* of a given property that characterizes its information-theoretic usefulness for clustering, and analyze this complexity for a broad class of properties, as well as develop efficient algorithms that are able to take advantage of them.

Our work can be viewed both in terms of providing formal advice to the *designer* of a similarity function for a given clustering task (such as clustering query search results) and in terms of advice about what *algorithms* to use given certain beliefs about the relation of the similarity function to the clustering task. Our model also provides a better understanding of when (in terms of the relation between the similarity measure and the ground-truth clustering) different hierarchical linkage-based algorithms will fare better than others. Abstractly speaking, our notion of a *property* parallels that of a *data-dependent concept class* [203] (such as large-margin separators) in the context of classification.

Open questions: Broadly, one would like to analyze other natural properties of similarity functions, as well as to further explore and formalize other models of interactive feedback. In terms of specific open questions, for the average attraction property (Property 3) we have an algorithm that for $k = 2$ produces a list of size approximately $2^{O(1/\gamma^2 \ln 1/\epsilon)}$ and a lower bound on clustering complexity of $2^{\Omega(1/\gamma)}$. One natural open question is whether one can close that gap. A second open question is that for the strong stability of large subsets property (Property 7), our algorithm produces hierarchy but has larger running time substantially larger than that for the simpler stability properties. Can an algorithm with running time polynomial in k and $1/\gamma$ be developed? Can one prove stability properties for clustering based on spectral methods, e.g., the hierarchical clustering algorithm given in [85]? More generally, it would be interesting to determine whether these stability properties can be further weakened and still admit a hierarchical clustering. Finally, in this work we have focused on formalizing clustering with non-interactive feedback. It would be interesting to formalize clustering with other natural forms of feedback.

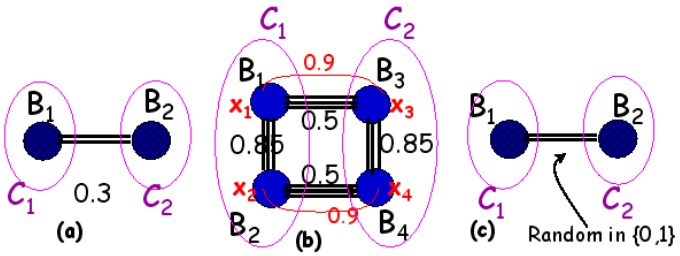

Figure 4.3: Part (a): Consider two blobs B_1, B_2 with m points each. Assume that $K(x,x') = 0.3$ if $x \in B_1$ and $x' \in B_2$, $K(x,x')$ is random in $\{0,1\}$ if $x, x' \in B_i$ for all i. Clustering C_1, C_2 does not satisfy Property 1, but for large enough m, w.h.p. will satisfy Property 5. Part (b): Consider four blobs B_1, B_2, B_3, B_4 of m points each. Assume $K(x,x') = 1$ if $x, x' \in B_i$, for all i, $K(x,x') = 0.85$ if $x \in B_1$ and $x' \in B_2$, $K(x,x') = 0.85$ if $x \in B_3$ and $x' \in B_4$, $K(x,x') = 0$ if $x \in B_1$ and $x' \in B_4$, $K(x,x') = 0$ if $x \in B_2$ and $x' \in B_3$. Now $K(x,x') = 0.5$ for all points $x \in B_1$ and $x' \in B_3$, except for two special points $x_1 \in B_1$ and $x_3 \in B_3$ for which $K(x_1,x_3) = 0.9$. Similarly $K(x,x') = 0.5$ for all points $x \in B_2$ and $x' \in B_4$, except for two special points $x_2 \in B_2$ and $x_4 \in B_4$ for which $K(x_2,x_4) = 0.9$. For large enough m, clustering C_1, C_2 satisfies Property 5. Part (c): Consider two blobs B_1, B_2 of m points each, with similarities within a blob all equal to 0.7, and similarities between blobs chosen uniformly at random from $\{0,1\}$.

Algorithm 6 Sampling Based Algorithm, List Model

Input: Data set S, similarity function K, parameters $\gamma, \epsilon > 0$, $k \in \mathbb{Z}^+$; $d_1(\epsilon,\gamma,k,\delta)$, $d_2(\epsilon,\gamma,k,\delta)$.

- Set $\mathcal{L} = \emptyset$.
- Pick a set $U = \{x_1, \ldots, x_{d_1}\}$ of d_1 random examples from S, where $d_1 = d_1(\epsilon,\gamma,k,\delta)$. Use U to define the mapping $\rho_U : X \to R^{d_1}$, $\rho_U(x) = (K(x,x_1), K(x,x_2), \ldots, K(x,x_{d_1}))$.
- Pick a set $\tilde{U}$ of $d_2 = d_2(\epsilon,\gamma,k,\delta)$ random examples from S and consider the induced set $\rho_U(\tilde{U})$.
- Consider all the $(k+1)^{d_2}$ possible labellings of the set $\rho_U(\tilde{U})$ where the $k+1$st label is used to throw out points in the ν fraction that do not satisfy the property. For each labelling use the Winnow algorithm [164, 212] to learn a multiclass linear separator h and add the clustering induced by h to $\mathcal{L}$.
- Output the list $\mathcal{L}$.

4.10 Other Proofs

Theorem 4.4.4 Let K be a similarity function satisfying the (ν,γ)-average weighted attraction property for the clustering problem (S,l). Using Algorithm 6 with parameters $d_1 = O\left(\frac{1}{\epsilon}\left(\frac{1}{\gamma^2}+1\right)\ln\left(\frac{1}{\delta}\right)\right)$ and $d_2 = O\left(\frac{1}{\epsilon}\left(\frac{1}{\gamma^2}\ln d_1 + \ln\frac{1}{\delta}\right)\right)$ we can produce a list of at most $k^{\tilde{O}\left(\frac{k}{\epsilon\gamma^2}\right)}$ clusterings such that with probability $1-\delta$ at least one of them is $\epsilon+\nu$-close to the ground-truth.

Proof:

For simplicity we describe the case $k = 2$. The generalization to larger k follows the standard multi-class to binary reduction [203].

For convenience let us assume that the labels of the two clusters are $\{-1,+1\}$ and without loss of generality assume that each of the two clusters has at least an ϵ probability mass. Let U be a random sample from S of $d_1 = \frac{1}{\epsilon}\left((4/\gamma)^2+1\right)\ln(4/\delta)$ points. We show first that with probability at least $1-\delta$,

the mapping $\rho_U : X \to R^{d_1}$ defined as

$$\rho_U(x) = (K(x, x_1), K(x, x_2), \ldots, K(x, x_{d_1}))$$

has the property that the induced distribution $\rho_U(S)$ in R^{d_1} has a separator of error at most δ (of the $1 - \nu$ fraction of the distribution satisfying the property) at L_1 margin at least $\gamma/4$.

First notice that d_1 is large enough so that with high probability our sample contains at least $d = (4/\gamma)^2 \ln(4/\delta)$ points in each cluster. Let U^+ be the subset of U consisting of the first d points of true label $+1$, and let U^- be the subset of U consisting of the first d points of true label -1. Consider the linear separator β in the ρ_U space defined as $\beta_i = l(x_i)w(x_i)$, for $x_i \in U^- \cup U^+$ and $\beta_i = 0$ otherwise. We show that, with probability at least $(1 - \delta)$, β has error at most δ at L_1 margin $\gamma/4$. Consider some fixed point $x \in S$. We begin by showing that for any such x,

$$\Pr_U \left(l(x)\beta \cdot \rho_U(x) \geq d\frac{\gamma}{4} \right) \geq 1 - \delta^2.$$

To do so, first notice that d is large enough so that with high probability, at least $1 - \delta^2$, we have both:

$$|\mathbf{E}_{x' \in U^+}[w(x')K(x, x')] - \mathbf{E}_{x' \sim S}[w(x')K(x, x')|l(x') = 1]| \leq \frac{\gamma}{4}$$

and

$$|\mathbf{E}_{x' \in U^-}[w(x')K(x, x')] - \mathbf{E}_{x' \sim S}[w(x')K(x, x')|l(x') = -1]| \leq \frac{\gamma}{4}.$$

Let's consider now the case when $l(x) = 1$. In this case we have

$$l(x)\beta \cdot \rho_U(x) = d \left(\frac{1}{d} \sum_{x_i \in U_+} w(x_i)K(x, x_i) - \frac{1}{d} \sum_{x_i \in U_-} w(x_i)K(x, x_i) \right),$$

and so combining these facts we have that with probability at least $(1 - \delta^2)$ the following holds:

$$l(x)\beta \cdot \rho_U(x) \geq d(\mathbf{E}_{x' \sim S}[w(x')K(x, x')|l(x') = 1] - \gamma/4 - \mathbf{E}_{x' \sim S}[w(x')K(x, x')|l(x') = -1] - \gamma/4).$$

This then implies that $l(x)\beta \cdot \rho_U(x) \geq d\gamma/2$. Finally, since $w(x') \in [-1, 1]$ for all x', and since $K(x, x') \in [-1, 1]$ for all pairs x, x', we have that $||\beta||_1 \leq d$ and $||\rho_U(x)||_\infty \leq 1$, which implies

$$\Pr_U \left(l(x)\frac{\beta \cdot \rho_U(x)}{||\beta||_1 ||\rho_U(x)||_\infty} \geq \frac{\gamma}{4} \right) \geq 1 - \delta^2.$$

The same analysis applies for the case that $l(x) = -1$.

Lastly, since the above holds for any x, it is also true for random $x \in S$, which implies by Markov's inequality that with probability at least $1 - \delta$, the vector β has error at most δ at L_1 margin $\gamma/4$ over $\rho_U(S)$, where examples have L_∞ norm at most 1.

So, we have proved that if K is a similarity function satisfying the $(0, \gamma)$-average weighted attraction property for the clustering problem (S, l), then with high probability there exists a low-error (at most δ) large-margin (at least $\frac{\gamma}{4}$) separator in the transformed space under mapping ρ_U. Thus, all we need now to cluster well is to draw a new fresh sample $\tilde{U}$, guess their labels (and which to throw out), map them into the transformed space using ρ_U, and then apply a good algorithm for learning linear separators in the new space that (if our guesses were correct) produces a hypothesis of error at most ϵ with probability at least $1 - \delta$. Thus we now simply need to calculate the appropriate value of d_2.

The appropriate value of d_2 can be determined as follows. Remember that the vector β has error at most δ at L_1 margin $\gamma/4$ over $\rho_U(S)$, where the mapping ρ_U produces examples of L_∞ norm at most 1. This implies that the Mistake bound of the Winnow algorithm on new labeled data (restricted to the $1-\delta$ good fraction) is $O\big(\frac{1}{\gamma^2}\ln d_1\big)$. Setting δ to be sufficiently small such that with high probability no bad points appear in the sample, and using standard mistake bound to PAC conversions [163], this then implies that a sample size of size $d_2 = O\big(\frac{1}{\epsilon}\left(\frac{1}{\gamma^2}\ln d_1 + \ln\frac{1}{\delta}\right)\big)$ is sufficient. ■

Chapter 5

Active Learning

In this chapter we return to the supervised classification setting and present some of our results on *Active Learning*. As mentioned in Chapter 1, in the active learning model [86, 94], the learning algorithm is allowed to draw random unlabeled examples from the underlying distribution and ask for the labels of any of these examples. The hope is that a good classifier can be learned with significantly fewer labels by actively directing the queries to informative examples.

As in passive supervised learning, but unlike in semi-supervised learning (which we discussed in Chapter 2), the only prior belief about the learning problem here is that the target function (or a good approximation of it) belongs to a given concept class. For some concept classes such as thresholds on the line, one can achieve an exponential improvement over the usual sample complexity of supervised learning, under no additional assumptions about the learning problem [86, 94]. In general, the speedups achievable in active learning depend on the match between the data distribution and the hypothesis class, and therefore on the target hypothesis in the class. The most noteworthy non-trivial example of improvement is the case of homogeneous (i.e., through the origin) linear separators, when the data is linearly separable and distributed uniformly over the unit sphere [94, 98, 112]. There are also simple examples where active learning does not help at all, even in the realizable case [94]. Note that in the active learning model the goal is to reduce the dependence on $1/\epsilon$ from linear or quadratic to logarithmic, and that this is somewhat orthogonal to the goals considered in Chapter 2 where the focus was on reducing the complexity of the class of functions.

In our work, we provide several new theoretical results for Active Learning. First, we prove for the first time, the feasibility of agnostic active learning. Specifically we propose and analyze the first active learning algorithm that finds an ϵ-optimal hypothesis in any hypothesis class, when the underlying distribution has arbitrary forms of noise. We also analyze margin based active learning of linear separators. We discuss these in Sections 5.1 and 5.2 below, and as mentioned in Section 1.2, these results are based on work appearing in [30, 33, 35]. Finally, in recent work [34, 41], we also show that in an asymptotic model for Active Learning where one bounds the number of queries the algorithm makes before it finds a good function (i.e. one of arbitrarily small error rate), but not the number of queries before it *knows* it has found a good function, one can obtain significantly better bounds on the number of label queries required to learn than in the traditional active learning models.

5.1 Agnostic Active Learning

In this section, we provide and analyze the first active learning algorithm that finds an ϵ-optimal hypothesis in any hypothesis class, when the underlying distribution has arbitrary forms of noise. The algorithm,

A^2 (for *A*gnostic *A*ctive), relies only upon the assumption that it has access to a stream of unlabeled examples drawn *i.i.d.* from a fixed distribution. We show that A^2 achieves an exponential improvement (i.e., requires only $O\left(\ln \frac{1}{\epsilon}\right)$ samples to find an ϵ-optimal classifier) over the usual sample complexity of supervised learning, for several settings considered before in the realizable case. These include learning threshold classifiers and learning homogeneous linear separators with respect to an input distribution which is uniform over the unit sphere.

5.1.1 Introduction

Most of the previous work on active learning has focused on the realizable case. In fact, many of the existing active learning strategies are *noise seeking* on natural learning problems, because the process of actively finding an optimal separation between one class and another often involves label queries for examples close to the decision boundary, and such examples often have a large conditional noise rate (e.g., due to a mismatch between the hypothesis class and the data distribution). Thus the most informative examples are also the ones that are typically the most noise-prone.

Consider an active learning algorithm which searches for the optimal threshold on an interval using binary search. This example is often used to demonstrate the potential of active learning in the noise-free case when there is a perfect threshold separating the classes [86]. Binary search needs $O(\ln \frac{1}{\epsilon})$ labeled examples to learn a threshold with error less than ϵ, while learning passively requires $O\left(\frac{1}{\epsilon}\right)$ labels. A fundamental drawback of this algorithm is that a small amount of adversarial noise can force the algorithm to behave badly. Is this extreme brittleness to small amounts of noise essential? Can an exponential decrease in sample complexity be achieved? Can assumptions about the mechanism producing noise be avoided? These are the questions addressed here.

Previous Work on Active Learning There has been substantial work on active learning under additional assumptions. For example, the Query by Committee analysis [112] assumes realizability (i.e., existence of a perfect classifier in a known set), and a correct Bayesian prior on the set of hypotheses. Dasgupta [94] has identified sufficient conditions (which are also necessary against an adversarially chosen distribution) for active learning given only the additional realizability assumption. There are several other papers that assume only realizability [93, 98]. If there exists a perfect hypotheses in the concept class, then any informative querying strategy can direct the learning process without the need to worry about the distribution it induces—any inconsistent hypothesis can be eliminated based on a *single* query, regardless of which distribution this query comes from. In the agnostic case, however, a hypothesis that performs badly on the query distribution may well be the optimal hypothesis with respect to the input distribution. This is the main challenge in agnostic active learning that is not present in the non-agnostic case. Burnashev and Zigangirov [75] allow noise, but require a correct Bayesian prior on threshold functions. Some papers require specific noise models such as a constant noise rate everywhere [79] or Tsybakov noise conditions [33, 78]. (In fact, in section 5.2 we discuss active learning of linear separators under a certain type of noise related to the Tsybakov noise conditions [33, 78].)

The *membership-query* setting [16, 17, 74, 137] is similar to active learning considered here, except that no unlabeled data is given. Instead, the learning algorithm is allowed to query examples of its own choice. This is problematic in several applications because natural oracles, such as hired humans, have difficulty labeling synthetic examples [46]. Ulam's Problem (quoted in [90]), where the goal is find a distinguished element in a set by asking subset membership queries, is also related. The quantity of interest is the smallest number of such queries required to find the element, given a bound on the number of queries that can be answered incorrectly. But both types of results do not apply here since an active learning strategy can only buy labels of the examples it observes. For example, a membership query

algorithm can be used to quickly hone on a separating hyperplane in a high-dimensional space. An active learning algorithm can not do so when the data distribution does not support queries close to the decision boundary.[1]

Our Contributions We present here the first *a*gnostic *a*ctive learning algorithm, A^2. The only necessary assumption is that the algorithm has access to a stream of examples drawn *i.i.d.* from some fixed distribution. No additional assumptions are made about the mechanism producing noise (e.g., class/target misfit, fundamental randomization, adversarial situations). The main contribution of our work is to prove the feasibility of agnostic active learning.

Two comments are in order:

1. We define the *noise rate* of a hypothesis class C with respect to a fixed distribution D as the minimum error rate of any hypothesis in C on D (see section 2 for a formal definition). Note that for the special case of so called *label noise* (where a coin of constant bias is used to determine whether any particular example is mislabeled with respect to the best hypothesis) these definitions coincide.
2. We regard unlabeled data as being of minimal so as to focus exclusively on the question of whether or not agnostic active learning is possible at all. Substantial follow-up to the original publication of our work [30] has successfully optimized unlabeled data usage to be on the same order as passive learning [99]. [2]

A^2 is provably correct (for any $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, it outputs an ϵ-optimal hypothesis with probability at least $1 - \delta$) and it is never harmful (it never requires significantly more labeled examples than batch learning). A^2 provides exponential sample complexity reductions in several settings previously analyzed without noise or with known noise conditions. This includes learning threshold functions with small noise with respect to ϵ and hypothesis classes consisting of homogeneous (through the origin) linear separators with the data distributed uniformly over the unit sphere in $\mathcal{R}^d$. The last example has been the most encouraging theoretical result so far in the realizable case [98].

The A^2 analysis achieves an almost contradictory property: for some sets of classifiers, an ϵ-optimal classifier can be output with fewer labeled examples than are needed to estimate the error rate of the chosen classifier with precision ϵ from random examples only.

Lower Bounds It is important to keep in mind that the speedups achievable with active learning depend on the match between the distribution over example-label pairs and the hypothesis class, and therefore on the target hypothesis in the class. Thus one should expect the results to be distribution-dependent. There are simple examples where active learning does not help at all in the model analyzed in this section, even if there is no noise [94]. These lower bounds essentially result from an "aliasing" effect and they are unavoidable in the setting we analyze in this section (where we bound the number of queries an algorithm makes before it *can prove* it has found a good function).[3]

In the noisy situation, the target function itself can be very simple (e.g., a threshold function), but if the error rate is very close to $1/2$ in a sizeable interval near the threshold, then no active learning procedure can significantly outperform passive learning. In particular, in the pure agnostic setting one

[1] Note also that much of the work on using membership queries [16, 17, 74, 137] has been focused on problems where the it was not possible to get a polynomial time learning algorithm in the passive learning setting (in a PAC sense) with the hope that the membership queries will allow learning in polynomial time. In contrast, much of the work in the Active Learning literature has been focused on reducing the sample complexity.

[2] One can show we might end up using a factor of $1/\epsilon$ more unlabeled examples than the number of labeled examples one would normally need in a passive learning setting.

[3] In recent work [34, 41], we have shown that in an asymptotic model for Active Learning where one bounds the number of queries the algorithm makes before it finds a good function (i.e. one of arbitrarily small error rate), but not the number of queries before it can prove or it knows it has found a good function, one can obtain significantly better bounds on the number of label queries required to learn.

cannot hope to achieve speedups when the noise rate ν is large, due to a lower bound of $\Omega(\frac{\nu^2}{\epsilon^2})$ on the sample complexity of any active learner [143]. However, under specific noise models (such as a constant noise rate everywhere [79] or Tsybakov noise conditions [33, 78]) and for specific classes, one can still show significant improvement over supervised learning.

Structure of this section Preliminaries and notation are covered in Section 5.1.2. A^2 is presented in Section 5.1.3; Section 5.1.3 also proves that A^2 is correct and that it is never harmful (i.e., it never requires significantly more samples than batch learning). Threshold functions such as $f_t(x) = sign(x - t)$ and homogeneous linear separators under the uniform distribution over the unit sphere are analyzed in Section 5.1.4. Conclusions, a discussion of subsequent work, and open questions are covered in Section 5.1.6.

5.1.2 Preliminaries

We consider a binary agnostic learning problem specified as follows. Let X be an instance space and $Y = \{-1, 1\}$ be the set of possible labels. Let C be the hypothesis class, a set of functions mapping from X to Y. We assume there is a distribution D over instances in X, and that the instances are labeled by a possibly randomized oracle O (i.e. the target function). The oracle O can be thought of as taking an unlabeled example x in, choosing a biased coin based on x, then flipping it to find the label -1 or 1. We let P denote the induced distribution over $X \times Y$. The *error rate* of a hypothesis h with respect to a distribution $\tilde{P}$ over $X \times Y$ is defined as $err_{\tilde{P}}(h) = \Pr_{x,y\sim\tilde{P}}[h(x) \neq y]$. The error rate $err_{\tilde{P}}(h)$ is not generally known since $\tilde{P}$ is unknown, however the empirical version $\widehat{err}_{\tilde{P}}(h) = \Pr_{x,y\sim S}[h(x) \neq y] = \frac{1}{S}\sum_{x,y\in S} I(h(x) \neq y)$ is computable based upon an observed sample set S drawn from $\tilde{P}$.

Let $\nu = \min_{h\in C}(err_{D,O}(h))$ denote the minimum error rate of any hypothesis in C with respect to the distribution (D, O) induced by D and the labeling oracle O. The goal is to find an ϵ-optimal hypothesis, i.e. a hypothesis $h \in C$ with $err_{D,O}(h)$ within ϵ of ν, where ϵ is some target error.

The algorithm A^2 relies on a subroutine, which computes a lower bound $\text{LB}(S, h, \delta)$ and an upper bound $\text{UB}(S, h, \delta)$ on the true error rate $err(h)$ of h by using a sample S of examples drawn *i.i.d.* from $\tilde{P}$. Each of these bounds must hold for all h simultaneously with probability at least $1 - \delta$. The subroutine is formally defined below.

Definition 5.1.1 *A subroutine for computing LB(S, h, δ) and UB(S, h, δ) is said to be* legal *if for all distributions $\tilde{P}$ over $X \times Y$, for all $0 < \delta < 1/2$ and $m \in \mathbb{N}$,*

$$LB(S, h, \delta) \leq \text{err}_{\tilde{P}}(h) \leq UB(S, h, \delta)$$

holds for all $h \in C$ simultaneously, with probability $1 - \delta$ over the draw of S according to $\tilde{P}^m$.

Classic examples of such subroutines are the (distribution independent) VC bound [202] and the Occam Razor bound [68], or the newer data dependent generalization bounds such as those based on Rademacher Complexities [72]. For concreteness, we could use the VC bound subroutine stated in Appendix A.1.1.

As we will see in the following section, a key point in the algorithm we present is that we will not have to bring the range close to ϵ (the desired target accuracy), but it will be enough to be constant width on a series of carefully chosen distributions over $X \times Y$.

5.1.3 The A^2 Agnostic Active Learner

At a high level, A^2 can be viewed as a robust version of the selective sampling algorithm of [86]. Selective sampling is a sequential process that keeps track of two spaces—the current *version space* C_i, defined as

the set of hypotheses in C consistent with all labels revealed so far, and the current *region of uncertainty* R_i, defined as the set of all $x \in X$, for which there exists a pair of hypotheses in C_i that disagrees on x. In round i, the algorithm picks a random unlabeled example from R_i and queries it, eliminating all hypotheses in C_i inconsistent with the received label. The algorithm then eliminates those $x \in R_i$ on which all surviving hypotheses agree, and recurses. This process fundamentally relies on the assumption that there exists a consistent hypothesis in C. In the agnostic case, a hypothesis cannot be eliminated based on its disagreement with a single example. Any algorithm must be more conservative in order to avoid risking eliminating the best hypotheses in the class.

A formal specification of A^2 is given in Algorithm 7. Let C_i be the set of hypotheses still under consideration by A^2 in round i. If all hypotheses in C_i agree on some region of the instance space, this region can be safely eliminated. To help us keep track of progress in decreasing the region of uncertainty, define $\text{DISAGREE}_D(C_i)$ as the probability that there exists a pair of hypotheses in C_i that disagrees on a random example drawn from D:

$$\text{DISAGREE}_D(C_i) = \Pr_{x \sim D}[\exists h_1, h_2 \in C_i : h_1(x) \neq h_2(x)].$$

Hence $\text{DISAGREE}_D(C_i)$ is the volume of the current region of uncertainty with respect to D.

Clearly, the ability to sample from the unlabeled data distribution D implies that ability to compute $\text{DISAGREE}_D(C_i)$. To see this, note that: $\text{DISAGREE}_D(C_i) = E_{x \sim D} I(\exists h_1, h_2 \in C_i : h_1(x) \neq h_2(x))$ is an expectation over unlabeled points drawn from D. Consequently, Chernoff bounds on the empirical expectation of a $\{0, 1\}$ random variable imply that $\text{DISAGREE}_D(C_i)$ can be estimated to any desired precision with any desired confidence using an unlabeled dataset with size limiting to infinity.

Let D_i be the distribution D restricted to the current region of uncertainty. Formally, $D_i = D(x \mid \exists h_1, h_2 \in C_i : h_1(x) \neq h_2(x))$. In round i, A^2 samples a fresh set of examples S from D_i, O, and uses it to compute upper and lower bounds for all hypotheses in C_i. It then eliminates all hypotheses whose lower bound is greater than the minimum upper bound.

Since A^2 doesn't label examples on which the surviving hypotheses agree, an optimal hypothesis in C_i with respect to D_i remains an optimal hypothesis in C_{i+1} with respect to D_{i+1}. Since each round i cuts $\text{DISAGREE}_D(C_i)$ down by half, the number of rounds is bounded by $\log \frac{1}{\epsilon}$. Sections 5.1.4 gives examples of distributions and hypothesis classes for which A^2 requires only a small number of labeled examples to transition between rounds, yielding an exponential improvement in sample complexity.

When evaluating bounds during the course of Algorithm 7, A^2 uses a schedule of δ according to the following rule: the kth bound evaluation has confidence $\delta_k = \frac{\delta}{k(k+1)}$, for $k \geq 1$. In Algorithm 7, k keeps track of the number of bound computations and i of the number of rounds.

Note: It is important to note that A^2 does not need to know ν in advance. Similarly, it does not need to know D in advance.

Correctness

Theorem 5.1.1 (Correctness) *For all C, for all (D, O), for all legal subroutines for computing UB and LB, for all $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, with probability $1 - \delta$, A^2 returns an ϵ-optimal hypothesis or does not terminate.*

Note 2 *For most "reasonable" subroutines for computing UB and LB, A^2 terminates with probability at least $1 - \delta$. For more discussion and a proof of this fact see Section 5.1.3.*

Proof: The first claim is that all bound evaluations are valid simultaneously with probability at least $1 - \delta$, and the second is that the procedure produces an ϵ-optimal hypothesis upon termination.

Algorithm 7 A^2 (allowed error rate ϵ, sampling oracle for D, labeling oracle O, hypothesis class C)

set $i \leftarrow 1$, $D_i \leftarrow D$, $C_i \leftarrow C$, $C_{i-1} \leftarrow C$, $S_{i-1} \leftarrow \emptyset$, and $k \leftarrow 1$.

(1) **while** $\textsc{Disagree}_D(C_{i-1}) \left[\min_{h \in C_{i-1}} \text{UB}(S_{i-1}, h, \mu_k) - \min_{h \in C_{i-1}} \text{LB}(S_{i-1}, h, \mu_k) \right] > \epsilon$

set $S_i \leftarrow \emptyset$, $C'_i \leftarrow C_i$, $k \leftarrow k+1$

(2) **while** $\textsc{Disagree}_D(C'_i) \geq \frac{1}{2}\textsc{Disagree}_D(C_i)$

if $\textsc{Disagree}_D(C_i)\,(\min_{h \in C_i} \text{UB}(S_i, h, \mu_k) - \min_{h \in C_i} \text{LB}(S_i, h, \mu_k)) \leq \epsilon$

$(*)$ **return** $h = \text{argmin}_{h \in C_i} \text{UB}(S_i, h, \mu_k)$.

else $S'_i =$ rejection sample $2|S_i| + 1$ samples x from D satisfying

$$\exists h_1, h_2 \in C_i : h_1(x) \neq h_2(x).$$

$S_i \leftarrow S_i \cup \{(x, O(x)) : x \in S'_i\}$, $k \leftarrow k+1$

$(**)$ $C'_i = \{h \in C_i : \text{LB}(S_i, h, \mu_k,) \leq \min_{h' \in C_i} \text{UB}(S_i, h', \mu_k)\}$, $k \leftarrow k+1$

end if

end while

$C_{i+1} \leftarrow C'_i$, $D_{i+1} \leftarrow D_i$ restricted to $\{x : \exists h_1, h_2 \in C'_i : h_1(x) \neq h_2(x)\}$

$i \leftarrow i+1$

end while

return $h = \text{argmin}_{h \in C_{i-1}} \text{UB}(S_{i-1}, h, \mu_k)$.

To prove the first claim, notice that the samples on which each bound is evaluated are drawn *i.i.d.* from some distribution over $X \times Y$. This can be verified by noting that the distribution D_i used in round i is precisely that given by drawing x from the underlying distribution D conditioned on the disagreement $\exists h_1, h_2 \in C_i : h_1(x) \neq h_2(x)$, and then labeling according to the oracle O.

The k-th bound evaluation fails with probability at most $\frac{\delta}{k(k+1)}$. By the union bound, the probability that any bound fails is less then the sum of the probabilities of individual bound failures. This sum is bounded by $\sum_{k=1}^{\infty} \frac{\delta}{k(k+1)} = \delta$.

To prove the second claim, notice first that since every bound evaluation is correct, step $(**)$ never eliminates a hypothesis that has minimum error rate with respect (D, O). Let us now introduce the following notation. For a hypothesis $h \in C$ and $G \subseteq C$ define:

$$e_{D,G,O}(h) = \Pr_{x,y \sim D,O | \exists h_1, h_2 \in G : h_1(x) \neq h_2(x)}[h(x) \neq y],$$

$$f_{D,G,O}(h) = \Pr_{x,y \sim D,O | \forall h_1, h_2 \in G : h_1(x) = h_2(x)}[h(x) \neq y].$$

Notice that $e_{D,G,O}(h)$ is in fact $err_{D_G,O}(h)$, where D_G is D conditioned on the disagreement $\exists h_1, h_2 \in G : h_1(x) \neq h_2(x)$. Moreover, given any $G \subseteq C$, the error rate of every hypothesis h decomposes into two parts as follows:

$$\begin{aligned} err_{D,O}(h) &= e_{D,G,O}(h) \cdot \textsc{Disagree}_D(G) + f_{D,G,O}(h) \cdot (1 - \textsc{Disagree}_D(G)) \\ &= err_{D_G,O}(h) \cdot \textsc{Disagree}_D(G) + f_{D,G,O}(h) \cdot (1 - \textsc{Disagree}_D(G)). \end{aligned}$$

Notice that the only term that varies with $h \in G$ in the above decomposition, is $e_{D,G,O}(h)$. Consequently, finding an ϵ-optimal hypothesis requires only bounding $err_{D_G,O}(h) \cdot \textsc{Disagree}_D(G)$ to precision ϵ. But this is exactly what the negation of the main while-loop guard does, and this is also the condition used in the first step of the second while loop of the algorithm. In other words, upon termination A^2 satisfies

$$\textsc{Disagree}_D(C_i)(\min_{h \in C_i} \text{UB}(S_i, h, \delta_k) - \min_{h \in C_i} \text{LB}(S_i, h, \delta_k)) \leq \epsilon,$$

which proves the desired result. ■

Fall-back Analysis

This section shows that A^2 is never much worse than a standard batch, bound-based algorithm in terms of the number of samples required in order to learn. (A standard example of a bound-based learning algorithm is Empirical Risk Minimization (ERM) [203].)

The sample complexity $m(\epsilon, \delta, C)$ required by a batch algorithm that uses a subroutine for computing $\text{LB}(S, h, \delta)$ and $\text{UB}(S, h, \delta)$ is defined as the minimum number of samples m such that for all $S \in X^m$, $|\text{UB}(S, h, \delta) - \text{LB}(S, h, \delta)| \leq \epsilon$ for all $h \in C$. For concreteness, this section uses the following bound on $m(\epsilon, \delta, C)$ stated as Theorem A.1.1 in Appendix A.1.1:

$$m(\epsilon, \delta, C) = \frac{64}{\epsilon^2}\left(2V_C \ln\left(\frac{12}{\epsilon}\right) + \ln\left(\frac{4}{\delta}\right)\right)$$

Here V_C is the VC-dimension of C. Assume that $m(2\epsilon, \delta, H) \leq \frac{m(\epsilon,\delta,H)}{2}$, and also that the function m is monotonically increasing in $1/\delta$. These conditions are satisfied by many subroutines for computing UB and LB, including those based on the VC-bound [202] and the Occam's Razor bound [68].

Theorem 5.1.2 *For all C, for all (D, O), for all UB and LB satisfying the assumption above, for all $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, the algorithm A^2 makes at most $2m(\epsilon, \delta', H)$ calls to the oracle O, where $\delta' = \frac{\delta}{N(\epsilon,\delta,C)(N(\epsilon,\delta,C)+1)}$ and $N(\epsilon, \delta, C)$ satisfies $N(\epsilon, \delta, C) \geq \ln \frac{1}{\epsilon} \ln m(\epsilon, \frac{\delta}{N(\epsilon,\delta,C)(N(\epsilon,\delta,C)+1)}, C)$. Here $m(\epsilon, \delta, H)$ is the sample complexity of UB and LB.*

Proof: Let $\delta_k = \frac{\delta}{k(k+1)}$ be the confidence parameter used in the k-th application of the subroutine for computing UB and LB. The proof works by finding an upper bound $N(\epsilon, \delta, C)$ on the number of bound evaluations throughout the life of the algorithm. This implies that the confidence parameter δ_k is always greater than $\delta' = \frac{\delta}{N(\epsilon,\delta,C)(N(\epsilon,\delta,C)+1)}$.

Recall that D_i is the distribution over x used on the ith iteration of the first while loop. Consider $i = 1$. If condition 2 of Algorithm A^2 is repeatedly satisfied then after labeling $m(\epsilon, \delta', C)$ examples from D_1 for all hypotheses $h \in C_1$,

$$\left|\text{UB}(S_1, h, \delta') - \text{LB}(S_1, h, \delta')\right| \leq \epsilon$$

simultaneously. Note that in these conditions A^2 safely halts. Notice also that the number of bound evaluations during this process is at most $\log_2 m(\epsilon, \delta', C)$.

On the other hand, if loop (2) ever completes and i increases, then it is enough, if you finish when $i = 2$, to have uniformly for all $h \in C_2$,

$$\left|\text{UB}(S_2, h, \delta') - \text{LB}(S_2, h, \delta')\right| \leq 2\epsilon\,.$$

(This follows from the exit conditions in the outer while-loop and the 'if' in Step 2 of A^2.) Uniformly bounding the gap between upper and lower bounds over all hypotheses $h \in C_2$ to within 2ϵ, requires $m(2\epsilon, \delta', C) \leq \frac{m(\epsilon,\delta',C)}{2}$ labeled examples from D_2 and the number of bound evaluations in round $i = 2$ is at most $\log_2 m(\epsilon, \delta', C)$.

In general, in round i it is enough to have uniformly for all $h \in C_i$,

$$\left|\text{UB}(S_i, h, \delta') - \text{LB}(S_i, h, \delta')\right| \leq 2^{i-1}\epsilon,$$

and which requires $m(2^{i-1}\epsilon, \delta', C) \leq \frac{m(\epsilon,\delta',C)}{2^{i-1}}$ labeled examples from D_i. Also the number of bound evaluations in round i is at most $\log_2 m(\epsilon, \delta', C)$.

Since the number of rounds is bounded by $\log_2 \frac{1}{\epsilon}$, it follows that the maximum number of bound evaluations throughout the life of the algorithm is at most $\log_2 \frac{1}{\epsilon} \log_2 m(\epsilon, \delta', C)$. This implies that in order to determine an upper bound $N(\epsilon, \delta, C)$ only a solution to the inequality:

$$N(\epsilon, \delta, C) \geq \log_2 \frac{1}{\epsilon} \log_2 m\left(\epsilon, \frac{\delta}{N(\epsilon, \delta, C)(N(\epsilon, \delta, C) + 1)}, C\right)$$

is required.

Finally, adding up the number of calls to the label oracle O in all rounds yields at most $2m(\epsilon, \delta', C)$ over the life of the algorithm. ■

Let V_C denote the VC-dimension of C, and let $m(\epsilon, \delta, C)$ be the number of examples required by the ERM algorithm. As stated in Theorem A.1.1 in Appendix A.1.1, a classic bound on $m(\epsilon, \delta, C)$ is $m(\epsilon, \delta, C) = \frac{64}{\epsilon^2}\left(2V_C \ln\left(\frac{12}{\epsilon}\right) + \ln\left(\frac{4}{\delta}\right)\right)$. Using Theorem 5.1.2, the following corollary holds.

Corollary 5.1.3 *For all hypothesis classes C of VC-dimension V_C, for all distributions (D, O) over $X \times Y$, for all $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, the algorithm A^2 requires at most $\tilde{O}\left(\frac{1}{\epsilon^2}(V_C \ln \frac{1}{\epsilon} + \ln \frac{1}{\delta})\right)$ labeled examples the oracle O.*

Proof: The form of $m(\epsilon, \delta, H)$ and Theorem 5.1.2 implies an upper bound on $N = N(\epsilon, \delta, H)$. It is enough to find the smallest N satisfying

$$N \geq \ln\left(\frac{1}{\epsilon}\right) \ln\left(\frac{64}{\epsilon^2}\left(2V_C \ln\left(\frac{12}{\epsilon}\right) + \ln\left(\frac{4N^2}{\delta}\right)\right)\right).$$

Using the inequality $\ln a \leq ab - \ln b - 1$ for all $a, b > 0$ and some simple algebraic manipulations, the desired upper bound on $N(\epsilon, \delta, C)$ holds. The result then follows from Theorem 5.1.2. ■

5.1.4 Active Learning Speedups

This section gives examples of exponential sample complexity improvements achieved by A^2.

Learning Threshold Functions

Linear threshold functions are the simplest and easiest to analyze class. It turns out that even for this class, exponential reductions in sample complexity are not achievable when the noise rate ν is large [143]. We prove the following three results:

1. An exponential improvement in sample complexity when the noise rate is small (Theorem 5.1.4).
2. A slower improvement when the noise rate is large (Theorem 5.1.5).
3. An exponential improvement when the noise rate is large but due to constant label noise (Theorem 5.1.6). This shows that for some forms of high noise exponential improvement remains possible.

All results in this subsection assume that subroutines LB and UB in A^2 are based on the VC bound.

Theorem 5.1.4 *Let C be the set of thresholds on an interval. For all distributions (D, O) where D is a continuous probability distribution function, for any $\epsilon < \frac{1}{2}$ and $\frac{\epsilon}{16} \geq \nu$, the algorithm A^2 makes*

$$O\left(\ln\left(\frac{1}{\epsilon}\right)\ln\left(\frac{\ln\left(\frac{1}{\epsilon\delta}\right)}{\delta}\right)\right)$$

calls to the oracle O on examples drawn i.i.d. *from D, with probability $1-\delta$.*

Proof: Consider round $i \geq 1$ of the algorithm. For $h_1, h_2 \in C_i$, let $d_i(h_1, h_2)$ be the probability that h_1 and h_2 predict differently on a random example drawn according to the distribution D_i, i.e., $d_i(h_1, h_2) = \Pr_{x\sim D_i}[h_1(x) \neq h_2(x)]$.

Let h^* be any minimum error rate hypothesis in C. Note that for any hypothesis $h \in C_i$, we have $err_{D_i,O}(h) \geq d_i(h, h^*) - err_{D_i,O}(h^*)$ and $err_{D_i,O}(h^*) \leq \nu/Z_i$, where $Z_i = \Pr_{x\sim D}[x \in [lower_i, upper_i]\,]$ is a shorthand for $\text{DISAGREE}_D(C_i)$ and $[lower_i, upper_i]$ denotes the support of D_i. Thus $err_{D_i,O}(h^*) \leq d_i(h, h^*) - \nu/Z_i$.

We will show that at least a $\frac{1}{2}$-fraction (measured with respect to D_i) of thresholds in C_i satisfy $d_i(h, h^*) \geq \frac{1}{4}$, and these thresholds are located at the ends of the interval $[lower_i, upper_i]$. Assume first that both $d_i(h^*, lower_i) \geq \frac{1}{4}$ and $d_i(h^*, upper_i) \geq \frac{1}{4}$, then let l_i and u_i be the hypotheses to the left and to the right of h^*, respectively, that satisfy $d_i(h^*, l_i) = \frac{1}{4}$ and $d_i(h^*, u_i) = \frac{1}{4}$. All $h \in [lower_i, l_i] \cup [u_i, upper_i]$ satisfy $d_i(h^*, h) \geq \frac{1}{4}$ and moreover

$$\Pr_{x\sim D_i}[x \in [lower_i, l_i] \cup [u_i, upper_i]\,] \geq \frac{1}{2}.$$

Now suppose that $d_i(h^*, lower_i) \leq \frac{1}{4}$. Let u_i be the hypothesis to the right of h^* with $d_i(h, upper_i) = \frac{1}{2}$. Then all $h \in [u_i, upper_i]$ satisfy $d_i(h^*, h) \geq \frac{1}{4}$ and moreover $\Pr_{x\sim D_i}[x \in [u_i, upper_i]\,] \geq \frac{1}{2}$. A similar argument holds for $d_i(h^*, upper_i) \leq \frac{1}{4}$.

Using the VC bound, with probability $1-\delta'$, if $|S_i| = O\left(\frac{\ln\frac{1}{\delta'}}{\left(\frac{1}{8}-\frac{\nu}{Z_i}\right)^2}\right)$, then for all hypotheses $h \in C_i$ simultaneously, $|\text{UB}(S_i, h, \delta) - \text{LB}(S_i, h, \delta)| \leq \frac{1}{8} - \frac{\nu}{Z_i}$ holds. Note that ν/Z_i is always upper bounded by $\frac{1}{16}$.

Consider a hypothesis $h \in C_i$ with $d_i(h, h^*) \geq \frac{1}{4}$. For any such h,

$$err_{D_i,O}(h) \geq d_i(h, h^*) - \nu/Z_i \geq \frac{1}{4} - \frac{\nu}{Z_i},$$

and so

$$\text{LB}(S_i, h, \delta) \geq \frac{1}{4} - \frac{\nu}{Z_i} - (\frac{1}{8} - \frac{\nu}{Z_i}) = \frac{1}{8}.$$

On the other hand, $err_{D_i,O}(h^*) \leq \frac{\nu}{Z_i}$, and so

$$\text{UB}(S_i, h^*, \delta) \leq \frac{\nu}{Z_i} + \frac{1}{8} - \frac{\nu}{Z_i} = \frac{1}{8}.$$

Thus A^2 eliminates all $h \in C_i$ with $d_i(h, h^*) \geq \frac{1}{4}$. But that means $\text{DISAGREE}_D(C_i') \leq \frac{1}{2}\text{DISAGREE}_D(C_i)$, thus terminating round i.[4]

Each exit from **while** loop (2) decreases $\text{DISAGREE}_D(C_i)$ by at least a factor of 2, implying that the number of executions is bounded by $\log \frac{1}{\epsilon}$. The algorithm makes $O\left(\ln\left(\frac{1}{\delta'}\right)\ln\left(\frac{1}{\epsilon}\right)\right)$ calls to the oracle, where $\delta' = \frac{\delta}{N(\epsilon,\delta,C)(N(\epsilon,\delta,C)+1)}$ and $N(\epsilon, \delta, C)$ is an upper bound on the number of bound evaluations throughout the life of the algorithm.

The number of bound evaluations required in round i is $O\left(\ln \frac{1}{\delta'}\right)$, which implies that $N(\epsilon, \delta, C)$ should satisfy

$$c \ln\left(\frac{N(\epsilon,\delta,C)(N(\epsilon,\delta,C)+1)}{\delta}\right)\ln\left(\frac{1}{\epsilon}\right) \leq N(\epsilon,\delta,C),$$

for some constant c. Solving this inequality completes the proof. ■

Theorem 5.1.5 below asymptotically matches a lower bound of Kääriäinen [143]. Recall that A^2 does not need to know ν in advance.

Theorem 5.1.5 *Let C be the set of thresholds on an interval. Suppose that $\epsilon < \frac{1}{2}$ and $\nu > 16\epsilon$. For all D, with probability $1-\delta$, the algorithm A^2 requires at most $\tilde{O}\left(\frac{\nu^2 \ln \frac{1}{\delta}}{\epsilon^2}\right)$ labeled samples.*

Proof: The proof is similar to the previous proof. Theorem 5.1.4 implies that loop (2) completes $\Theta(\log \frac{1}{\nu})$ times. At this point, the minimum error rate of the remaining hypotheses conditioned on disagreement becomes sufficient so that the algorithm may only halt via the return step $(*)$. In this case, $\text{DISAGREE}_D(C) = \Theta(\nu)$ implying that the number of samples required is $\tilde{O}\left(\frac{\nu^2 \ln \frac{1}{\delta}}{\epsilon^2}\right)$. ■

The final theorem is for the constant noise case where $|\Pr_{y\sim O|x}[h^*(x) \neq y] - \frac{1}{2}| = \nu$ for all $x \in X$. The theorem is similar to earlier work [75], except that we achieve these improvements with a general purpose active learning algorithm that does not use any prior over the hypothesis space or knowledge of the noise rate, and is applicable to arbitrary hypothesis spaces.

Theorem 5.1.6 *Let C be the set of thresholds on an interval. For all unlabeled data distributions D, for all labeled data distributions O, for any constant label noise $\nu < 1/2$ and $\epsilon < \frac{1}{2}$, the algorithm A^2 makes $O\left(\frac{1}{(1-2\nu)^2}\ln\left(\frac{1}{\epsilon}\right)\ln\left(\frac{\ln\left(\frac{1}{\epsilon\delta}\right)}{\delta}\right)\right)$ calls to the oracle O on examples drawn* i.i.d. *from D, with probability $1-\delta$.*

The proof is essentially the same as for Theorem 5.1.4, except that the constant label noise condition implies that the amount of noise in the remaining actively labeled subset stays bounded through the recursions.

Proof: Consider round $i \geq 1$. For $h_1, h_2 \in C_i$, let $d_i(h_1, h_2) = \Pr_{x\sim D_i}[h_1(x) \neq h_2(x)]$. Note that for any hypothesis $h \in C_i$, we have $err_{D_i,O}(h) = d_i(h, h^*)(1-2\nu) + \nu$ and $err_{D_i,O}(h^*) = \nu$, where h^* is a minimum error rate threshold.

As in the proof of Theorem 5.1.4, at least a $\frac{1}{2}$-fraction (measured with respect to D_i) of thresholds in C_i satisfy $d_i(h, h^*) \geq \frac{1}{4}$, and these thresholds are located at the ends of the support $[lower_i, upper_i]$ of

[4]The assumption in the theorem statement can be weakened to $\nu < \frac{\epsilon}{(8+\Delta)\sqrt{d}}$ for any constant $\Delta > 0$.

D_i. The VC bound implies that for any $\delta' > 0$ with probability $1 - \delta'$, if $|S_i| = O\left(\frac{\ln(1/\delta')}{(1-2\nu)^2}\right)$, then for all hypotheses $h \in C_i$ simultaneously, $|\text{UB}(S_i, h, \delta) - \text{LB}(S_i, h, \delta)| < \frac{1-2\nu}{8}$.

Consider a hypothesis $h \in C_i$ with $d_i(h, h^*) \geq \frac{1}{4}$. For any such h, $err_{D_i,O}(h) \geq \frac{1-2\nu}{4} + \nu = \frac{1}{4} + \frac{\nu}{2}$, and so $\text{LB}(S_i, h, \delta) > \frac{1}{4} + \frac{\nu}{2} - \frac{1}{8}(1 - 2\nu) = \frac{1}{8} + \frac{3\nu}{4}$. On the other hand, $err_{D_i,O}(h^*) = \nu$, and so $\text{UB}(S_i, h^*, \delta) < \nu + (\frac{1}{8} - \frac{\nu}{4}) = \frac{1}{8} + \frac{3\nu}{4}$. Thus A^2 eliminates all $h \in C_i$ with $d_i(h, h^*) \geq \frac{1}{4}$. But this means that $\text{DISAGREE}_D(C_i') \leq \frac{1}{2}\text{DISAGREE}_D(C_i)$, thus terminating round i.

Finally notice that A^2 makes $O\left(\ln\left(\frac{1}{\delta'}\right)\ln\left(\frac{1}{\epsilon}\right)\right)$ calls to the oracle, where $\delta' = \frac{\delta}{N(\epsilon,\delta,C)(N(\epsilon,\delta,C)+1)}$ and $N(\epsilon, \delta, C)$ is an upper bound on the number of bound evaluations throughout the life of the algorithm. The number of bound evaluations required in round i is $O(\ln(1/\delta'))$, which implies that the number of bound evaluations throughout the life of the algorithm $N(\epsilon, \delta, C)$ should satisfy

$$c \ln\left(\frac{N(\epsilon, \delta, C)(N(\epsilon, \delta, C) + 1)}{\delta}\right) \ln\left(\frac{1}{\epsilon}\right) \leq N(\epsilon, \delta, C),$$

for some constant c. Solving this inequality, completes the proof. ■

Linear Separators under the Uniform Distribution

A commonly analyzed case for which active learning is known to give exponential savings in the number of labeled examples is when the data is drawn uniformly from the unit sphere in $\mathcal{R}^d$, and the labels are consistent with a linear separator going through the origin. Note that even in this seemingly simple scenario, there exists an $\Omega\left(\frac{1}{\epsilon}\left(d + \log\frac{1}{\delta}\right)\right)$ lower bound on the PAC passive supervised learning sample complexity [165]. We will show that A^2 provides exponential savings in this case even in the presence of arbitrary forms of noise.

Let $X = \{x \in \mathcal{R}^d : \|x\| = 1\}$, the unit sphere in $\mathcal{R}^d$. Assume that D is uniform over X, and let C be the class of linear separators through the origin. Any $h \in C$ is a homogeneous hyperplane represented by a unit vector $w \in X$ with the classification rule $h(x) = \text{sign}(w \cdot x)$. The distance between two hypotheses u and v in C with respect to a distribution D (i.e., the probability that they predict differently on a random example drawn from D) is given by $d_D(u, v) = \frac{\arccos(u \cdot v)}{\pi}$. Finally, let $\theta(u, v) = \arccos(u \cdot v)$. Thus $d_D(u, v) = \frac{\theta(u,v)}{\pi}$.

In this section we sill use a classic lemma about the uniform distribution. For a proof see, for example, [33, 98].

Lemma 5.1.7 *For any fixed unit vector w and any $0 < \gamma \leq 1$,*

$$\frac{\gamma}{4} \leq \Pr_x\left[|w \cdot x| \leq \frac{\gamma}{\sqrt{d}}\right] \leq \gamma,$$

where x is drawn uniformly from the unit sphere.

Theorem 5.1.8 *Let X, C, and D be as defined above, and let LB and UB be the VC bound. Then for any $0 < \epsilon < \frac{1}{2}$, $0 < \nu < \frac{\epsilon}{16\sqrt{d}}$, and $\delta > 0$, with probability $1 - \delta$, A^2 requires*

$$O\left(d\left(d \ln d + \ln\frac{1}{\delta'}\right)\ln\frac{1}{\epsilon}\right)$$

calls to the labeling oracle, where $\delta' = \frac{\delta}{N(\epsilon,\delta,C)(N(\epsilon,\delta,C)+1)}$ and

$$N(\epsilon, \delta, C) = O\left(\ln\frac{1}{\epsilon}\left(d^2 \ln d + d \ln\frac{d \ln\frac{1}{\epsilon}}{\delta}\right)\right).$$

Proof: Let $w^* \in C$ be a hypothesis with the minimum error rate ν. Denote the region of uncertainty in round i by R_i. Thus $\Pr_{x\sim D}[x \in R_i] = \textsc{Disagree}_D(C_i)$. Consider round i of A^2. We prove that the round completes with high probability if a certain threshold on the number of labeled examples is reached. The round may complete with a smaller number of examples, but this is fine because the metric of progress $\textsc{Disagree}_D(C_i)$ must halve in order to complete.

Theorem A.1.1 says that it suffices to query the oracle on a set S of $O(d^2 \ln d + d \ln \frac{1}{\delta'})$ examples from ith distribution D_i to guarantee, with probability $1-\delta'$, that for all $w \in C_i$,

$$|err_{D_i,O}(w) - \widehat{err}_{D_i,O}(w)| < \frac{1}{2}\left(\frac{1}{8\sqrt{d}} - \frac{\nu}{r_i}\right),$$

where r_i is a shorthand for $\textsc{Disagree}_D(C_i)$. (By assumption, $\nu < \frac{\epsilon}{16\sqrt{d}}$ and the loop guard guarantees that $\textsc{Disagree}_D(C_i) \geq \epsilon$. Thus the precision above is at least $\frac{1}{32\sqrt{d}}$.)[5] This implies that $\text{UB}(S, w, \delta') - err_{D_i,O}(w) < \frac{1}{8\sqrt{d}} - \frac{\nu}{r_i}$, and $err_{D_i,O}(w) - \text{LB}(S, w, \delta') < \frac{1}{8\sqrt{d}} - \frac{\nu}{r_i}$. Consider any $w \in C_i$ with $d_{D_i}(w, w^*) \geq \frac{1}{4\sqrt{d}}$. For any such w, $\text{err}_{D_i,O}(w) \geq \frac{1}{4\sqrt{d}} - \frac{\nu}{r_i}$, and so

$$\text{LB}(S, w, \delta') > \frac{1}{4\sqrt{d}} - \frac{\nu}{r_i} - \frac{1}{8\sqrt{d}} + \frac{\nu}{r_i} = \frac{1}{8\sqrt{d}}.$$

However, $\text{err}_{D_i,O}(w^*) \leq \frac{\nu}{r_i}$, and thus $\text{UB}(S, w^*, \delta') < \frac{\nu}{r_i} + \frac{1}{8\sqrt{d}} - \frac{\nu}{r_i} = \frac{1}{8\sqrt{d}}$, so A^2 eliminates w in step $(**)$.

Thus round i eliminates all hypotheses $w \in C_i$ with $d_{D_i}(w, w^*) \geq \frac{1}{4\sqrt{d}}$. Since all hypotheses in C_i agree on every $x \notin R_i$,

$$d_{D_i}(w, w^*) = \frac{1}{r_i} d_D(w, w^*) = \frac{\theta(w, w^*)}{\pi r_i}.$$

Thus round i eliminates all hypotheses $w \in C_i$ with $\theta(w, w^*) \geq \frac{\pi r_i}{4\sqrt{d}}$. But since $2\theta/\pi \leq \sin\theta$, for $\theta \in (0, \frac{\pi}{2}]$, it certainly eliminates all w with $\sin\theta(w, w^*) \geq \frac{r_i}{2\sqrt{d}}$.

Consider any $x \in R_{i+1}$ and the value $|w^* \cdot x| = \cos\theta(w^*, x)$. There must exist a hypothesis $w \in C_{i+1}$ that disagrees with w^* on x; otherwise x would not be in R_{i+1}. But then $\cos\theta(w^*, x) \leq \cos(\frac{\pi}{2} - \theta(w, w^*)) = \sin\theta(w, w^*) < \frac{r_i}{2\sqrt{d}}$, where the last inequality is due to the fact that A^2 eliminates all w with $\sin\theta(w, w^*) \geq \frac{r_i}{2\sqrt{d}}$. Thus any $x \in R_{i+1}$ must satisfy $|w^* \cdot x| < \frac{r_i}{2\sqrt{d}}$. Using the fact that $\Pr[A \,|\, B] = \frac{\Pr[AB]}{\Pr[B]} \leq \frac{\Pr[A]}{\Pr[B]}$ for any A and B,

$$\Pr_{x\sim D_i}[x \in R_{i+1}] \leq \Pr_{x\sim D_i}\left[|w \cdot x| \leq \frac{r_i}{2\sqrt{d}}\right] \leq \frac{\Pr_{x\sim D}\left[|w \cdot x| \leq \frac{r_i}{2\sqrt{d}}\right]}{\Pr_{x\sim D}[x \in R_i]} \leq \frac{r_i}{2r_i} = \frac{1}{2},$$

where the third inequality follows from Lemma 5.1.7. Thus $\textsc{Disagree}_D(C_{i+1}) \leq \frac{1}{2}\textsc{Disagree}_D(C_i)$, as desired.

In order to finish the argument, it suffices to notice that since every round cuts $\textsc{Disagree}_D(C_i)$ at least in half, the total number of rounds is upper bounded by $\log\frac{1}{\epsilon}$. Notice also that the A^2 algorithm makes $O\left(d^2 \ln d + d \ln \frac{1}{\delta'}\right) \ln\left(\frac{1}{\epsilon}\right)$ calls to the oracle, where $\delta' = \frac{\delta}{N(\epsilon,\delta,C)(N(\epsilon,\delta,C)+1)}$ and $N(\epsilon, \delta, C)$ is an upper bound on the number of bound evaluations throughout the life of the algorithm. The number

[5] The assumption in the theorem statement can be weakened to $\nu < \frac{\epsilon}{(8+\Delta)\sqrt{d}}$ for any constant $\Delta > 0$.

of bound evaluations required in round i is $O\left(d^2 \ln d + d \ln \frac{1}{\delta'}\right)$. This implies that the number of bound evaluations throughout the life of the algorithm $N(\epsilon, \delta, C)$ should satisfy

$$c\left(d^2 \ln d + d \ln\left(\frac{N(\epsilon,\delta,C)(N(\epsilon,\delta,C)+1)}{\delta}\right)\right) \ln\left(\frac{1}{\epsilon}\right) \leq N(\epsilon,\delta,C),$$

for some constant c. Solving this inequality, completes the proof. ■

Note: For comparison, the query complexity of the Perceptron-based active learning algorithm of [98] is $O(d \ln \frac{1}{\epsilon\delta}(\ln \frac{d}{\delta} + \ln\ln \frac{1}{\epsilon}))$, for the same C, X, and D, but only for the realizable case when $\nu = 0$.[6] Similar bounds are obtained in [33] both in the realizable case and for a specific form of noise related to the Tsybakov small noise condition. (We present these results in Section 5.2.) The cleanest and simplest argument that exponential improvement is in principle possible in the realizable case for the same C, X, and D appears in [94]. Our work provides the first justification of why one can hope to achieve similarly strong guarantees in the much harder agnostic case, when the noise rate is sufficiently small with respect to the desired error.

5.1.5 Subsequent Work

Following the initial publication of A^2, Hanneke has further analyzed the A^2 algorithm [130], deriving a general upper bound on the number of label requests made by A^2. This bound is expressed in terms of particular quantity called the *disagreement coefficient*, which roughly quantifies how quickly the region of disagreement can grow as a function of the radius of the version space. For concreteness this bound is included below.

In addition, Dasgupta, Hsu, and Monteleoni [99] introduce and analyze a new agnostic active learning algorithm. While similar to A^2, this algorithm simplifies the maintenance of the region of uncertainty with a reduction to supervised learning, keeping track of the version space implicitly via label constraints.

Subsequent Guarantees for A^2

This section describes the disagreement coefficient [130] and the guarantees it provides for the A^2 algorithm. We begin with a few additional definitions, in the notation of Section 5.1.2.

Definition 5.1.2 *The* disagreement rate $\Delta(V)$ *of a set* $V \subseteq C$ *is defined as*

$$\Delta(V) = \Pr_{x \sim D}[x \in \textsc{Disagree}_D(V)].$$

Definition 5.1.3 *For* $h \in C$, $r > 0$, *let* $B(h, r) = \{h' \in C : d(h', h) \leq r\}$ *and define the* disagreement rate at radius r *as*

$$\Delta_r = \sup_{h \in C}(\Delta(B(h, r))).$$

The disagreement coefficient *is the infimum value of* $\theta > 0$ *such that* $\forall r > \nu + \epsilon$,

$$\Delta_r \leq \theta r.$$

We now present the main result of [130].

[6] Note also that it is not clear if the analysis in [98] is extendable to commonly used types of noise, e.g., Tsybakov noise.

Theorem 5.1.9 *If θ is the disagreement coefficient for C, then with probability at least $1-\delta$, given the inputs ϵ and δ, A^2 outputs an ϵ-optimal hypothesis h. Moreover, the number of label requests made by A^2 is at most:*

$$\tilde{O}\left(\theta^2\left(\frac{\nu^2}{\epsilon^2}+1\right)\left(V_C\ln\frac{1}{\epsilon}+\ln\frac{1}{\delta}\right)\ln\frac{1}{\epsilon}\right),$$

where $V_C \geq 1$ is the VC-dimension of C.

As shown in [130] for the concept space C of thresholds on an interval the disagreement coefficients$\theta = 2$. Also $X = \{x \in \mathcal{R}^d : \|x\| = 1\}$ is the unit sphere in $\mathcal{R}^d$, D is uniform over X, and let C be the class of linear separators through the origin, then the disagreement coefficient θ satisfies

$$\frac{1}{4}\min\left\{\pi\sqrt{d}, \frac{1}{\nu+\epsilon}\right\} \leq \theta \leq \min\left\{\pi\sqrt{d}, \frac{1}{\nu+\epsilon}\right\}.$$

These clearly match the results in Sections 5.1.4 and 5.1.4.

5.1.6 Conclusions

We present here A^2, the first active learning algorithm that finds an ϵ-optimal hypothesis in any hypothesis class, when the distribution has arbitrary forms of noise. The algorithm relies only upon the assumption that the samples are drawn *i.i.d.* from a fixed (unknown) distribution, and it does not need to know the error rate of the best classifier in the class in advance. We analyze A^2 for several settings considered before in the realizable case, showing that A^2 achieves an exponential improvement over the usual sample complexity of supervised learning in these settings. We also provide a guarantee that A^2 never requires substantially more labeled examples than passive learning.

A more general open question is what conditions are sufficient and necessary for active learning to succeed in the agnostic case. What is the right quantity that can characterize the sample complexity of agnostic active learning? As mentioned already, some progress in this direction has been recently made in [130] and [99]; however, those results characterize non-aggressive agnostic active learning. Deriving and analyzing the optimal agnostic active learning strategy is still an open question.

Much of the existing literature on active learning has been focused on binary classification; it would be interesting to analyze active learning for other loss functions. The key ingredient allowing recursion in the proof of correctness is a loss that is unvarying with respect to substantial variation over the hypothesis space. Many losses such as squared error loss do not have this property, so achieving substantial speedups, if that is possible, requires new insights. For other losses with this property (such as hinge loss or clipped squared loss), generalizations of A^2 appear straightforward.

5.2 Margin Based Active Learning

A common feature of the selective sampling algorithm [86], A^2, and others [99] is that they are all non-aggressive in their choice of query points. Even points on which there is a small amount of uncertainty are queried, rather than pursuing the maximally uncertain point. We show here that a more aggressive strategies can generally lead to better bounds. Specifically, we analyze a margin based active learning algorithm for learning linear separators and instantiate it for a few important cases, some of which have been previously considered in the literature. The generic procedure we analyze is Algorithm 8. The key contributions of this section are the following:

1. We point out that in order to get a labeled data sample complexity which has a logarithmic dependence on $1/\epsilon$ without increasing the dependence on d (i.e., a truly exponential improvement in the labeled data sample complexity over the passive learning) we have to use a strategy which is more *aggressive* than a version space strategy (the one proposed by Cohen, Atlas and Ladner in [86] and later analyzed in [30] – which we discussed in Section 5.1). We point out that this is true even in the special case when the data instances are drawn uniformly from the the unit ball in R^d, and when the labels are consistent with a linear separator going through the origin. Indeed, in order to obtain a truly exponential improvement, and to be able to learn with only $\tilde{O}\big(d \log\big(\frac{1}{\epsilon}\big)\big)$ labeled examples, we need, in each iteration, to sample our examples from a subregion carefully chosen, and not from the entire region of uncertainty, which would imply a labeled data sample complexity of $\tilde{O}\Big(d^{\frac{3}{2}} \log\big(\frac{1}{\epsilon}\big)\Big)$. The fact that a truly exponential improvement is possible in this special setting (through computationally efficient procedures) was proven before both in [98] and [112], but via more complicated and more specific arguments (and which additionally are not easily generalizable to deal with various types of noise).
2. We show that our algorithm and argument extend to the non-realizable case. A specific case we analyze here is again the setting where the data instances are drawn uniformly from the the unit ball in R^d, and a linear classifier w^* is the Bayes classifier. We additionally assume that our data satisfies the popular Tsybakov small noise condition along the decision boundary [200]. We consider both a simple version which leads to *exponential* improvement similar to the item 1 above, and a setting where we get only a polynomial improvement in the sample complexity, and where this is provably the best we can do [80]. Our analysis here for this specific cases improves significantly the work presented in Section 5.1 and the previous related work in [80].

Definitions and Notation: In this section, we consider learning linear classifiers, so C is the class of functions of the form $h(x) = \text{sign}(w \cdot x)$. As in section 5.1, we assume that the data points (x, y) are drawn from an unknown underlying distribution P over $X \times Y$ and we focus on the binary classification case (i.e., $Y = \{-1, 1\}$). Our goal is to find a classifier f with small true error where where $\text{err}(h) = \Pr_{x,y\sim P}[h(x) \neq y]$. We denote by $\text{d}(h, g)$ the probability that the two classifiers h and g predict differently on an example coming at random from P. Furthermore, for $\alpha \in [0, 1]$ we denote by $\text{B}(h, \alpha)$ the set $\{g \mid \text{d}(h, g) \leq \alpha\}$. As in section 5.1 we let D denote P_X.

In this section we focus on analyzing margin based active learning algorithms, in particular variant of Algorithm 8. Specific choices for the learning algorithm $\mathcal{A}$, sample sizes m_k, and cut-off values b_k depends on various assumptions we will make about the data, which we will investigate in details in the following sections. We note that margin based active learning algorithms have been widely used in practical applications (see e.g. [199]).

5.2.1 The Realizable Case under the Uniform Distribution

We assume here that the data instances are drawn uniformly from the the unit ball in R^d, and that the labels are consistent with a linear separator w^* going through the origin (that is $P(w^* \cdot xy \leq 0) = 0$). We assume that $\|w^*\|_2 = 1$. As mentioned in Section 5.1 even in this seemingly simple looking scenario, there exists an $\Omega\left(\frac{1}{\epsilon}\left(d + \log\frac{1}{\delta}\right)\right)$ lower bound on the PAC learning sample complexity [165].

Before presenting our better bounds, we start by informally how it is possible to get a $\tilde{O}\Big(d^{\frac{3}{2}} \log\big(\frac{1}{\epsilon}\big)\Big)$ labeled sample complexity via a margin based active learning algorithm. (Note that the analysis for the A^2 algorithm in Section 5.1.4 already implies a bound of $\tilde{O}\big(d^2 \log\big(\frac{1}{\epsilon}\big)\big)$, and as we in fact argue below that analysis can be improved to $\tilde{O}\Big(d^{\frac{3}{2}} \log\big(\frac{1}{\epsilon}\big)\Big)$ in the realizable case. We make this clearer in the note

Algorithm 8 Margin-based Active Learning.

Input: unlabeled data set $S_U = \{x_1, x_2, \ldots, \}$,
a learning algorithm $\mathcal{A}$ that learns a weight vector from labeled data,
a sequence of sample sizes $0 < \tilde{m}_1 < \tilde{m}_2 < \ldots < \tilde{m}_s = \tilde{m}_{s+1}$,
a sequence of cut-off values $b_k > 0$ $(k = 1, \ldots, s)$

Output: classifier $\hat{w}_s$

Label data points $x_1, \ldots, x_{\tilde{m}_1}$ using the oracle

iterate $k = 1, \ldots, s$

use $\mathcal{A}$ to learn weight vector $\hat{w}_k$ from the first $\tilde{m}_k$ labeled samples.

for $j = \tilde{m}_k + 1, \ldots, \tilde{m}_{k+1}$

if $|\hat{w}_k \cdot x_j| > b_k$ **then** let $y_j = \text{sign}(\hat{w}_k \cdot x_j)$

else label data point x_j using the oracle

end iterate

after Theorem 5.2.1.) Let us consider Algorithm 8, where $\mathcal{A}$ is a learning algorithm for finding a linear classifier consistent with the training data. Assume that in each iteration k, $\mathcal{A}$ finds a linear separator $\hat{w}_k$, $\|\hat{w}_k\|_2 = 1$ which is consistent with the first $\tilde{m}_k$ labeled examples. We want to ensure that $\text{err}(\hat{w}_k) \leq \frac{1}{2^k}$ (with large probability), which (by standard VC bounds) requires a sample of size $\tilde{m}_k = \tilde{O}\big(2^k d\big)$; note that this implies we need to add in each iteration about $n_k = \tilde{m}_{k+1} - \tilde{m}_k = \tilde{O}\big(2^k d\big)$ new labeled examples. The desired result will follow if we can show that by choosing appropriate b_k, we only need to ask the oracle to label $m_k = \tilde{O}(d^{3/2})$ out of the $n_k = \tilde{O}\big(2^k d\big)$ data points and ensure that all n_k data points are correctly labeled (i.e. the examples labeled automatically are in fact correctly labeled).

Note that given our assumption about the data distribution the error rate of any given separator w is $\text{err}(w) = \frac{\theta(w, w^*)}{\pi}$, where $\theta(w, w^*) = \arccos(w \cdot w^*)$. Therefore $\text{err}(\hat{w}_k) \leq 2^{-k}$ implies that $\|\hat{w}_k - w^*\|_2 \leq 2^{-k}\pi$. This implies we can *safely* label all the points with $|\hat{w}_k \cdot x| \geq 2^{-k}\pi$ because w^* and $\hat{w}_k$ predict the same on those examples. The probability of x such that $|\hat{w}_k \cdot x| \leq 2^{-k}\pi$ is $\tilde{O}(2^{-k}\sqrt{d})$ because in high dimensions, the 1-dimensional projection of uniform random variables in the unit ball is approximately a Gaussian variable with variance $1/d$. Therefore if we let $b_k = 2^{-k}\pi$ in the k-th iteration, and draw $\tilde{m}_{k+1} - \tilde{m}_k = \tilde{O}\big(2^k d\big)$ new examples to achieve an error rate of $2^{-(k+1)}$ for $\hat{w}_{k+1}$, the expected number of human labels needed is at most $\tilde{O}(d^{\frac{3}{2}})$. This essentially implies the desired result. For a high probability statement, we can use Algorithm 9, which is a modification of Algorithm 8.

Note that we can apply our favorite algorithm for finding a consistent linear separator (e.g., SVM for the realizable case, linear programming, etc.) at each iteration of Algorithm 9, and the overall procedure is *computationally efficient*.

Theorem 5.2.1 *There exists a constant C, such that for any $\epsilon, \delta > 0$, using Algorithm 9 with*

$$b_k = \frac{\pi}{2^{k-1}} \quad \text{and} \quad m_k = C d^{\frac{1}{2}} \left(d \ln d + \ln \frac{k}{\delta} \right),$$

after $s = \lceil \log_2 \frac{1}{\epsilon} \rceil$ iterations, we can efficiently find a separator of error at most ϵ with probability $1 - \delta$.

Proof: The proof is essentially a more a rigorous version of the informal one given earlier. We prove by induction on k that at the k'th iteration, with probability $1 - \delta(1 - 1/(k+1))$, we have $\text{err}(\hat{w}) \leq 2^{-k}$ for all $\hat{w}$ consistent with data in the set $W(k)$; in particular, $\text{err}(\hat{w}_k) \leq 2^{-k}$.

Algorithm 9 Margin-based Active Learning (separable case).

Input: allowed error rate ϵ, probability of failure δ, a sampling oracle for D, and a labeling oracle a sequence of sample sizes $m_k > 0$, $k \in \mathbb{Z}$; a sequence of cut-off values $b_k > 0$, $k \in \mathbb{Z}$

Output: weight vector $\hat{w}_s$ of error at most ϵ with probability $1 - \delta$

Draw m_1 examples from D, label them and put into a working set $W(1)$.

iterate $k = 1, \ldots, s$

find a hypothesis $\hat{w}_k$ ($\|\hat{w}_k\|_2 = 1$) consistent with all labeled examples in $W(k)$.

let $W(k+1) = W(k)$.

until m_{k+1} additional data points are labeled, draw sample x from D

if $|\hat{w}_k \cdot x| \geq b_k$ **then** reject x

else ask for label of x, and put into $W(k+1)$

end iterate

For $k = 1$, according to Theorem A.2.1 in Appendix A.2, we only need $m_1 = O(d + \ln(1/\delta))$ examples to obtain the desired result. In particular, we have $\mathrm{err}(\hat{w}_1) \leq 1/2$ with probability $1 - \delta/2$. Assume now the claim is true for $k-1$. Then at the k-th iteration, we can let

$$S_1 = \{x : |\hat{w}_{k-1} \cdot x| \leq b_{k-1}\} \quad \text{and} \quad S_2 = \{x : |\hat{w}_{k-1} \cdot x| > b_{k-1}\}.$$

Using the notation $\mathrm{err}(w|S) = \Pr_x((w \cdot x)(w^* \cdot x) < 0 | x \in S)$, for all $\hat{w}$ we have:

$$\mathrm{err}(\hat{w}) = \mathrm{err}(\hat{w}|S_1)\Pr(S_1) + \mathrm{err}(\hat{w}|S_2)\Pr(S_2).$$

Consider an arbitrary $\hat{w}$ consistent with the data in $W(k-1)$. By induction hypothesis, we know that with probability at least $1 - \delta(1 - 1/k)$, both $\hat{w}_{k-1}$ and $\hat{w}$ have errors at most 2^{1-k} (because both are consistent with $W(k-1)$). As discussed earlier, this implies that $\|\hat{w}_{k-1} - w^*\|_2 \leq 2^{1-k}\pi$ and $\|\hat{w} - w^*\|_2 \leq 2^{1-k}\pi$. Therefore $\forall x \in S_2$, we have

$$(\hat{w}_{k-1} \cdot x)(\hat{w} \cdot x) > 0 \quad \text{and} \quad (\hat{w}_{k-1} \cdot x)(w^* \cdot x) > .0$$

This implies that $\mathrm{err}(\hat{w}|S_2) = 0$. Now using the estimate provided in Lemma A.2.2 with $\gamma_1 = b_{k-1}$ and $\gamma_2 = 0$, we obtain $\Pr_x(S_1) \leq b_{k-1}\sqrt{4d/\pi}$. Therefore

$$\mathrm{err}(\hat{w}) \leq 2^{2-k}\sqrt{4\pi d} \cdot \mathrm{err}(\hat{w}|S_1),$$

for all $\hat{w}$ consistent with $W(k-1)$. Now, since we are labeling m_k data points in S_1 at iteration $k-1$, it follows from Theorem A.2.1 that we can find C s. t. with probability $1 - \delta/(k^2 + k)$, for all $\hat{w}$ consistent with the data in $W(k)$, $\mathrm{err}(\hat{w}|S_1)$, the error of $\hat{w}$ on S_1, is no more than $1/(4\sqrt{4\pi d})$. That is, $\mathrm{err}(\hat{w}) \leq 2^{-k}$ with probability at least $1 - \delta((1 - 1/k) + 1/(k^2 + k)) = 1 - \delta(1 - 1/(k+1))$ for all $\hat{w}$ consistent with $W(k)$, and in particular $\mathrm{err}(\hat{w}_k) \leq 2^{-k}$, as desired. ■

The choice of rejection region in Theorem 5.2.1 essentially follows the "sampling from the region of disagreement idea" idea introduced in [86] for the realizable case. As mentioned in Section 5.1, [86] suggested that one should not sample from a region (S_2 in the proof) in which all classifiers in the current version space (in our case, classifiers consistent with the labeled examples in $W(k)$) predict the same

label. In Section 5.1 and in [30] we have analyzed a more general version of the strategy proposed in [86] that is correct in the much more difficult agnostic case and we have provided theoretical analysis. Here we have used a more a refined VC-bound for the realizable case, e.g., Theorem A.2.1, to get a better bound. However, the strategy of choosing b_k in Theorem 5.2.1 (thus the idea of [86]) is not optimal. This can be seen from the proof, in which we showed $\mathrm{err}(\hat{w}_s|S_2) = 0$. If we enlarge S_2 (using a smaller b_k), we can still ensure that $\mathrm{err}(\hat{w}_s|S_2)$ is small; furthermore, $\Pr(S_1)$ becomes smaller, which allows us to use fewer labeled examples to achieve the same reduction in error. Therefore in order to show that we can achieve an improvement from $\tilde{O}\left(\frac{d}{\epsilon}\right)$ to $\tilde{O}\left(d\log\left(\frac{1}{\epsilon}\right)\right)$ as in [98], we need a more *aggressive* strategy. Specifically, at round k we set as margin parameter $b_k = \tilde{O}\left(\frac{\log(k)}{2^k\sqrt{d}}\right)$, and in consequence use fewer examples to transition between rounds. In order to prove correctness we need to refine the analysis as follows:

Theorem 5.2.2 *There exists a constant C such that for $d \geq 4$, and for any $\epsilon, \delta > 0$, $\epsilon < 1/4$, using Algorithm 9 with*

$$m_k = C\sqrt{\ln(1+k)}\left(d\ln(1+\ln k) + \ln\frac{k}{\delta}\right) \quad \text{and} \quad b_k = 2^{1-k}\pi d^{-1/2}\sqrt{5+\ln(1+k)},$$

after $s = \lceil \log_2 \frac{1}{\epsilon} \rceil - 2$ iterations, we efficiently find a separator of error $\leq \epsilon$ with probability at least $1-\delta$.

Proof: As in Theorem 5.2.1, we prove by induction on k that at the k's iteration, for $k \leq s$, with probability at least $1-\delta(1-1/(k+1))$, we $\mathrm{err}(\hat{w}) \leq 2^{-k-2}$ for all choices of $\hat{w}$ consistent with data in the working set $W(k)$; in particular $\mathrm{err}(\hat{w}_k) \leq 2^{-k-2}$.

For $k = 1$, according to Theorem A.2.1, we only need $m_k = O(d + \ln(1/\delta))$ examples to obtain the desired result; in particular, we have $\mathrm{err}(\hat{w}_1) \leq 2^{-k-2}$ with probability $1 - \delta/(k+1)$. Assume now the claim is true for $k-1$ ($k > 1$). Then at the k-th iteration, we can let

$$S_1 = \{x : |\hat{w}_{k-1} \cdot x| \leq b_{k-1}\}$$

and

$$S_2 = \{x : |\hat{w}_{k-1} \cdot x| > b_{k-1}\}.$$

Consider an arbitrary $\hat{w}$ consistent with the data in $W(k-1)$. By induction hypothesis, we know that with probability $1-\delta(1-1/k)$, both $\hat{w}_{k-1}$ and $\hat{w}$ have errors at most 2^{-k-1}, implying that

$$\theta(\hat{w}_{k-1}, w^*) \leq 2^{-k-1}\pi \quad \text{and} \quad \theta(\hat{w}, w^*) \leq 2^{-k-1}\pi.$$

Therefore $\theta(\hat{w}, \hat{w}_{k-1}) \leq 2^{-k}\pi$. Let $\tilde{\beta} = 2^{-k}\pi$ and using $\cos\tilde{\beta}/\sin\tilde{\beta} \leq 1/\tilde{\beta}$ and $\sin\tilde{\beta} \leq \tilde{\beta}$ it is easy to verify that the following inequality holds

$$b_{k-1} \geq 2\sin\tilde{\beta}d^{-1/2}\sqrt{5+\ln\left(1+\sqrt{\ln\max(1,\cos\tilde{\beta}/\sin\tilde{\beta})}\right)}.$$

By Lemma A.2.5, we have both

$$\Pr_x\left[(\hat{w}_{k-1}\cdot x)(\hat{w}\cdot x) < 0, x \in S_2\right] \leq \frac{\sin\tilde{\beta}}{e^5\cos\tilde{\beta}} \leq \frac{\sqrt{2}\tilde{\beta}}{e^5} \quad \text{and}$$

$$\Pr_x\left[(\hat{w}_{k-1}\cdot x)(w^*\cdot x) < 0, x \in S_2\right] \leq \frac{\sin\tilde{\beta}}{e^5\cos\tilde{\beta}} \leq \frac{\sqrt{2}\tilde{\beta}}{e^5}.$$

Taking the sum, we obtain

$$\Pr_x\left[(\hat{w}\cdot x)(w^*\cdot x)<0, x\in S_2\right]\leq \frac{2\sqrt{2}\tilde{\beta}}{e^5}\leq 2^{-(k+3)}.$$

Using now Lemma A.2.2 we get that for all $\hat{w}$ consistent with the data in $W(k-1)$ we have:

$$\begin{aligned}\mathrm{err}(\hat{w}) &\leq \mathrm{err}(\hat{w}|S_1)\Pr(S_1)+2^{-(k+3)}\leq \mathrm{err}(\hat{w}_k|S_1)b_{k-1}\sqrt{4d/\pi}+2^{-(k+3)}\\ &\leq 2^{-(k+2)}\left(\mathrm{err}(\hat{w}|S_1)16\sqrt{4\pi}\sqrt{5+\ln(1+k)}+1/2\right).\end{aligned}$$

Since we are labelling m_k points in S_1 at iteration $k-1$, we know from Theorem A.2.1 in Appendix A.2, that $\exists C$ s. t. with probability $1-\delta/(k+k^2)$ we have

$$\mathrm{err}(\hat{w}_k|S_1)16\sqrt{4\pi}\sqrt{5+\ln(1+k)}\leq 0.5$$

for all $\hat{w}$ consistent with $W(k)$; so, with probability $1-\delta((1-1/k)+1/(k+k^2))=1-\delta(1-1/(k+1))$, we have $\mathrm{err}(\hat{w})\leq 2^{-k-2}$ for all $\hat{w}$ consistent with $W(k)$. ■

The bound in Theorem 5.2.2 is generally better than the one in Theorem 5.2.1 due to the improved dependency on d in m_k. However, m_k depends on $\sqrt{\ln k}\ln\ln k$, for $k\leq\lceil\log_2\frac{1}{\epsilon}\rceil-2$. Therefore when $d\ll\ln k(\ln\ln k)^2$, Theorem 5.2.1 offers a better bound. Note that the strategy used in Theorem 5.2.2 is more aggressive than the strategy used in the selective sampling algorithm of [30, 86]. Indeed, we do not sample from the entire region of uncertainty – but we sample just from a subregion carefully chosen. This helps us to get rid of the undesired $d^{1/2}$. Our analysis also holds with very small modifications when the input distribution comes from a high dimensional Gaussian.

5.2.2 The Non-realizable Case under the Uniform Distribution

We show that a result similar to Theorem 5.2.2 can be obtained even for non-separable problems under a specific type of noise although not necessarily in a computationally efficient manner. The non-realizable (noisy) case for active learning in the context of classification was recently explored in [80] and as we have seen in Section 5.1 in [30, 35] as well. We consider here a model which is related to the simple one-dimensional problem in [80], which assumes that the data satisfy the increasingly popular Tsybakov small noise condition along the decision boundary[200]. We first consider a simple version which still leads to exponential convergence similar to Theorem 5.2.2. Specifically, we still assume that the data instances are drawn uniformly from the the unit ball in R^d, and a linear classifier w^* is the Bayes classifier. However, we do not assume that the Bayes error is zero. We consider the following low noise condition: there exists a known parameter $\beta>0$ such that:

$$P_x(|P(y=1|x)-P(y=-1|x)|\geq 4\beta)=1.$$

It is known that in the passive supervised learning setting this condition can lead to fast convergence rates. As we will show in this section, the condition can also be used to quantify the effectiveness of active-learning. The key point is that this assumption implies the stability condition required for active learning:

$$\beta\min\left(1,\frac{4\theta(w,w^*)}{\pi}\right)^{1/(1-\alpha)}\leq \mathrm{err}(w)-\mathrm{err}(w^*) \tag{5.2.1}$$

Algorithm 10 Margin-based Active Learning (non-separable case).

Input: allowed error rate ϵ, probability of failure δ, a sampling oracle for D, and a labeling oracle
a sequence of sample sizes $m_k > 0$, $k \in \mathbb{Z}$; a sequence of cut-off values $b_k > 0$, $k \in \mathbb{Z}$
a sequence of hypothesis space radii $r_k > 0$, $k \in \mathbb{Z}$;
a sequence of precision values $\epsilon_k > 0$, $k \in \mathbb{Z}$

Output: weight vector $\hat{w}_s$ of excess error at most ϵ with probability $1 - \delta$

Pick random $\hat{w}_0$: $\|\hat{w}_0\|_2 = 1$.

Draw m_1 examples from D, label them and put into a working set W.

iterate $k = 1, \ldots, s$

find $\hat{w}_k \in B(\hat{w}_{k-1}, r_k)$ ($\|\hat{w}_k\|_2 = 1$) to approximately minimize training error:

$\sum_{(x,y)\in W} I(\hat{w}_k \cdot xy) \leq \min_{w \in B(\hat{w}_{k-1}, r_k)} \sum_{(x,y)\in W} I(w \cdot xy) + m_k \epsilon_k$.

clear the working set W

until m_{k+1} additional data points are labeled, draw sample x from D

if $|\hat{w}_k \cdot x| \geq b_k$ **then** reject x

else ask for label of x, and put into W

end iterate

with $\alpha = 0$. We analyze here a more general setting with $\alpha \in [0, 1)$. As mentioned already, the one dimensional setting was examined in [80]. We call $\mathrm{err}(w) - \mathrm{err}(w^*)$ the *excess error* of w. In this setting, the Algorithm 9 needs to be slightly modified, as in Algorithm 10.

Theorem 5.2.3 *Let $d \geq 4$. Assume there exists a weight vector w^* s. t. the stability condition5.2.1 holds. Then there exists a constant C, s. t. for any $\epsilon, \delta > 0$, $\epsilon < \beta/8$, using Algorithm 10 with*

$$b_k = 2^{-(1-\alpha)k} \pi d^{-1/2} \sqrt{5 + \alpha k \ln 2 - \ln \beta + \ln(2 + k)},$$

$r_k = 2^{-(1-\alpha)k-2}\pi$ *for* $k > 1$, $r_1 = \pi$,

$$\epsilon_k = 2^{-\alpha(k-1)-4} \beta / \sqrt{5 + \alpha k \ln 2 - \ln \beta + \ln(1 + k)} \quad \text{and}$$

$$m_k = C \epsilon_k^{-2} \left(d + \ln \frac{k}{\delta} \right),$$

after $s = \lceil \log_2(\beta/\epsilon) \rceil$ iterations, we find a separator with excess error $\leq \epsilon$ with probability $1 - \delta$.

Proof: The proof is similar to that of Theorem 5.2.2. We prove by induction on k that after $k \leq s$ iterations, $\mathrm{err}(\hat{w}_k) - \mathrm{err}(w^*) \leq 2^{-k}\beta$ with probability $1 - \delta(1 - 1/(k + 1))$.

For $k = 1$, according to Theorem A.1.1, we only need $m_k = \beta^{-2} O(d + \ln(k/\delta))$ examples to obtain $\hat{w}_1$ with excess error $2^{-k}\beta$ with probability $1 - \delta/(k+1)$. Assume now the claim is true for $k - 1$ ($k \geq 2$). Then at the k-th iteration, we can let

$$S_1 = \{x : |\hat{w}_{k-1} \cdot x| \leq b_{k-1}\} \quad \text{and} \quad S_2 = \{x : |\hat{w}_{k-1} \cdot x| > b_{k-1}\}.$$

By induction hypothesis, we know that with probability at least $1 - \delta(1 - 1/k)$, $\hat{w}_{k-1}$ has excess errors at most $2^{-k+1}\beta$, implying $\theta(\hat{w}_{k-1}, w^*) \leq 2^{-(1-\alpha)(k-1)}\pi/4$. By assumption, $\theta(\hat{w}_{k-1}, \hat{w}_k) \leq 2^{-(1-\alpha)k-2}\pi$.

Let $\tilde{\beta} = 2^{-(1-\alpha)k-2}\pi$ and using $\cos\tilde{\beta}/\sin\tilde{\beta} \leq 1/\tilde{\beta}$ and $\sin\tilde{\beta} \leq \tilde{\beta}$, it is easy to verify that the following inequality holds:

$$b_{k-1} \geq 2\sin\tilde{\beta}d^{-1/2}\sqrt{5 + \alpha k\ln 2 - \ln\beta + \ln\left(1 + \sqrt{\ln(\cos\tilde{\beta}/\sin\tilde{\beta})}\right)}.$$

From Lemma A.2.5, we have both

$$\Pr_x\left[(\hat{w}_{k-1}\cdot x)(\hat{w}_k\cdot x) < 0, x\in S_2\right] \leq \frac{\sin\tilde{\beta}}{e^5\beta^{-1}2^{\alpha k}\cos\tilde{\beta}} \leq \frac{\sqrt{2}\tilde{\beta}\beta}{2^{\alpha k}e^5}$$

and

$$\Pr_x\left[(\hat{w}_{k-1}\cdot x)(w^*\cdot x) < 0, x\in S_2\right] \leq \frac{\sin\tilde{\beta}}{e^5\beta^{-1}2^{\alpha k}\cos\tilde{\beta}} \leq \frac{\sqrt{2}\tilde{\beta}\beta}{2^{\alpha k}e^5}.$$

Taking the sum, we obtain

$$\Pr_x\left[(\hat{w}_k\cdot x)(w^*\cdot x) < 0, x\in S_2\right] \leq \frac{2\sqrt{2}\tilde{\beta}\beta}{2^{\alpha k}e^5} \leq 2^{-(k+1)}\beta.$$

Therefore we have (using Lemma A.2.2):

$$\begin{aligned}
\mathrm{err}(\hat{w}_k) - \mathrm{err}(w^*) &\leq (\mathrm{err}(\hat{w}_k|S_1) - \mathrm{err}(w^*|S_1))\Pr(S_1) + 2^{-(k+1)}\beta \\
&\leq (\mathrm{err}(\hat{w}_k|S_1) - \mathrm{err}(w^*|S_1))b_{k-1}\sqrt{4d/\pi} + 2^{-(k+1)}\beta \\
&\leq 2^{-k}\beta\left((\mathrm{err}(\hat{w}_k|S_1) - \mathrm{err}(w^*|S_1))\sqrt{\pi}/(4\epsilon_k) + 1/2\right).
\end{aligned}$$

From Theorem A.2.1, we know we can choose C s. t. with m_k samples, we obtain

$$\mathrm{err}(\hat{w}_k|S_1) - \mathrm{err}(w^*|S_1) \leq 2\epsilon_k/\sqrt{\pi}$$

with probability $1 - \delta/(k+k^2)$. Therefore $\mathrm{err}(\hat{w}_k) \leq 2^{-k}\beta$ with probability $1 - \delta((1-1/k) + 1/(k+k^2)) = 1 - \delta(1-1/(k+1))$. ■

If $\alpha = 0$, then we can achieve exponential convergence similar to Theorem 5.2.2, even for *noisy* problems. However, for $\alpha \in (0,1)$, we have to label $\sum_k m_k = O(\epsilon^{-2\alpha}\ln(1/\epsilon)(d + \ln(s/\delta))$ examples to an achieve error rate of ϵ. That is, we only get a polynomial improvement compared to the batch learning case (with sample complexity between $O(\epsilon^{-2})$ and $O(\epsilon^{-1})$). In general, one *cannot* improve such polynomial behavior – see [80] for some simple one-dimensional examples.

Note: This bounds here improve significantly over the previous work in [30, 80]. [80] studies a similar model to ours, but for the much simpler one dimensional case. The model studied in [30] and also considered in Section 5.1 is more general, it applies to the purely agnostic setting and also the algorithm itself works generically for any concept space; however, for the specific case of learning linear separators the bounds end up having a worse quadratic rather than linear dependence on d.

Note: Instead of rejecting x when $|\hat{w}_k\cdot x| \geq b_k$, we can add them to W using the automatic labels from $\hat{w}_k$. We can then remove the requirement $\hat{w}_k \in B(\hat{w}_{k-1}, r_k)$ (thus removing the parameters r_k). The resulting procedure will have the same convergence behavior as Theorem 5.2.3 because the probability of making error by $\hat{w}_k$ when $|\hat{w}_k\cdot x| \geq b_k$ is no more than $2^{-(k+2)}\beta$.

Other Results on Margin Based Active Learning In [33] we also give an analysis of our algorithm for a case where we have a "good margin distribution", and we show how active learning can dramatically improve (the supervised learning) sample complexity in that setting as well; the bounds we obtain for that *do not depend* on the dimensionality d. We also provide a generic analysis of our main algorithm, Algorithm 8.

5.2.3 Discussion

We have shown here that a more aggressive active learning strategies can generally lead to better bounds. Note however that the analysis in this section (based on [33]) was specific to the realizable case, or done for a special type of noise. It is an open question to design aggressive agnostic active learning algorithms.

While our algorithm is computationally efficient in the realizable case, it remains an open problem to make it efficient in the general case. It is conceivable that for some special cases (e.g. the marginal distribution over the instance space is uniform, as in section 5.2.2) one could use the recent results of Kalai et. al. for Agnostically Learning Halfspaces [145]. In fact, it would be interesting to derive precise bounds (both in the realizable and the non-realizable cases) for the more general of class of log-concave distributions.

5.3 Other Results in Active Learning

In recent work, we also show that in an asymptotic model for active learning where one bounds the number of queries the algorithm makes before it finds a good function (i.e. one of arbitrarily small error rate), but not the number of queries before it *knows* it has found a good function, one can obtain significantly better bounds on the number of label queries required to learn than in the traditional active learning models. These results appear in [34, 41].

Specifically, in [34, 41] we point out that traditional analyses [94] have studied the number of label requests required before an algorithm can both produce an ϵ-good classifier and prove that the classifier's error is no more than ϵ. These studies have turned up simple examples where this number is no smaller than the number of random labeled examples required for passive learning. This is the case for learning certain *nonhomogeneous* linear separators and intervals on the real line, and generally seems to be a common problem for many learning scenarios. As such, it has led some to conclude that active learning does not help for most learning problems. In our work [34, 41] we dispel this misconception. Specifically, we study the number of labels an algorithm needs to request before it can produce an $\epsilon-$good classifier, even if there is no accessible confidence bound available to verify the quality of the classifier. With this type of analysis, we prove that active learning can essentially always achieve asymptotically superior sample complexity compared to passive learning when the VC dimension is finite. Furthermore, we find that for most natural learning problems, including the negative examples given in the previous literature, active learning can achieve exponential improvements over passive learning with respect to dependence on ϵ. Full details of the model and results can be found in [34, 41].

Chapter 6

Kernels, Margins, and Random Projections

In this chapter we return to study learning with kernel functions. As discussed in Chapter 3, a kernel is a function that takes in two data objects (which could be images, DNA sequences, or points in R^n) and outputs a number, with the property that the function is symmetric and positive-semidefinite. That is, for any kernel K, there must exist an (implicit) mapping ϕ, such that for all inputs x, x' we have $K(x, x') = \phi(x) \cdot \phi(x')$. The kernel is then used inside a "kernelized" learning algorithm such as SVM or kernel-perceptron as the way in which the algorithm interacts with the data. Furthermore even though ϕ may be a mapping into a very high-dimensional space, these algorithms have convergence rates that depend only on the *margin* γ of the best separator, and not on the dimension of the ϕ space [18, 191]. Thus, kernel functions are often viewed as providing much of the power of this implicit high-dimensional space, without paying for it computationally (because the ϕ mapping is only implicit) or in terms of sample size (if data is indeed well-separated in that space).

In this chapter, we point out that the Johnson-Lindenstrauss [96] lemma suggests that in the presence of a large margin, a kernel function can also be viewed as a mapping to a *low*-dimensional space, one of dimension only $\tilde{O}(1/\gamma^2)$. We then explore the question of whether one can efficiently produce such low-dimensional mappings, using only black-box access to a kernel function. That is, given just a program that computes $K(x, y)$ on inputs x, y of our choosing, can we efficiently construct an explicit (small) set of features that effectively capture the power of the implicit high-dimensional space? We answer this question in the affirmative if our method is also allowed black-box access to the underlying data distribution (i.e., unlabeled examples). We also give a lower bound, showing that if we do not have access to the distribution, then this is not possible for an *arbitrary* black-box kernel function.

Our positive result can be viewed as saying that designing a good kernel function is much like designing a good feature space. Given a kernel, by running it in a black-box manner on random *unlabeled* examples, we can *efficiently* generate an explicit set of $\tilde{O}(1/\gamma^2)$ features, such that if the data was linearly separable with margin γ under the kernel, then it is approximately separable in this new feature space.

6.1 Introduction

The starting point for this chapter is the observation that if a learning problem indeed has the large margin property under some kernel $K(x, y) = \phi(x) \cdot \phi(y)$, then by the Johnson-Lindenstrauss lemma, a *random* linear projection of the "ϕ-space" down to a *low* dimensional space approximately preserves linear separability [7, 21, 96, 142]. Specifically, suppose data comes from some underlying distribution D over the input space X and is labeled by some target function c. If D is such that the target function has

margin γ in the ϕ-space,[1] then a random linear projection of the ϕ-space down to a space of dimension $d = O\left(\frac{1}{\gamma^2}\log\frac{1}{\varepsilon\delta}\right)$ will, with probability at least $1-\delta$, have a linear separator with error rate at most ε (see Arriaga and Vempala [21] and also Theorem 6.4.2 in this chapter). This means that for any kernel K and margin γ, we can, in principle, think of K as mapping the input space X into an $\tilde{O}(1/\gamma^2)$-dimensional space, in essence serving as a method for representing the data in a new (and not too large) feature space.

The question we consider in this chapter is whether, given kernel K, we can in fact produce such a mapping efficiently. The problem with the above observation is that it requires explicitly computing the function $\phi(x)$. In particular, the mapping of X into R^d that results from applying the Johnson-Lindenstrauss lemma is a function $F(x) = (r_1 \cdot \phi(x), \ldots, r_d \cdot \phi(x))$, where $r_1, \ldots, r_d$ are random vectors in the ϕ-space. Since for a given kernel K, the dimensionality of the ϕ-space might be quite large, this is not efficient. Instead, what we would like is an efficient procedure that given $K(.,.)$ as a black-box program, produces a mapping with the desired properties and with running time that depends (polynomially) only on $1/\gamma$ and the time to compute the kernel function K, with no dependence on the dimensionality of the ϕ-space.

Our main result is a positive answer to this question, if our procedure for computing the mapping is also given black-box access to the distribution D (i.e., unlabeled data). Specifically, given black-box access to a kernel function $K(x, y)$, a margin value γ, access to unlabeled examples from distribution D, and parameters ε and δ, we can in polynomial time construct a mapping $F : X \to R^d$ (i.e., to a set of d real-valued features) where $d = O\left(\frac{1}{\gamma^2}\log\frac{1}{\varepsilon\delta}\right)$ with the following property. If the target concept indeed has margin γ in the ϕ-space, then with probability $1-\delta$ (over randomization in our choice of mapping function), the induced distribution in R^d is separable with error $\leq \varepsilon$. In fact, not only will the data in R^d be separable, but it will be separable with margin $\Omega(\gamma)$. Note that the logarithmic dependence on ε implies that if the learning problem has a perfect separator of margin γ in the ϕ-space, we can set ε small enough so that with high probability a set S of $O(d \log d)$ labeled examples would be perfectly separable in the mapped space. This means we could apply an arbitrary zero-noise linear-separator learning algorithm in the mapped space, such as a highly-optimized linear-programming package. However, while the dimension d has a logarithmic dependence on $1/\varepsilon$, the number of (unlabeled) examples we use to produce our mapping is $\tilde{O}(1/(\gamma^2\varepsilon))$.

To give a feel of what such a mapping might look like, suppose we are willing to use dimension $d = O(\frac{1}{\varepsilon}[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}])$ (so this is linear in $1/\varepsilon$ rather than logarithmic) and we are not concerned with preserving margins and only want approximate separability. Then we show the following simple procedure suffices. Just draw a random sample of d unlabeled points $x_1, \ldots, x_d$ from D and define $F(x) = (K(x, x_1), \ldots, K(x, x_d))$. That is, if we think of K not so much as an implicit mapping into a high-dimensional space but just as a similarity function over examples, what we are doing is drawing d "reference" points and then defining the ith feature of x to be its similarity with reference point i. We show (Corollary 6.3.2) that under the assumption that the target function has margin γ in the ϕ space, with high probability the data will be approximately separable under this mapping. Thus, this gives a particularly simple way of using the kernel and unlabeled data for feature generation, and in fact this was the motivation for the work presented in Chapter 3.

Given the above results, a natural question is whether it might be possible to perform mappings of this type without access to the underlying distribution. In Section 6.5 we show that this is in general *not* possible, given only black-box access (and polynomially-many queries) to an *arbitrary* kernel K. However, it may well be possible for specific standard kernels such as the polynomial kernel or the gaussian kernel.

[1] That is, there exists a linear separator in the ϕ-space such that any example from D is correctly classified by margin γ. See Section 6.2 for formal definitions. In Section 6.4.1 we consider the more general case that only a $1-\alpha$ fraction of the distribution D is separated by margin γ.

Relation to Support Vector Machines and Margin Bounds: Given a set S of n training examples, the kernel matrix defined over S can be viewed as placing S into an n-dimensional space, and the weight-vector found by an SVM will lie in this space and maximize the margin with respect to the training data. Our goal is to define a mapping over the entire distribution, with guarantees with respect to the distribution itself. In addition, the construction of our mapping requires only unlabeled examples, and so could be performed before seeing any labeled training data if unlabeled examples are freely available. There is, however, a close relation to margin bounds [44, 191] for SVMs (see the remark after the statement of Lemma 6.3.1 in Section 6.3), though the dimension of our output space is lower than that produced by combining SVMs with standard margin bounds.

Our goals are to some extent related to those of Ben-David et al. [48, 50]. They show negative results giving simple classes of learning problems for which one cannot construct a mapping to a low-dimensional space under which all functions in the class are linearly separable. We restrict ourselves to situations where we know that such mappings exist, but our goal is to produce them efficiently.

Interpretation: Kernel functions are often viewed as providing much of the power of an implicit high-dimensional space without having to pay for it. Our results suggest that an alternative view of kernels is as a (distribution-dependent) mapping into a low-dimensional space. In this view, designing a good kernel function is much like designing a good feature space. Given a kernel, by running it in a black-box manner on random unlabeled examples, one can efficiently generate an explicit set of $\tilde{O}(1/\gamma^2)$ features, such that if the data was linearly separable with margin γ under the kernel, then it is approximately separable using these new features.

Outline of this chapter: We begin with by giving our formal model and definitions in Section 6.2. We then in Section 6.3 show that the simple mapping described earlier in this section preserves approximate separability, and give a modification that approximately preserves both separability and margin. Both of these map data into a d-dimensional space for $d = O(\frac{1}{\varepsilon}[\frac{1}{\gamma^2} + \ln \frac{1}{\delta}])$. In Section 6.4, we give an improved mapping, that maps data to a space of dimension only $O(\frac{1}{\gamma^2} \log \frac{1}{\varepsilon\delta})$. This logarithmic dependence on $\frac{1}{\varepsilon}$ means we can set ε small enough as a function of the dimension and our input error parameter that we can then plug in a generic zero-noise linear separator algorithm in the mapped space (assuming the target function was perfectly separable with margin γ in the ϕ-space). In Section 6.5 we give a lower bound, showing that for a black-box kernel, one must have access to the underlying distribution D if one wishes to produce a good mapping into a low-dimensional space.

6.2 Notation and Definitions

We briefly introduce here the notation needed throughout the chapter. We assume that data is drawn from some distribution D over an instance space X and labeled by some unknown target function $c : X \to \{-1, +1\}$. We use P to denote the combined distribution over labeled examples.

A *kernel* K is a pairwise function $K(x, y)$ that can be viewed as a "legal" definition of inner product. Specifically, there must exist a function ϕ mapping X into a possibly high-dimensional Euclidean space such that $K(x, y) = \phi(x) \cdot \phi(y)$. We call the range of ϕ the "ϕ-space", and use $\phi(D)$ to denote the induced distribution in the ϕ-space produced by choosing random x from D and then applying $\phi(x)$.

For simplicity we focus on the $0 - 1$ loss for most of this chapter. We say that for a set S of labeled examples, a vector w in the ϕ-space has margin γ if:

$$\min_{(x,l)\in S} \left[l \frac{w \cdot \phi(x)}{||w||\ ||\phi(x)||} \right] \geq \gamma.$$

That is, w has margin γ if any labeled example in S is correctly classified by the linear separator $w \cdot \phi(x) \geq 0$, and furthermore the cosine of the angle between w and $\phi(x)$ has magnitude at least γ.[2] If such a vector w exists, then we say that S is linearly separable with margin γ under the kernel K. For simplicity, we are only considering separators that pass through the origin, though our results can be adapted to the general case as well (see Section 6.4.1).

We can similarly talk in terms of the distribution P rather than a sample S. We say that a vector w in the ϕ-space has margin γ with respect to P if:

$$\Pr_{(x,l)\sim P}\left[l\frac{w \cdot \phi(x)}{||w||\ ||\phi(x)||} < \gamma\right] = 0.$$

If such a vector w exists, then we say that P is linearly separable with margin γ under K (or just that P has margin γ in the ϕ-space). One can also weaken the notion of perfect separability. We say that a vector w in the ϕ-space has error α at margin γ if:

$$\Pr_{(x,l)\sim P}\left[l\frac{w \cdot \phi(x)}{||w||\ ||\phi(x)||} < \gamma\right] \leq \alpha.$$

Our starting assumption in this chapter will be that P is perfectly separable with margin γ under K, but we can also weaken the assumption to the existence of a vector w with error α at margin γ, with a corresponding weakening of the implications (see Section 6.4.1). Our goal is a mapping $F : X \to R^d$ where d is not too large that approximately preserves separability, and, ideally, the margin. We use $F(D)$ to denote the induced distribution in R^d produced by selecting points in X from D and then applying F, and use $F(P) = F(D, c)$ to denote the induced distribution on labeled examples.

For a set of vectors $v_1, v_2, \ldots, v_k$ in Euclidean space, let $\mathrm{span}(v_1, \ldots, v_k)$ denote the set of vectors v that can be written as a linear combination $a_1v_1 + \ldots + a_kv_k$. Also, for a vector v and a subspace Y, let $\mathrm{proj}(v, Y)$ be the orthogonal projection of v down to Y. So, for instance, $\mathrm{proj}(v, \mathrm{span}(v_1, \ldots, v_k))$ is the orthogonal projection of v down to the space spanned by $v_1, \ldots, v_k$. We note that given a set of vectors $v_1, \ldots, v_k$ and the ability to compute dot-products, this projection can be computed efficiently by solving a set of linear equalities.

6.3 Two simple mappings

Our goal is a procedure that given black-box access to a kernel function $K(.,.)$, unlabeled examples from distribution D, and a margin value γ, produces a (probability distribution over) mappings $F : X \to R^d$ with the following property: if the target function indeed has margin γ in the ϕ-space, then with high probability our mapping will approximately preserve linear separability. In this section, we analyze two methods that both produce a space of dimension $d = O(\frac{1}{\varepsilon}[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}])$, where ε is our desired bound on the error rate of the best separator in the mapped space. The second of these mappings in fact satisfies a stronger condition that its output will be approximately separable at margin $\gamma/2$ (rather than just approximately separable). This property will allow us to use this mapping as a first step in a better mapping in Section 6.4.

The following lemma is key to our analysis.

Lemma 6.3.1 *Consider any distribution over labeled examples in Euclidean space such that there exists a vector w with margin γ. Then if we draw*

$$d \geq \frac{8}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$$

[2] This is equivalent to the notion of margin in Chapter 3 since there we have assumed $||\phi(x)|| \leq 1$.

examples $z_1, \ldots, z_d$ i.i.d. from this distribution, with probability $\geq 1 - \delta$, there exists a vector w' in $\mathrm{span}(z_1, \ldots, z_d)$ *that has error at most ε at margin $\gamma/2$.*

Before proving Lemma 6.3.1, we remark that a somewhat weaker bound on d can be derived from the machinery of margin bounds. Margin bounds [44, 191] tell us that using $d = O(\frac{1}{\varepsilon}[\frac{1}{\gamma^2}\log^2(\frac{1}{\gamma\varepsilon}) + \log\frac{1}{\delta}])$ points, with probability $1 - \delta$, *any* separator with margin $\geq \gamma$ over the observed data has true error $\leq \varepsilon$. Thus, the projection of the target function w into the space spanned by the observed data will have true error $\leq \varepsilon$ as well. (Projecting w into this space maintains the value of $w \cdot z_i$, while possibly shrinking the vector w, which can only increase the margin over the observed data.) The only technical issue is that we want as a conclusion for the separator not only to have a low error rate over the distribution, but also to have a large margin. However, this can be obtained from the double-sample argument used in [44, 191] by using a $\gamma/4$-cover instead of a $\gamma/2$-cover. Margin bounds, however, are a bit of an overkill for our needs, since we are only asking for an existential statement (the *existence* of w') and not a universal statement about all separators with large empirical margins. For this reason we are able to get a better bound by a direct argument from first principles.

Proof of Lemma 6.3.1: For any set of points S, let $w_{in}(S)$ be the projection of w to $\mathrm{span}(S)$, and let $w_{out}(S)$ be the orthogonal portion of w, so that $w = w_{in}(S) + w_{out}(S)$ and $w_{in}(S) \perp w_{out}(S)$. Also, for convenience, assume w and all examples z are unit-length vectors (since we have defined margins in terms of angles, we can do this without loss of generality). Now, let us make the following definitions. Say that $w_{out}(S)$ is *large* if $\Pr_z(|w_{out}(S) \cdot z| > \gamma/2) \geq \varepsilon$, and otherwise say that $w_{out}(S)$ is *small*. Notice that if $w_{out}(S)$ is small, we are done, because

$$w \cdot z = (w_{in}(S) \cdot z) + (w_{out}(S) \cdot z),$$

which means that $w_{in}(S)$ has the properties we want. That is, there is at most an ε probability mass of points z whose dot-product with w and $w_{in}(S)$ differ by more than $\gamma/2$. So, we need only to consider what happens when $w_{out}(S)$ is large.

The crux of the proof now is that if $w_{out}(S)$ is large, this means that a new random point z has at least an ε chance of significantly improving the set S. Specifically, consider z such that $|w_{out}(S) \cdot z| > \gamma/2$. Let $z_{in}(S)$ be the projection of z to $\mathrm{span}(S)$, let $z_{out}(S) = z - z_{in}(S)$ be the portion of z orthogonal to $\mathrm{span}(S)$, and let $z' = z_{out}(S)/||z_{out}(S)||$. Now, for $S' = S \cup \{z\}$, we have

$$w_{out}(S') = w_{out}(S) - \mathrm{proj}(w_{out}(S), \mathrm{span}(S')) = w_{out}(S) - (w_{out}(S) \cdot z')z',$$

where the last equality holds because $w_{out}(S)$ is orthogonal to $\mathrm{span}(S)$ and so its projection onto $\mathrm{span}(S')$ is the same as its projection onto z'. Finally, since $w_{out}(S')$ is orthogonal to z' we have

$$||w_{out}(S')||^2 = ||w_{out}(S)||^2 - |w_{out}(S) \cdot z'|^2,$$

and since

$$|w_{out}(S) \cdot z'| \geq |w_{out}(S) \cdot z_{out}(S)| = |w_{out}(S) \cdot z|,$$

this implies by definition of z that

$$||w_{out}(S')||^2 < ||w_{out}(S)||^2 - (\gamma/2)^2.$$

So, we have a situation where so long as w_{out} is large, each example has at least an ε chance of reducing $||w_{out}||^2$ by at least $\gamma^2/4$, and since $||w||^2 = ||w_{out}(\emptyset)||^2 = 1$, this can happen at most $4/\gamma^2$ times. Chernoff bounds state that a coin of bias ε flipped $n = \frac{8}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$ times will with probability

$1-\delta$ have at least $n\varepsilon/2 \geq 4/\gamma^2$ heads. Together, these imply that with probability at least $1-\delta$, $w_{out}(S)$ will be small for $|S| \geq \frac{8}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$ as desired. ■

Lemma 6.3.1 implies that if P is linearly separable with margin γ under K, and we draw $d = \frac{8}{\varepsilon}[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}]$ random unlabeled examples $x_1, \ldots, x_d$ from D, then with probability at least $1-\delta$ there is a separator w' in the ϕ-space with error rate at most ε that can be written as

$$w' = \alpha_1\phi(x_1) + \ldots + \alpha_d\phi(x_d).$$

Notice that since $w' \cdot \phi(x) = \alpha_1 K(x, x_1) + \ldots + \alpha_d K(x, x_d)$, an immediate implication is that if we simply think of $K(x, x_i)$ as the ith "feature" of x — that is, if we define $F_1(x) = (K(x, x_1), \ldots, K(x, x_d))$ — then with high probability the vector $(\alpha_1, \ldots, \alpha_d)$ is an approximate linear separator of $F_1(P)$. So, the kernel and distribution together give us a particularly simple way of performing feature generation that preserves (approximate) separability. Formally, we have the following.

Corollary 6.3.2 *If P has margin γ in the ϕ-space, then with probability $\geq 1-\delta$, if $x_1, \ldots, x_d$ are drawn from D for $d = \frac{8}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$, the mapping*

$$F_1(x) = (K(x, x_1), \ldots, K(x, x_d))$$

produces a distribution $F_1(P)$ that is linearly separable with error at most ε.

The above mapping F_1 may not preserve margins (within a constant factor) because we do not have a good bound on the length of the vector $(\alpha_1, \ldots, \alpha_d)$ defining the separator in the new space, or the length of the examples $F_1(x)$. The key problem is that if many of the $\phi(x_i)$ are very similar, then their associated features $K(x, x_i)$ will be highly correlated. Instead, to preserve margin we want to choose an orthonormal basis of the space spanned by the $\phi(x_i)$: i.e., to do an orthogonal projection of $\phi(x)$ into this space. Specifically, let $S = \{x_1, ..., x_d\}$ be a set of of $\frac{8}{\varepsilon}[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}]$ unlabeled examples from D. We can then implement the desired orthogonal projection of $\phi(x)$ as follows. Run $K(x, y)$ for all pairs $x, y \in S$, and let $M(S) = (K(x_i, x_j))_{x_i, x_j \in S}$ be the resulting kernel matrix. Now decompose $M(S)$ into $U^T U$, where U is an upper-triangular matrix. Finally, define the mapping $F_2 : X \to R^d$ to be $F_2(x) = F_1(x)U^{-1}$, where F_1 is the mapping of Corollary 6.3.2. This is equivalent to an orthogonal projection of $\phi(x)$ into $\mathrm{span}(\phi(x_1), \ldots, \phi(x_d))$. Technically, if U is not full rank then we want to use the (Moore-Penrose) pseudoinverse [51] of U in place of U^{-1}.

We now claim that by Lemma 6.3.1, this mapping F_2 maintains approximate separability at margin $\gamma/2$.

Theorem 6.3.3 *If P has margin γ in the ϕ-space, then with probability $\geq 1-\delta$, the mapping $F_2 : X \to R^d$ for $d \geq \frac{8}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$ has the property that $F_2(P)$ is linearly separable with error at most ε at margin $\gamma/2$.*

Proof: The theorem follows directly from Lemma 6.3.1 and the fact that F_2 is an orthogonal projection. Specifically, since $\phi(D)$ is separable at margin γ, Lemma 6.3.1 implies that for $d \geq \frac{8}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$, with probability at least $1-\delta$, there exists a vector w' that can be written as $w' = \alpha_1\phi(x_1) + ... + \alpha_d\phi(x_d)$, that has error at most ε at margin $\gamma/2$ with respect to $\phi(P)$, i.e.,

$$\Pr_{(x,l)\sim P}\left[\frac{l(w' \cdot \phi(x))}{||w'||\ ||\phi(x)||} < \frac{\gamma}{2}\right] \leq \varepsilon.$$

Now consider $\overline{w} = \alpha_1 F_2(x_1) + \ldots + \alpha_d F_2(x_d)$. Since F_2 is an orthogonal projection and the $\phi(x_i)$ are clearly already in the space spanned by the $\phi(x_i)$, $\overline{w}$ can be viewed as the same as w' but just written in

a different basis. In particular, we have $||\overline{w}|| = ||w'||$, and $w' \cdot \phi(x) = \overline{w} \cdot F_2(x)$ for all $x \in X$. Since $||F_2(x)|| \leq ||\phi(x)||$ for every $x \in X$, we get that $\overline{w}$ has error at most ε at margin $\gamma/2$ with respect to $F_2(P)$, i.e.,

$$\Pr_{(x,l)\sim P}\left[\frac{l(\overline{w} \cdot F_2(x))}{||\overline{w}||\,||F_2(x)||} < \frac{\gamma}{2}\right] \leq \varepsilon.$$

Therefore, for our choice of d, with probability at least $1-\delta$ (over randomization in our choice of F_2), there exists a vector $\overline{w} \in R^d$ that has error at most ε at margin $\gamma/2$ with respect to $F_2(P)$. ■

Notice that the running time to compute $F_2(x)$ is polynomial in $1/\gamma, 1/\varepsilon, 1/\delta$ and the time to compute the kernel function K.

6.4 An improved mapping

We now describe an improved mapping, in which the dimension d has only a logarithmic, rather than linear, dependence on $1/\varepsilon$. The idea is to perform a two-stage process, composing the mapping from the previous section with a random linear projection from the range of that mapping down to the desired space. Thus, this mapping can be thought of as combining two types of random projection: a projection based on points chosen at random from D, and a projection based on choosing points uniformly at random in the intermediate space.

We begin by stating a result from [7, 21, 96, 136, 142] that we will use. Here $N(0,1)$ is the standard Normal distribution with mean 0 and variance 1 and $U(-1,1)$ is the distribution that has probability $1/2$ on -1 and probability $1/2$ on 1. Here we present the specific form given in [21].

Theorem 6.4.1 (Neuronal RP [21]) *Let $u, v \in \mathcal{R}^n$. Let $u' = \frac{1}{\sqrt{k}}uA$ and $v' = \frac{1}{\sqrt{k}}vA$ where A is a $n \times k$ random matrix whose entries are chosen independently from either $N(0,1)$ or $U(-1,1)$. Then,*

$$\Pr_A\left[(1-\varepsilon)||u-v||^2 \leq ||u'-v'||^2 \leq (1+\varepsilon)||u-v||^2\right] \geq 1 - 2e^{-(\varepsilon^2-\varepsilon^3)\frac{k}{4}}.$$

Let $F_2 : X \to R^{d_2}$ be the mapping from Section 6.3 using $\varepsilon/2$ and $\delta/2$ as its error and confidence parameters respectively. Let $\hat{F} : R^{d_2} \to R^{d_3}$ be a random projection as in Theorem 6.4.1. Specifically, we pick A to be a random $d_2 \times d_3$ matrix whose entries are chosen i.i.d. $N(0,1)$ or $U(-1,1)$. We then set $\hat{F}(x) = \frac{1}{\sqrt{d_3}}xA$. We finally consider our overall mapping $F_3 : X \to R^{d_3}$ to be $F_3(x) = \hat{F}(F_2(x))$.

We now claim that for $d_2 = O(\frac{1}{\varepsilon}[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}])$ and $d_3 = O(\frac{1}{\gamma^2}\log(\frac{1}{\varepsilon\delta}))$, with high probability, this mapping has the desired properties. The basic argument is that the initial mapping F_2 maintains approximate separability at margin $\gamma/2$ by Lemma 6.3.1, and then the second mapping approximately preserves this property by Theorem 6.4.1.

Theorem 6.4.2 *If P has margin γ in the ϕ-space, then with probability at least $1-\delta$, the mapping $F_3 = \hat{F} \circ F_2 : X \to R^{d_3}$, for values $d_2 = O\left(\frac{1}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]\right)$ and $d_3 = O\left(\frac{1}{\gamma^2}\log(\frac{1}{\varepsilon\delta})\right)$, has the property that $F_3(P)$ is linearly separable with error at most ε at margin $\gamma/4$.*

Proof:

By Lemma 6.3.1, with probability at least $1-\delta/2$ there exists a separator w in the intermediate space R^{d_2} with error at most $\varepsilon/2$ at margin $\gamma/2$. Let us assume this in fact occurs. Now, consider some point $x \in R^{d_2}$. Theorem 6.4.1 implies that a choice of $d_3 = O(\frac{1}{\gamma^2}\log(\frac{1}{\varepsilon\delta}))$ is sufficient so that under the random projection $\hat{F}$, with probability at least $1-\varepsilon\delta/4$, the squared-lengths of w, x, and $w-x$ are all preserved up to multiplicative factors of $1 \pm \gamma/16$. This then implies that the cosine of the angle between w and x

(i.e., the margin of x with respect to w) is preserved up to an additive factor of $\pm\gamma/4$. Specifically, using $\hat{x} = \frac{x}{||x||}$ and $\hat{w} = \frac{w}{||w||}$, which implies $\frac{\hat{F}(w)\cdot\hat{F}(x)}{||\hat{F}(w)||\,||\hat{F}(x)||} = \frac{\hat{F}(\hat{w})\cdot\hat{F}(\hat{x})}{||\hat{F}(\hat{w})||\,||\hat{F}(\hat{x})||}$, we have:

$$\begin{aligned}\frac{\hat{F}(\hat{w})\cdot\hat{F}(\hat{x})}{||\hat{F}(\hat{w})||\,||\hat{F}(\hat{x})||} &= \frac{\frac{1}{2}(||\hat{F}(\hat{w})||^2 + ||\hat{F}(\hat{x})||^2 - ||\hat{F}(\hat{w}) - \hat{F}(\hat{x})||^2)}{||\hat{F}(\hat{w})||\,||\hat{F}(\hat{x})||} \\ &\in [\hat{w}\cdot\hat{x} - \gamma/4, \hat{w}\cdot\hat{x} + \gamma/4].\end{aligned}$$

In other words, we have shown the following:

$$\text{For all } x, \quad \Pr_A\left[\left|\frac{w\cdot x}{||w||\,||x||} - \frac{\hat{F}(w)\cdot\hat{F}(x)}{||\hat{F}(w)||\,||\hat{F}(x)||}\right| \geq \gamma/4\right] \leq \varepsilon\delta/4.$$

Since the above is true for all x, it is clearly true for random x from $F_2(D)$. So,

$$\Pr_{x\sim F_2(D),A}\left[\left|\frac{w\cdot x}{||w||\,||x||} - \frac{\hat{F}(w)\cdot\hat{F}(x)}{||\hat{F}(w)||\,||\hat{F}(x)||}\right| \geq \gamma/4\right] \leq \varepsilon\delta/4,$$

which implies that:

$$\Pr_A\left[\Pr_{x\sim F_2(D)}\left(\left|\frac{w\cdot x}{||w||\,||x||} - \frac{\hat{F}(w)\cdot\hat{F}(x)}{||\hat{F}(w)||\,||\hat{F}(x)||}\right| \geq \gamma/4\right) \geq \varepsilon/2\right] \leq \delta/2.$$

Since w has error at most $\varepsilon/2$ at margin $\gamma/2$, this then implies that the probability that $\hat{F}(w)$ has error more than ε over $\hat{F}(F_2(D))$ at margin $\gamma/4$ is at most $\delta/2$. Combining this with the $\delta/2$ failure probability of F_2 completes the proof.

■

As before, the running time to compute our mappings is polynomial in $1/\gamma, 1/\varepsilon, 1/\delta$ and the time to compute the kernel function K.

Since the dimension d_3 of the mapping in Theorem 6.4.2 is only logarithmic in $1/\varepsilon$, this means we can set ε to be small enough so that with high probability, a sample of size $O(d_3 \log d_3)$ would be perfectly separable. This means we could use *any* noise-free linear-separator learning algorithm in R^{d_3} to learn the target concept. However, this requires using $d_2 = \tilde{O}(1/\gamma^4)$ (i.e., $\tilde{O}(1/\gamma^4)$ unlabeled examples to construct the mapping).

Corollary 6.4.3 *Given $\varepsilon', \delta, \gamma < 1$, if P has margin γ in the ϕ-space, then $\tilde{O}(\frac{1}{\varepsilon'\gamma^4})$ unlabeled examples are sufficient so that with probability $1-\delta$, mapping $F_3 : X \to R^{d_3}$ has the property that $F_3(P)$ is linearly separable with error $o(\varepsilon'/(d_3 \log d_3))$, where $d_3 = O(\frac{1}{\gamma^2}\log\frac{1}{\varepsilon'\gamma\delta})$.*

Proof: Just plug in the desired error rate into the bounds of Theorem 6.4.2. ■

6.4.1 A few extensions

So far, we have assumed that the distribution P is perfectly separable with margin γ in the ϕ-space. Suppose, however, that P is only separable with error α at margin γ. That is, there exists a vector w in the ϕ-space that correctly classifies a $1-\alpha$ probability mass of examples by margin at least γ, but the remaining α probability mass may be either within the margin or incorrectly classified. In that case, we can apply all the previous results to the $1-\alpha$ portion of the distribution that is correctly separated by

margin γ, and the remaining α probability mass of examples may or may not behave as desired. Thus all preceding results (Lemma 6.3.1, Corollary 6.3.2, Theorem 6.3.3, and Theorem 6.4.2) still hold, but with ε replaced by $(1-\alpha)\varepsilon + \alpha$ in the error rate of the resulting mapping.

Another extension is to the case that the target separator does not pass through the origin: that is, it is of the form $w \cdot \phi(x) \geq \beta$ for some value β. If ϕ is normalized, so that $||\phi(x)|| = 1$ for all $x \in X$, then all results carry over directly. In particular, all our results follow from arguments showing that the cosine of the angle between w and $\phi(x)$ changes by at most ε due to the reduction in dimension. If $\phi(x)$ is not normalized, then all results carry over with γ replaced by γ/R, where R is an upper bound on $||\phi(x)||$, as is done with standard margin bounds [44, 111, 191].

6.5 On the necessity of access to D

Our algorithms construct mappings $F : X \to R^d$ using black-box access to the kernel function $K(x, y)$ together with unlabeled examples from the input distribution D. It is natural to ask whether it might be possible to remove the need for access to D. In particular, notice that the mapping resulting from the Johnson-Lindenstrauss lemma has nothing to do with the input distribution: if we have access to the ϕ-space, then no matter what the distribution is, a random projection down to R^d will approximately preserve the existence of a large-margin separator with high probability.[3] So perhaps such a mapping F can be produced by just computing K on some polynomial number of cleverly-chosen (or uniform random) points in X. (Let us assume X is a "nice" space such as the unit ball or $\{0,1\}^n$ that can be randomly sampled.) In this section, we show this is not possible in general for an arbitrary black-box kernel. This leaves open, however, the case of specific natural kernels.

One way to view the result of this section is as follows. If we define a feature space based on uniform binary (Rademacher) or gaussian-random points in the ϕ-space, then we know this will work by the Johnson-Lindenstrauss lemma. If we define features based on points in $\phi(X)$ (the image of X under ϕ) chosen according to $\phi(D)$, then this will work by Corollary 6.3.2. However, if we define features based on points in $\phi(X)$ chosen according to some method that does not depend on D, then there will exist kernels for which this does not work.

In particular, we demonstrate the necessity of access to D as follows. Consider $X = \{0,1\}^n$, let X' be a random subset of $2^{n/2}$ elements of X, and let D be the uniform distribution on X'. For a given target function c, we will define a special ϕ-function ϕ_c such that c is a large margin separator in the ϕ-space under distribution D, but that only the points in X' behave nicely, and points not in X' provide no useful information. Specifically, consider $\phi_c : X \to R^2$ defined as:

$$\phi_c(x) = \begin{cases} (1,0) & \text{if } x \notin X' \\ (-1/2, \sqrt{3}/2) & \text{if } x \in X' \text{ and } c(x) = 1 \\ (-1/2, -\sqrt{3}/2) & \text{if } x \in X' \text{ and } c(x) = -1 \end{cases}$$

See figure 6.5.1. This then induces the kernel:

$$K_c(x,y) = \begin{cases} 1 & \text{if } x, y \notin X' \text{ or } [x, y \in X' \text{ and } c(x) = c(y)] \\ -1/2 & \text{otherwise} \end{cases}$$

Notice that the distribution $P = (D, c)$ over labeled examples has margin $\gamma = \sqrt{3}/2$ in the ϕ-space.

[3] To be clear about the order of quantification, the statement is that for any distribution, a random projection will work with high probability. However, for any given projection, there may exist bad distributions. So, even if we could define a mapping of the sort desired, we might still expect the algorithm to be randomized.

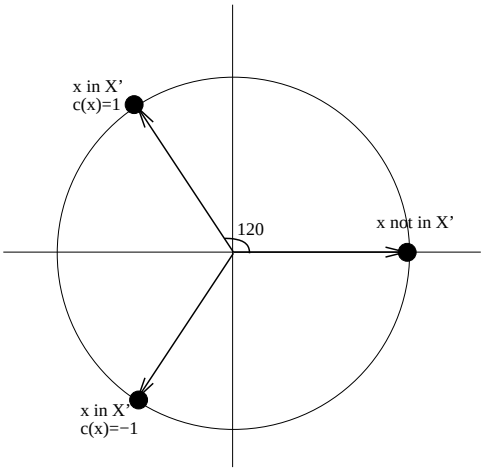

Figure 6.5.1: Function ϕ_c used in lower bound.

Theorem 6.5.1 *Suppose an algorithm makes polynomially many calls to a black-box kernel function over input space $\{0,1\}^n$ and produces a mapping $F : X \to R^d$ where d is polynomial in n. Then for random X' and random c in the above construction, with high probability $F(P)$ will not even be weakly-separable (even though P has margin $\gamma = \sqrt{3}/2$ in the ϕ-space).*

Proof: Consider any algorithm with black-box access to K attempting to create a mapping $F : X \to R^d$. Since X' is a random exponentially-small fraction of X, with high probability all calls made to K when constructing the function F are on inputs not in X'. Let us assume this indeed is the case. This implies that (a) all calls made to K when constructing the function F return the value 1, and (b) at "runtime" when x chosen from D (i.e., when F is used to map training data), even though the function $F(x)$ may itself call $K(x,y)$ for different previously-seen points y, these will all give $K(x,y) = -1/2$. In particular, this means that $F(x)$ is independent of the target function c. Finally, since X' has size $2^{n/2}$ and d is only polynomial in n, we have by simply counting the number of possible partitions of $F(X')$ by halfspaces that with high probability $F(P)$ will not even be weakly separable for a random function c over X'. Specifically, for any given halfspace, the probability over choice of c that it has error less than $1/2 - \epsilon$ is exponentially small in $|X'|$ (by Hoeffding bounds), which is doubly-exponentially small in n, whereas there are "only" $2^{O(dn)}$ possible partitions by halfspaces. ■

Notice that the kernel in the above argument is positive semidefinite. If we wish to have a positive definite kernel, we can simply change "1" to "$1-\alpha$" and "$-1/2$" to "$-\frac{1}{2}(1-\alpha)$" in the definition of $K(x,y)$, except for $y = x$ in which case we keep $K(x,y) = 1$. This corresponds to a function ϕ in which rather that mapping points exactly into R^2, we map into R^{2+2^n} giving each example a $\sqrt{\alpha}$-component in its own dimension, and we scale the first two components by $\sqrt{1-\alpha}$ to keep $\phi_c(x)$ a unit vector. The margin now becomes $\frac{\sqrt{3}}{2}(1-\alpha)$. Since the modifications provide no real change (an algorithm with access to the original kernel can simulate this one), the above arguments apply to this kernel as well.

One might complain that the kernels used in the above argument are not efficiently computable. However, this can be rectified (assuming the existence of one-way functions) by defining X' to be a cryptographically pseudorandom subset of X and c to be a pseudorandom function [125]. In this case, except for the very last step, the above argument still holds for polynomial-time algorithms. The only issue, which arises in the last step, is that we do not know any polynomial-time algorithm to test if $F(P)$ is weakly-separable in R^d (which would distinguish c from a truly-random function and provide the needed contradiction). Thus, we would need to change the conclusion of the theorem to be that "$F(P)$ is not even

weakly-learnable by a polynomial time algorithm".

Of course, these kernels are extremely unnatural, each with its own hidden target function built in. It seems quite conceivable that positive results independent of the distribution D can be achieved for standard, natural kernels.

6.6 Conclusions and Discussion

We show how given black-box access to a kernel function K and a distribution D (i.e., unlabeled examples) we can use K and D together to *efficiently* construct a new low-dimensional feature space in which to place the data that approximately preserves the desired properties of the kernel. Our procedure uses two types of "random" mappings. The first is a mapping based on random examples drawn from D that is used to construct the intermediate space, and the second is a mapping based on Rademacher/binary (or Gaussian) random vectors in the intermediate space as in the Johnson-Lindenstrauss lemma.

Our analysis suggests that designing a good kernel function is much like designing a good feature space. It also provides an alternative to "kernelizing" a learning algorithm: rather than modifying the algorithm to use kernels, one can instead construct a mapping into a low-dimensional space using the kernel and the data distribution, and then run an un-kernelized algorithm over examples drawn from the mapped distribution.

Our main concrete open question is whether, for natural standard kernel functions, one can produce mappings $F : X \to R^d$ in an oblivious manner, without using examples from the data distribution. The Johnson-Lindenstrauss lemma tells us that such mappings exist, but the goal is to produce them without explicitly computing the ϕ-function. Barring that, perhaps one can at least reduce the unlabeled sample-complexity of our approach.

On the practical side, it would be interesting to explore the alternatives that these (or other) mappings provide to widely used algorithms such as SVM, or Kernel Perceptron.

Chapter 7

Mechanism Design, Machine Learning, and Pricing Problems

In this chapter we make an explicit connection between machine learning and mechanism design. In particular, we show how Sample Complexity techniques in Statistical Learning Theory can be used to reduce problems of incentive-compatible mechanism design to standard algorithmic questions, for a wide range of revenue-maximizing problems in an unlimited (or unrestricted) supply setting.

7.1 Introduction, Problem Formulation

In recent years there has been substantial work on problems of algorithmic mechanism design. These problems typically take a form similar to classic algorithm design or approximation-algorithm questions, except that the inputs are each given by *selfish agents* who have their own interest in the outcome of the computation. As a result it is desirable that the mechanisms (the algorithms and protocol) be *incentive compatible* — meaning that it is in each agent's best interest to report its true value — so that agents do not try to game the system. This requirement can greatly complicate the design problem.

In this work we consider the design of mechanisms for one of the most fundamental economic objectives: *profit maximization*. Agents participating in such a mechanism may choose to falsely report their preferences if it might benefit them. What we show, however, is that so long as the number of agents is sufficiently large as a function of a measure of the complexity of the mechanism design problem, we can apply sample-complexity techniques from learning theory to reduce this problem to standard algorithmic questions in a broad class of settings. It is useful to think of the techniques we develop in the context of designing an auction to sell some goods or services, though they also apply in more general scenarios.

In a seminal paper Myerson [171] derives the optimal auction for selling a single item given that the bidders' true valuations for the item come from some known *prior distribution*. Following a trend in the recent computer science literature on optimal auction design, we consider the *prior-free* setting in which there is no underlying distribution on valuations and we wish to perform well for any (sufficiently large) set of bidders. In absence of a known prior distribution we will use machine learning techniques to estimate properties of the bidders' valuations. We consider the *unlimited supply* setting in which this problem is conceptually simpler because there are no infeasible allocations; though, it is often possible to obtain results for limited supply or with cost functions on the outcome via reduction to the unlimited supply case [9, 106, 124]. Research in optimal prior-free auction design is important for optimal auction design because it directly links inaccurate distributional knowledge typical of small markets with loss in performance.

Implicit in mechanism design problems is the fact that the selfish agents that will be participating in the mechanism have *private information* that is known only to them. Often this private information is simply the agent's valuation over the possible outcomes the mechanism could produce. For example, when selling a single item (with the standard assumption that an agent only cares if they get the item or not and not whether another agent gets it) this valuation is simply how much they are willing to pay for the item. There may also be *public information* associated with each agent. This information is assumed to be available to the mechanism. Such information is present in structured optimization problems such as the *knapsack auction problem* [9] and *multicast auction problem* [106] and is the natural way to generalize optimal auction design for independent but non-identically distributed prior distributions (which are considered by Myerson [171]) to the prior-free setting. There are many standard economic settings where such public information is available, e.g., in the college tuition mechanism, in-state or out-of-state residential status is public; for acquiring a loan, a consumer's credit report is public information; for automobile insurance, driving records, credit reports, and the make and color of the vehicle are public information.

A fundamental building block of an incentive compatible mechanism is an *offer*. For full generality an offer can be viewed as an incentive compatible mechanism for one agent. As an example, if we are selling multiple units of a single item, an offer could be a *take-it-or-leave-it* price per unit. A rational agent would accept such an offer if it is lower than the agent's valuation for the item and reject if it is greater. Notice that if all agents are given the same take-it-or-leave-it price then the outcome is *non-discriminatory* and the same price is paid by all winners. Prior-free auctions based on this type of non-discriminatory pricing have been considered previously (see, e.g., [124]).

One of the main motivations of this work is to explore *discriminatory pricing* in optimal auction design. There are two standard means to achieve discriminatory pricing. The first, is to discriminate based on the public information of the consumer. Naturally, loans are more costly for individuals with poor credit scores, car insurance is more expensive for drivers with points on their driving record, and college tuition at state run universities is cheaper for students that are in-state residents. In this setting a reasonable offer might be a mapping from the public information of the agents to a take-it-or-leave-it price. We refer to these types of offers as *pricing functions*. The second standard means for discriminatory pricing is to introduce similar products of different qualities and price them differently. Consumers who cannot afford the expensive high-quality version may still purchase an inexpensive low-quality version. This practice is common, for example, in software sales, electronics sales, and airline ticket sales. An offer for the multiple good setting could be a take-it-or-leave it price for each good. An agent would then be free to select the good (or bundle of goods) with the (total) price that they most prefer. We refer to these types of offers as *item pricings*.

Notice that allowing offers in the form of pricing functions and item pricings, as described above, provides richness to both algorithmic and mechanism design questions. This richness; however, is not without cost. Our performance bounds are parameterized by a suitable notion of the *complexity* of the class of allowable offers. It is natural that this kind of complexity should affect the ability of a mechanism to optimize. It is easier to approximate the optimal offer from a simple classes of offers, such as take-it-or-leave-it prices for a single item, than it is for a more complex class of offers, such as take-it-or-leave-it prices for multiple items. Our prior-free analysis makes the relationship between a mechanism's performance and the complexity of allowed offers precise.

We phrase our auction problem generically as: given some class of reasonable offers, can we construct an incentive-compatible auction that obtains profit close to the profit obtained by the optimal offer from this class? The auctions we discuss are generalizations of the random sampling auction of Goldberg et al. [121]. These auctions make use of a (non-incentive-compatible) algorithm for computing a best (or approximately best) offer from a given class for any set of consumers. Thus, we can view this construction

as reducing the optimal mechanism design problem to the optimal algorithm design problem.

The idea of the reduction is as follows. Let $\mathcal{A}$ be an algorithm (exact or approximate) for the purely algorithmic problem of finding the optimal offer in some class $\mathcal{G}$ for any given set of consumers S with known valuations. Our auction, which does not know the valuations a priori, asks the agents to report their valuations (as bids), splits agents randomly into two sets S_1 and S_2, runs the algorithm $\mathcal{A}$ separately on each set (perhaps adding an additional penalty term to the objective to penalize solutions that are too "complex" according to some measure), and then applies the offer found for S_1 to S_2 and the offer found on S_2 to S_1. The incentive compatibility of this auction allows us to assume that the agents will indeed report their true valuations. Sample-complexity techniques adapted from machine learning theory can then give a guarantee on the quality of the results if the market size is sufficiently large compared to a measure of complexity of the class of possible solutions. From an economics perspective, this can be viewed as replacing the Bayesian assumption that bidders come from a known prior distribution (e.g., as in Myerson's work [171]) with the use of learning, over a random subset S_1 of an arbitrary set of bidders S, to get enough information to apply to S_2 (and vice versa).

It is easy to see that as the size of the market grows, the law of large numbers indicates that the above approach is asymptotically optimal. This is not surprising as conventional economic wisdom suggests that even the approach of market analysis followed by the Bayesian optimal mechanism would incur negligibly small loss compared to the Bayesian optimal mechanism which was endowed with foreknowledge of the distribution. In contrast, the main contribution of this work is to give a mechanism with upper bounds on the convergence rate, i.e., the relationship between the size of the market, the approximation factor, and the complexity of the class of reasonable offers.

Our contributions: We present a general framework for reducing problems of incentive-compatible mechanism design to standard algorithmic questions, for a broad class of revenue-maximizing pricing problems. To obtain our bounds we use and extend sample-complexity techniques from machine learning theory (see [18, 69, 149, 203]) and to design our mechanisms we employ machine learning methods such as *structural risk minimization*. In general we show that an algorithm (or β-approximation) can be converted into a $(1+\epsilon)$-approximation (or $\beta(1+\epsilon)$-approximation) for the optimal mechanism design problem when the market size is at least $O(\beta\epsilon^{-2})$ times a reasonable notion of the complexity of the class of offers considered. Our formulas relating the size of the market to the approximation factor give upper bounds on the performance loss due to unknown market conditions and we view these as bounds on the *convergence rate* of our mechanism. From a learning perspective, the mechanism-design setting presents a number of technical challenges when attempting to get good bounds: in particular, the payoff function is discontinuous and asymmetric, and the payoffs for different offers are non-uniform. For example, in Section 7.3.3 we develop bounds based on a different notion of *covering number* than typically used in machine learning, in order to obtain results that are more meaningful for our setting.

We instantiate our framework for a variety of problems, some of which have been previously considered in the literature, including:

Digital Good Auction Problem: The *digital good auction problem* considers the sale of an unlimited number of units of an item to indistinguishable consumers, and has been considered by Goldberg et al. [121] and a number of subsequent papers. As argued in [121] the only reasonable offers for this setting are take-it-or-leave-it prices.

The analysis techniques developed in our work give a *simple* proof that the random sampling auction (related to that of [121]) obtains a $(1-\epsilon)$ fraction of the optimal offer as long as the market size is at least $O(\frac{h}{\epsilon^2}\log\frac{1}{\epsilon})$ (where h is an upper bound on the valuation of any agent).

Attribute Auction Problem: The *attribute auction problem* is an abstraction of the problem using dis-

criminatory prices based on public information (a.k.a., *attributes*) of the agents. A seller can often increase its profit by using discriminatory pricing: for example, the motion picture industry uses region encodings so that they can charge different prices for DVDs sold in different markets. Further, in many generalizations of the digital good auction problem, the agents are distinguishable via public information so the techniques exposed in the study of attribute auctions are fundamental to the study of profit maximization in general settings.

Here a reasonable class of offers to consider are mappings from the agents' attributes to take-it-or-leave-it prices. As such, we refer to these offers as *pricing functions*. For example, for one-dimensional attributes, a natural class of pricing functions might be piece-wise constant functions with k prices, as studied in [60]. In our work we give a *general* treatment that can be applied to arbitrary classes of pricing functions. For example, if attributes are multi-dimensional, pricing functions might involve partitioning agents into markets defined by coordinate values or by some natural clustering, and then offering a constant price or a price that is some other simple function of the attributes within each market. Our bounds give a $(1+\epsilon)$-approximation when the market size is large in comparison to ϵ^{-2} scaled by a suitable notion of the complexity of the class of offers.

Combinatorial Auction Problem: We also consider the goal of profit maximization in an unlimited-supply combinatorial auction. This generalizes the digital good auction and exemplifies the problem of discriminatory pricing through the sale of multiple products. The setting here is the following. We have m different items, each in unlimited supply (like a supermarket), and bidders have valuations over *subsets* of items. Our goal is to achieve revenue nearly as large as the best revenue that uses take-it-or-leave-it prices for each item individually, i.e., the best *item-pricing*.

For arbitrary item pricings we show that our reduction has a convergence rate of $\tilde{\Omega}\left(\frac{hm^2}{\epsilon^2}\right)$ no matter how complicated those bidders' valuations are (where the $\tilde{\Omega}$ hides terms logarithmic in n, the number of agents; m, the number of items; and h, the highest valuation). If instead the specification of the problem constrains the item prices to be integral (e.g., in pennies) or the consumers to be *unit-demand* (desiring only one of several items) or *single-minded* (desiring only a particular bundle of items) then our bound improves to $\tilde{\Omega}\left(\frac{hm}{\epsilon^2}\right)$. This improves on the bounds given by [119] for the unit-demand case by roughly a factor of m.

A special case of this setting is the problem of auctioning the right to traverse paths in a network. When the network is a tree and each user wants to reach the root (like drivers commuting into a city or a multicast tree in the Internet), Guruswami et al. [129] give an exact algorithm for the algorithmic problem to which our reduction applies as noted above.

Related Work: Several papers [60, 65] have applied machine learning techniques to mechanism design in the context of maximizing revenue in online auctions. The online setting is more difficult than the "batch" setting we consider, but the flip-side is that as a result, that work only applies to quite simple mechanism design settings where the class $\mathcal{G}$ of allowable offers has small size and can be easily listed. Also, in a similar spirit to the goals of our work, Awerbuch et al. [22] give reductions from online mechanism design to online optimization for a broad class of revenue maximization problems. Their work compares performance to the sum of bidders' valuations, a quite demanding measure. As a result, however, their approximation factors are necessarily logarithmic rather than $(1+\epsilon)$ as in our results.

Structure of this chapter: The structure of the chapter is as follows. We describe the general setting in which our results apply in Section 7.2 and give our generic reduction and bounds Section 7.3. We then apply our techniques to the digital good auction problem (Section 7.4), attribute auction problems (Section 7.5), the problem of item-pricing in combinatorial auctions (Section 7.6). We present our conclusions

in Section 7.7.

7.2 Model, Notation, and Definitions

7.2.1 Abstract Model

We assume a set $S = \{1, \ldots, n\}$ of agents. At the heart of our approach to mechanism design is the idea that the interaction between a mechanism and an agent results from the combination of an agent's *preference* with an *offer* made by the mechanism. The precise notion of what preferences and offers *are* will depend on the setting and is defined in Section 7.2.2. However, fixing the preference of agent i and an offer g we let $g(i)$ represent the payment made to the mechanism when agent i's preference is applied to the offer g. Essentially, we are letting the structure of an agent's preference and the structure of the offer be represented solely by $g(i)$. We extend our notation to allow $g(S)$ to be the total profit when offering g to all agents in S, and we assume that $g(S) = \sum_{i \in S} g(i)$. This effectively corresponds to an unlimited-supply assumption in the auction setting.

In our setting we have a class $\mathcal{G}$ of allowable offers. Our problem will be to find offers in $\mathcal{G}$ to make to the agents to maximize our profit. For this abstract setting we propose an algorithmic optimization problem and a mechanism design problem, the difference being that in the former we constrain the algorithm to make the same offer to all agents, and in the latter the mechanism is constrained by lack of prior knowledge of the agents' true preferences and must be *incentive compatible*.

Given the *true* preferences of S and a class of offers $\mathcal{G}$, the *algorithmic optimization problem* is to find the $g \in \mathcal{G}$ with maximum profit, i.e., $\mathrm{opt}_{\mathcal{G}}(S) = \mathrm{argmax}_{g \in \mathcal{G}}\, g(S)$. Let $\mathrm{OPT}_{\mathcal{G}}(S) = \max_{g \in \mathcal{G}} g(S)$ be this maximum profit. This computational problem is interesting in its own right, especially when the structure of agent preferences and the allowable offers results in a concise formula for $g(i)$ for all $g \in \mathcal{G}$ and all $i \in S$. All of the techniques we develop assume that such an algorithm (or an approximation to it) exists, and some require existence of an algorithm that optimizes over the profit of an offer minus some penalty term that is related to the complexity of the offer, i.e., $\max_{g \in \mathcal{G}} \left[g(S) - \mathsf{pen}_g(S) \right]$.

We now define an abstract mechanism-design-like problem that is modelled after the standard characterization of single-round sealed-bid direct-revelation incentive-compatible mechanisms (see below). For the class of offers $\mathcal{G}$, each agent has a payoff profile which lists the payment they would make for each possible offer, i.e., $[g(i)]_{g \in \mathcal{G}}$ for agent i (notice that this represents all of the relevant information in agent i's preference). Our abstract mechanism chooses an offer g_i for each agent i in a way that is independent of that agent's payoff profile, but can be a function of the agent's identity and the payoff profiles of other agents. That is, for some function f, $g_i = f(i, [g(j)]_{g \in \mathcal{G}, j \neq i})$. The mechanism then selects the outcome for agent i determined by their preference and g_i, which nets a profit of $g_i(i)$. The total profit of such a mechanism is $\sum_i g_i(i)$. We define an abstract deterministic mechanism to be completely specified by such a function f and an abstract randomized mechanism is a randomization over abstract deterministic mechanisms. The main design problem considered in our work is to come up with a mechanism (e.g., an f or randomization over functions f) to maximize our (expected) profit.

Our approach is through a reduction from the mechanism design problem to the algorithm design problem that is applicable at this level of generality (both design and analysis), though tighter analysis is possible when we expose more structure in the agent preferences and class of offers (as described next). Our bounds make use of a parameter h which upper bounds on the value of $g(i)$ for all $i \in S$ and $g \in \mathcal{G}$; that is, no individual agent can influence the total profit by more than h. The auctions we describe that make use of the technique of structural risk minimization will need to know h in advance.

7.2.2 Offers, Preferences, and Incentives

To describe how the framework above allows us to consider a large class of mechanism design problems, we formally discuss the details of offers, agent preferences, and the constraints imposed by incentive compatibility. To do this we develop some notation; however, the main results in our work will be given using the general framework above.

Formally, a *market* consists of a set of n agents, S, and a space of possible outcomes, $\mathcal{O}$. We consider *unlimited supply* allocation problems where $\mathcal{O}_i$ is set of possible outcomes (allocations) to agent i and $\mathcal{O} = \mathcal{O}_1 \times \cdots \times \mathcal{O}_n$ (i.e., all possible combinations of allocations are feasible). Except where noted, we assume there is no cost to the mechanism for producing any outcome.

As is standard in the mechanism design literature [177], an agent i's preference is fully specified by its private type, which we denote v_i. We assume *no externalities*, which means that v_i can be viewed as a preference ordering, $\succeq_{v_i}$, over (outcome, payment) pairs in $\mathcal{O}_i \times \mathcal{R}$. That is, each agent cares only about what outcome it receives and pays, and not about what other agents get. A *bid*, b_i, is a reporting of one's type, i.e., it is also a preference ordering over (outcome, payment) pairs, and we say a bidder is bidding truthfully if the preference ordering under b_i matches that given by its true type, v_i.

A deterministic mechanism is *incentive compatible* if for all agents i and all actions of the other agents, bidding truthfully is at least as good as bidding non-truthfully. If $o_i(b_i, \mathbf{b}_{-i})$ and $p_i(b_i, \mathbf{b}_{-i})$ are the outcome and payment when agent i bids b_i and the other agents bid $\mathbf{b}_{-i}$, then incentive compatibility requires for all v_i, b_i, and $\mathbf{b}_{-i}$,

$$(o_i(v_i, \mathbf{b}_{-i}), p_i(v_i, \mathbf{b}_{-i})) \succeq_{v_i} (o_i(b_i, \mathbf{b}_{-i}), p_i(b_i, \mathbf{b}_{-i})).$$

A randomized mechanism is incentive compatible if it is a randomization over deterministic incentive compatible mechanisms.

An offer, as described abstractly in the preceding section, need not be *anonymous*. This allows the freedom to charge different agents different prices for the same outcome. In particular, for a fixed offer g, the payment to two agents, $g(i)$ and $g(i')$, may be different even if $b_i = b_{i'}$. We consider a structured approach to this sort of discriminatory pricing by associating to each agent i some publicly observable *attribute* value pub_i. An *offer* then is a mapping from a bidder's public information to a collection of (outcome, payment) pairs which the agent's preference ranks. We interpret making an offer to an agent as choosing the outcome and payment that they most prefer according to their reported preference. For an incentive compatible mechanism, where we can assume that $v_i = b_i$, $g(i)$ is the payment component of this (outcome, payment) pair. Clearly, the mechanism that always makes every agent a fixed offer is by definition incentive-compatible. In fact the following more general result, which motivates the above definition of an abstract mechanism, is easy to show:

Fact 7.2.1 *A mechanism is* incentive compatible *if the choice of which offer to make to any agent does not depend on the agent's reported preference.*

Because all our mechanisms are incentive compatible, the established notation of $g(i)$ as the profit of offer g on agent i will be sufficient for most discussions and we will omit explicit reference to v_i and b_i where possible.

7.2.3 Quasi-linear Preferences

We will apply our general framework and analysis to a number of special cases where the agents' preferences are to maximize their *quasi-linear utility*. This is the most studied case in mechanism design literature. The type, v_i, of a quasi-linear utility maximizing agent i specifies its *valuation* for each outcome. We denote the valuation of agent i for outcome $o_i \in \mathcal{O}_i$ as $v_i(o_i)$. This agent's *utility* is the

difference between its valuation and the price it is required to pay. I.e., for outcome o_i and payment p_i, agent i's utility is $u_i = v_i(o_i) - p_i$. An agent prefers the outcome and payment that maximizes its utility. I.e., $v_i(o_i) - p_i \geq v_i(o_i') - p_i'$ if and only if $(o_i, p_i) \succeq_{v_i} (o_i', p_i')$.

For the quasi-linear case, the incentive compatibility constraints imply for all v_i, b_i, and $\mathbf{b}_{-i}$ that,

$$v_i(o_i(v_i, \mathbf{b}_{-i})) - p_i(v_i, \mathbf{b}_{-i}) \geq v_i(o_i(b_i, \mathbf{b}_{-i})) - p_i(b_i, \mathbf{b}_{-i}).$$

Notice that in the quasi-linear setting our constraint that $g(i) \leq h$ would be implied by the condition that $v_i(o_i) \leq h$ for all $o_i \in \mathcal{O}_i$.

7.2.4 Examples

The following examples illustrate the relationship between the outcome of the mechanism, offers, valuations, and attributes. (The first three examples are quasi-linear, the fourth is not.)

Digital Good Auction: The digital good auction models an auction of a single item in unlimited supply to indistinguishable bidders. Here the set of possible outcomes for bidder i is $\mathcal{O}_i = \{0, 1\}$ where $o_i = 1$ represents bidder i receiving a copy of the good and $o_i = 0$ otherwise. We normalize their valuation function $v_i(0) = 0$ and use a simple shorthand notation of $v_i = v_i(1)$ as the bidders privately known valuation for receiving the good. As described in the introduction, in this setting the bidders have no public information. Here, a natural class of offers, $\mathcal{G}$, is the class of all take-it-or-leave-it prices. For bidder i with valuation v_i and offer $g_p =$ "take the good for \$$p$, or leave it" the profit is

$$g_p(i) = \begin{cases} p & \text{if } p \leq v_i \\ 0 & \text{otherwise.} \end{cases}$$

We consider the digital good auction problem in detail in Section 7.4.

Attribute Auctions: This is the same as the digital good setting except now each bidder i is associated a public attribute, $\textit{pub}_i \in \mathcal{X}$, where $\mathcal{X}$ is the *attribute space*. We view $\mathcal{X}$ as an abstract space, but one can envision it as $\mathcal{R}^d$, for example. Let $\mathcal{P}$ be a class of pricing functions from $\mathcal{X}$ to $\mathcal{R}_+$, such as all linear functions, or all functions that partition $\mathcal{X}$ into k markets in some natural way (say, based on distance to k cluster centers) and offer a different price in each. Let $\mathcal{G}$ be the class of take-it-or-leave-it offers induced by $\mathcal{P}$. That is, if $p \in \mathcal{P}$ is a pricing function, then the offer $g_p \in \mathcal{G}$ induced by p is: "for bidder i, take the good for \$$p(\textit{pub}_i)$, or leave it". The profit to the mechanism from bidder i with valuation v_i and public information $\textit{pub}_i$ is

$$g_p(i) = \begin{cases} p(\textit{pub}_i) & \text{if } p(\textit{pub}_i) \leq v_i, \\ 0 & \text{otherwise.} \end{cases}$$

We will give analyses for several interesting classes of pricing functions in Section 7.5.

Combinatorial Auctions: Here we have a set J of m distinct items, each in unlimited supply. Each consumer has a private valuation $v_i(J')$ for each bundle $J' \subseteq J$ of items, which measures how much receiving bundle J' would be worth to the consumer i (again we normalize such that $v_i(\emptyset) = 0$). For simplicity, we assume bidders are indistinguishable, i.e., there is no public information. A natural class of offers $\mathcal{G}$ (studied in [129]) is the class of functions that assign a separate price to each item, such that the price of a bundle is just the sum of the prices of the items in it (called item pricing). For price vector $\mathbf{p} = (p_1, \ldots, p_m)$ let the offer $g_{\mathbf{p}} =$ "for bundle J', pay $\sum_{j \in J'} p_j$". The profit for

bidder i on offer $g_{\mathbf{p}}$ is

$$g_{\mathbf{p}}(i) = \sum \left\{ p_j \,:\, j \in \operatorname{argmax}_{J' \subset J} \left[v_i(J') - \sum\nolimits_{j' \in J'} p_{j'} \right] \right\}.$$

(If the bundle J' maximizing the bidder's utility is not unique, we define the mechanism to select the utility-maximizing bundle of greatest profit.) We discuss combinatorial auctions in Section 7.6.

Marginal Cost Auctions with Budgets: To illustrate an interesting model with agents in a non-quasi-linear setting consider the case each bidder i's preference is given tuple (B_i, v_i) where B_i is their budget and v_i is their value-per-unit received. Possible allocations for bidder i, $\mathcal{O}_i$, are non-negative real numbers corresponding to the number of units they receive. Assuming their total payment is less than their budget, bidder i's utility is simply $v_i o_i$ minus their payment; a bidder's utility when payments exceed their budget is negative infinity.

We assume that the seller has a fixed marginal cost c for producing a unit of the good. Consider the class of offers $\mathcal{G}$ with $g_p =$ "pay $\$p$ per unit received". A bidder i faced with offer g_p with $p < v_i$ will maximize their utility by buying enough units to exactly exhaust their budget. The payoff to the auctioneer for this bidder i is therefor B_i less c times the number of units the bidder demands. I.e.,

$$g_p(i) = \begin{cases} B_i - cB_i/p & \text{if } p \leq v_i, \\ 0 & \text{otherwise.} \end{cases}$$

This model is quite similar to one considered by Borgs et al. [70]. Though we do not explicitly analyze this setting, it is simple to apply our generic analysis to get reasonable bounds.

7.3 Generic Reductions

We are interested in reducing incentive-compatible mechanism design to the (non-incentive-compatible) algorithmic optimization problem. Our reductions will be based on random sampling. Let $\mathcal{A}$ be an algorithm (exact or approximate) for the algorithmic optimization problem over $\mathcal{G}$. The simplest mechanism that we consider, which we call $\text{RSO}_{(\mathcal{G},\mathcal{A})}$ (Random Sampling Optimal offer), is the following generalization of the random sampling digital-goods auction from [121]:

0. Bidders commit to their preferences by submitting their bids.
1. Randomly split the bidders into two groups S_1 and S_2 by flipping a fair coin for each bidder to determine its group.
2. Run $\mathcal{A}$ to determine the best (or approximately best) offer $g_1 \in \mathcal{G}$ over S_1, and similarly the best (or approximately best) $g_2 \in \mathcal{G}$ over S_2.
3. Finally, apply g_1 to all bidders in S_2 and g_2 to all bidders in S_1 using their reported bids.

We will also consider various more refined versions of $\text{RSO}_{(\mathcal{G},\mathcal{A})}$ that discretize $\mathcal{G}$ or perform some type of *structural risk minimization* (in which case we will need to assume $\mathcal{A}$ can optimize over the modifications made to $\mathcal{G}$).

Note 1: One might think that the "leave-one-out" mechanism, where the offer made to a given bidder i is the best offer for all other bidders, i.e., $\text{opt}_{\mathcal{G}}(S \setminus \{i\})$, would be a better mechanism than the random sampling mechanism above. However, as pointed out in [121, 124], such a mechanism (and indeed, any symmetric deterministic mechanism) has poor worst-case revenue. Furthermore, even if bidders' valuations are independently drawn from some distribution, the leave-one-out revenue can be much less

stable than $\text{RSO}_{(\mathcal{G},\mathcal{A})}$ in that it may have a non-negligable probability of achieving revenue that is far from optimal, whereas such an event is exponentially small for $\text{RSO}_{(\mathcal{G},\mathcal{A})}$.[1]

Note 2: The reader will notice that in converting an algorithm for finding the best offer in $\mathcal{G}$ into an incentive-compatible mechanism, we produce a mechanism whose outcome is not simply that of a single offer applied to all consumers. For example, even in the simplest case of auctioning a digital good to indistinguishable bidders, we compare our performance to the best take-it-or-leave-it price, and yet the auction itself does not in fact offer each bidder the same price (all bidders in S_1 get the same price, and all bidders in S_2 get the same price, but those two prices may be different). In fact, Goldberg and Hartline [120] show that this sort of behavior is necessary: it is not possible for an incentive-compatible auction to approximately maximize profit and offer all the bidders the same price.

7.3.1 Generic Analyses

The following theorem shows that the random sampling auction incurs only a small loss in performance if the profit of the optimal offer is large in comparison to the logarithm of the number of offers we are choosing from. Later sections of this chapter will focus on techniques for bounding the effective size (or complexity) of $\mathcal{G}$ that can yield even stronger guarantees.

Theorem 7.3.1 *Given the offer class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$, then with probability at least $1-\delta$ the profit of $\text{RSO}_{(\mathcal{G},\mathcal{A})}$ is at least $(1-\epsilon)\text{OPT}_{\mathcal{G}}/\beta$ as long as*

$$\text{OPT}_{\mathcal{G}} \geq \beta \frac{18h}{\epsilon^2} \ln\left(\frac{2|\mathcal{G}|}{\delta}\right).$$

Notice that this bound holds for all ϵ and δ simultaniously as these are not parameters of the mechanism. In particular, this bound and those given by the two immediate corollaries, below, show how the approximation factor improves as a function of market size.

Corollary 7.3.2 *Given the offer class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$, then with probability at least $1-\delta$, the profit of $\text{RSO}_{(\mathcal{G},\mathcal{A})}$ is at least $(1-\epsilon)\text{OPT}_{\mathcal{G}}/\beta$, when $\text{OPT}_{\mathcal{G}} \geq n$ and the number of bidders n satisfies*

$$n \geq \frac{18h\beta}{\epsilon^2} \ln\left(\frac{2|\mathcal{G}|}{\delta}\right).$$

Corollary 7.3.3 *Given the offer class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$ then with probability at least $1-\delta$, the profit of $\text{RSO}_{(\mathcal{G},\mathcal{A})}$ is at least*

$$(1-\epsilon)\text{OPT}_{\mathcal{G}}/\beta - \frac{18h\beta}{\epsilon^2} \ln\left(\frac{2|\mathcal{G}|}{\delta}\right).$$

If bidders' valuations are in the interval $[1,h]$ and the take-it-or-leave-it offer of \$1 is in $\mathcal{G}$, then the condition $\text{OPT}_{\mathcal{G}} \geq n$ is trivially satisfied and Corollary 7.3.2 can be interpreted as giving a bound on the *convergence rate* of the random sampling auction. Corollary 7.3.3 is a useful form of our bound when considering structural risk minimization and it also matches the form of bounds given in prior work (e.g., [60]).

For example, in the digital good auction with the class of offers $\mathcal{G}_\epsilon$ consisting of all take-it-or-leave-it offers in the interval $[1,h]$ discretized to powers of $1+\epsilon$, we have $\text{OPT}_{\mathcal{G}_\epsilon} \geq n$ (since each bidder's

[1]For example, say we are selling just one item and the distribution over valuations is 50% probability of valuation 1 and 50% probability of valuation 2. If we have n bidders, then there is a nontrivial chance (about $1/\sqrt{n}$) that there will be the exact same number of each type ($n/2$ bidders with valuation 1 and $n/2$ bidders with valuation 2), and the mechanism will make the wrong decision on everybody. The $\text{RSO}_{(\mathcal{G},\mathcal{A})}$ mechanism on the other hand has only an exponentially small probability of doing this poorly.

valuation is at least 1), $\beta = 1$ (since the algorithmic problem is easy), and $|\mathcal{G}_\epsilon| = \lceil \log_{1+\epsilon} h \rceil$. So, Corollary 7.3.2 states that $O(\frac{h}{\epsilon^2} \log \log_{1+\epsilon} h)$ bidders are sufficient to perform nearly as well as optimal (we derive better bounds for this problem in Section 7.4).

In general we will give our bounds in a similar form as Theorem 7.3.1, knowing that bounds of the form of Corollary 7.3.2 and 7.3.3 can be easily derived. The only exceptions are the structural risk minimization results which we give in the same form as Corollary 7.3.3.

In the remainder of this section we prove Theorem 7.3.1. We start with a lemma that is key to our analysis.

Lemma 7.3.4 *Given S, an offer g satisfying $0 \leq g(i) \leq h$ for all $i \in S$, and a profit level p, if we randomly partition S into S_1 and S_2, then the probability that $|g(S_1) - g(S_2)| \geq \epsilon \max[g(S), p]$ is at most $2e^{-\frac{\epsilon^2 p}{2h}}$.*

Proof: Let $Y_1, \ldots, Y_n$ be i.i.d. random variables that define the partition of S into S_1 and S_2: that is, Y_i is 1 with probability $\frac{1}{2}$ and Y_i is 2 with probability $\frac{1}{2}$. Let $t(Y_1, ..., Y_n) = \sum_{i:Y_i=1} g(i)$. So, as a random variable, $g(S_1) = t(Y_1, ..., Y_n)$ and clearly $\mathbf{E}[t(Y_1, ..., Y_n)] = \frac{g(S)}{2}$. Assume first that $g(S) \geq p$. From the McDiarmid concentration inequality (see Theorem A.3.1 in Appendix A.3), by plugging in $c_i = g(i)$, we get:

$$\Pr\left\{\left|g(S_1) - \frac{g(S)}{2}\right| \geq \frac{\epsilon}{2} g(S)\right\} \leq 2e^{-\frac{1}{2}\epsilon^2 g(S)^2 / \sum_{i=1}^{n} g(i)^2}.$$

Since

$$\sum_{i=1}^{n} g(i)^2 \leq \max_i\{g(i)\} \sum_{i=1}^{n} g(i) \leq h g(S),$$

we obtain:

$$\Pr\left\{\left|g(S_1) - \frac{g(S)}{2}\right| \geq \frac{\epsilon}{2} g(S)\right\} \leq 2e^{-\frac{\epsilon^2 g(S)}{2h}}.$$

Moreover, since $g(S_1) + g(S_2) = g(S)$ and $g(S) \geq p$, we obtain:

$$\Pr\{|g(S_1) - g(S_2)| \geq \epsilon g(S)\} \leq 2e^{-\epsilon^2 p/(2h)},$$

as desired. Consider now the case that $g(S) < p$. Again, using the McDiarmid inequality we have

$$\Pr\{|g(S_1) - g(S_2)| \geq \epsilon p\} \leq 2e^{-\frac{1}{2}\epsilon^2 p^2 / \sum_{i=1}^{n} g(i)^2}.$$

Since $\sum_{i=1}^{n} g(i)^2 \leq h g(S) \leq ph$ we obtain again that

$$\Pr\{|g(S_1) - g(S_2)| \geq \epsilon p\} \leq 2e^{-\frac{\epsilon^2 p}{2h}},$$

which gives us the desired bound. ■

It is worth noting that using tail inequalities that depend on the maximum range of the random variables rather than the sum of their squares in the proof of Lemma 7.3.4 would increase the h to an h^2 in the exponent. Note also that if $g(i) = g'(i)$ for all $i \in S$ then they are equivalent from the point of view of the auction; we will use $|\mathcal{G}|$ to denote the number of *different* such offers in $\mathcal{G}$.[2] Lemma 7.3.4 implies that:

[2]Notice that in our generic reduction, $|\mathcal{G}|$ only appears in the analysis and we do not actually have to know whether two offers are equivalent with respect to S when running the auction.

Corollary 7.3.5 *For a random partition of S into S_1 and S_2, with probability at least $1-\delta$, all offers g in $\mathcal{G}$ such that $g(S) \geq \frac{2h}{\epsilon^2}\ln\left(\frac{2|\mathcal{G}|}{\delta}\right)$ satisfy $|g(S_1)-g(S_2)| \leq \epsilon g(S)$.*

Proof: Follows from Lemma 7.3.4 by plugging in $p = \frac{2h}{\epsilon^2}\ln\left(\frac{2|\mathcal{G}|}{\delta}\right)$ and then using the union bound over all $g \in \mathcal{G}$. ■

We complete this section with the proof of the main theorem.

Proof of Theorem 7.3.1: Let g_1 be the offer in $\mathcal{G}$ produced by $\mathcal{A}$ over S_1 and g_2 be the offer in $\mathcal{G}$ produced by $\mathcal{A}$ over S_2. Let g_{OPT} be the optimal offer in $\mathcal{G}$ over S; so $g_{\mathrm{OPT}}(S) = \mathrm{OPT}_{\mathcal{G}}$. Since the optimal offer over S_1 is at least as good as g_{OPT} on S_1 (and likewise for S_2), the fact that $\mathcal{A}$ is a β-approximation implies that $g_1(S_1) \geq \frac{g_{\mathrm{OPT}}(S_1)}{\beta}$ and $g_2(S_2) \geq \frac{g_{\mathrm{OPT}}(S_2)}{\beta}$.

Let $p = \frac{18h}{\epsilon^2}\ln\left(\frac{2|\mathcal{G}|}{\delta}\right)$. Using Lemma 7.3.4 (applying the union bound over all $g \in \mathcal{G}$), we have that with probability $1-\delta$, every $g \in \mathcal{G}$ satisfies $|g(S_1)-g(S_2)| \leq \frac{\epsilon}{3}\max[g(S),p]$. In particular, $g_1(S_2) \geq g_1(S_1) - \frac{\epsilon}{3}\max[g_1(S),p]$, and $g_2(S_1) \geq g_2(S_2) - \frac{\epsilon}{3}\max[g_2(S),p]$.

Since the theorem assumes that $\mathrm{OPT}_{\mathcal{G}} \geq \beta p$, summing the above two inequalities and performing a case analysis[3] we get that the profit of $\mathrm{RSO}_{(\mathcal{G},\mathcal{A})}$, namely the sum $g_1(S_2)+g_2(S_1)$, is at least $(1-\epsilon)\frac{\mathrm{OPT}_{\mathcal{G}}}{\beta}$. More specifically, assume first that $g_1(S) \geq p$ and $g_2(S) \geq p$. This implies that

$$g_1(S_2) \geq g_1(S_1) - \frac{\epsilon}{3}g_1(S) \quad \text{and} \quad g_2(S_1) \geq g_2(S_2) - \frac{\epsilon}{3}g_2(S),$$

and therefore

$$(1+\frac{\epsilon}{3})g_1(S_2) \geq (1-\frac{\epsilon}{3})g_1(S_1) \quad \text{and} \quad (1+\frac{\epsilon}{3})g_2(S_1) \geq (1-\frac{\epsilon}{3})g_2(S_2).$$

So, the profit of $\mathrm{RSO}_{(\mathcal{G},\mathcal{A})}$ in this case is at least

$$\frac{1-\frac{\epsilon}{3}}{1+\frac{\epsilon}{3}}(g_1(S_1)+g_2(S_2)) \geq \frac{1-\frac{\epsilon}{3}}{1+\frac{\epsilon}{3}}\frac{\mathrm{OPT}_{\mathcal{G}}}{\beta} \geq (1-\epsilon)\frac{\mathrm{OPT}_{\mathcal{G}}}{\beta}.$$

If both $g_1(S) < p$ and $g_2(S) < p$, then $g_1(S_2) \geq g_1(S_1) - \frac{\epsilon}{3}p$ and $g_2(S_1) \geq g_2(S_2) - \frac{\epsilon}{3}p$, and so the profit of $\mathrm{RSO}_{(\mathcal{G},\mathcal{A})}$ in this case is at least $\frac{\mathrm{OPT}_{\mathcal{G}}}{\beta} - \frac{2\epsilon}{3}p$ which is at least $(1-\epsilon)\frac{\mathrm{OPT}_{\mathcal{G}}}{\beta}$ by our assumption that $\mathrm{OPT}_{\mathcal{G}} \geq \beta p$.

Finally, assume without loss of generality that $g_1(S) \geq p$ and $g_2(S) < p$. This implies that

$$g_1(S_2) \geq g_1(S_1) - \frac{\epsilon}{3}g_1(S) \quad \text{and} \quad g_2(S_1) \geq g_2(S_2) - \frac{\epsilon}{3}p.$$

The former inequality implies that $(1+\frac{\epsilon}{3})g_1(S_2) \geq (1-\frac{\epsilon}{3})g_1(S_1)$, and so $g_1(S_2) \geq \left(1-\frac{2\epsilon}{3}\right)g_1(S_1)$, and the latter inequality implies that $g_2(S_1) \geq g_2(S_2) - \frac{\epsilon}{3}\frac{\mathrm{OPT}_{\mathcal{G}}}{\beta}$. Together we have that

$$g_1(S_2)+g_2(S_1) \geq \left(1-\frac{2\epsilon}{3}\right)\frac{g_{\mathrm{OPT}}(S_1)}{\beta} + \frac{g_{\mathrm{OPT}}(S_2)}{\beta} - \frac{\epsilon}{3}\frac{\mathrm{OPT}_{\mathcal{G}}}{\beta} \geq (1-\epsilon)\frac{\mathrm{OPT}_{\mathcal{G}}}{\beta},$$

as desired. ■

[3]Note that if $\beta = 1$, then the conclusion follows easily. The case analysis is only need to deal with the case $\beta > 1$.

7.3.2 Structural Risk Minimization

In many natural cases, $\mathcal{G}$ consists of offers at different "levels of complexity" k. In the case of attribute auctions, for instance, $\mathcal{G}$ could be an offer class induced by pricing functions that partition bidders into k markets and offer a constant price in each market, for different values of k. The larger k is the more complex the offer is. One natural approach to such a setting is to perform *structural risk minimization* (SRM): that is, to assign a penalty term to offers based on their complexity and then to run a version of $\text{RSO}_{(\mathcal{G},\mathcal{A})}$ in which $\mathcal{A}$ optimizes profit minus penalty. Specifically, let $\bar{\mathcal{G}}$ be a series of offers classes $\mathcal{G}_1, \mathcal{G}_2, \ldots$, and let pen be a penalty function defined over these classes. We then define the procedure $\text{RSO-SRM}_{(\bar{\mathcal{G}},\mathsf{pen})}$ as follows:

1. Randomly partition the bidders into two sets, S_1 and S_2, by flipping fair coin for each bidder.
2. Compute g_1 to maximize $\max_k \max_{g\in\mathcal{G}_k} [g(S_1) - \mathsf{pen}(\mathcal{G}_k)]$ and similarly compute g_2 from S_2.
3. Use the offer g_1 for bidders in S_2 and the offer g_2 for bidders in S_1.

We can now derive a guarantee for the $\text{RSO-SRM}_{(\bar{\mathcal{G}},\mathsf{pen})}$ mechanism as follows:

Theorem 7.3.6 *Assuming that we have an algorithm for solving the optimization problem required by* $\textit{RSO-SRM}_{(\bar{\mathcal{G}},\mathsf{pen})}$*, then for any given value of* n, ϵ*, and* δ*, with probability at least* $1-\delta$*, the revenue of* $\textit{RSO-SRM}_{(\bar{\mathcal{G}},\mathsf{pen})}$ *for* $\mathsf{pen}(\mathcal{G}_k) = \frac{8h_k}{\epsilon^2}\ln\left(\frac{8k^2|\mathcal{G}_k|}{\delta}\right)$ *is at least*

$$\max_k \left([(1-\epsilon)\,\mathrm{OPT}_k - 2\mathsf{pen}(\mathcal{G}_k)]\right),$$

where h_k *is the maximum payoff from* $\mathcal{G}_k$ *and* $\mathrm{OPT}_k = \mathrm{OPT}_{\mathcal{G}_k}$.

Proof: Using Corollary 7.3.5 and a union bound over the values $\delta_k = \delta/(4k^2)$, we obtain that with probability at least $1-\delta$, simultaneously for all k and for all offers g in $\mathcal{G}_k$ such that $g(S) \geq \frac{8h_k}{\epsilon^2}\ln(8k^2|\mathcal{G}_k|/\delta) = \mathsf{pen}(\mathcal{G}_k)$, we have $|g(S_1) - g(S_2)| \leq \frac{\epsilon}{2}g(S)$. Let k^* be the optimal index, namely let k^* be the index such that

$$(1-\epsilon)\,\mathrm{OPT}_{k^*} - 2\mathsf{pen}(\mathcal{G}_{k^*}) = \max_k \left((1-\epsilon)\,\mathrm{OPT}_k - 2\mathsf{pen}(\mathcal{G}_k)\right),$$

and let k_i be the index of the best offer (according to our criterion) over S_i, for $i = 1, 2$. By our assumption that g_1 and g_2 were chosen by an optimal algorithm, we have

$$g_i(S_i) - \mathsf{pen}(\mathcal{G}_{k_i}) \geq g_{\mathrm{OPT}_{k^*}}(S_i) - \mathsf{pen}(\mathcal{G}_{k^*}), \quad \text{for} \quad i = 1, 2.$$

We will argue next that $g_1(S_2) \geq \frac{1-\frac{\epsilon}{2}}{1+\frac{\epsilon}{2}}\left(g_{\mathrm{OPT}_{k^*}}(S_1) - \mathsf{pen}(\mathcal{G}_{k^*})\right)$. First, if $g_1(S_1) < \mathsf{pen}(\mathcal{G}_{k_1})$, then the conclusion is clear since we have

$$0 > g_1(S_1) - \mathsf{pen}(\mathcal{G}_{k_1}) \geq g_{\mathrm{OPT}_{k^*}}(S_1) - \mathsf{pen}(\mathcal{G}_{k^*}).$$

If $g_1(S_1) \geq \mathsf{pen}(\mathcal{G}_{k_1})$, then as argued above we have $|g_1(S_1) - g_1(S_2)| \leq \frac{\epsilon}{2}g_1(S)$ and so

$$g_1(S_2) \geq \frac{1-\frac{\epsilon}{2}}{1+\frac{\epsilon}{2}}g_1(S_1) \geq \frac{1-\frac{\epsilon}{2}}{1+\frac{\epsilon}{2}}\left(g_{\mathrm{OPT}_{k^*}}(S_1) - \mathsf{pen}(\mathcal{G}_{k^*})\right).$$

Similarly, we can prove that we have $g_2(S_1) \geq \frac{1-\frac{\epsilon}{2}}{1+\frac{\epsilon}{2}}\left(g_{\mathrm{OPT}_{k^*}}(S_2) - \mathsf{pen}(\mathcal{G}_{k^*})\right)$. All these together imply that the profit of the mechanism $\text{RSO-SRM}_{(\bar{\mathcal{G}},\mathsf{pen})}$, namely $g_1(S_2) + g_2(S_1)$, is at least

$$\frac{1-\frac{\epsilon}{2}}{1+\frac{\epsilon}{2}}\left(g_{\mathrm{OPT}_{k^*}}(S) - 2\mathsf{pen}(\mathcal{G}_{k^*})\right) \geq \left((1-\epsilon)\,\mathrm{OPT}_{k^*} - 2\mathsf{pen}(\mathcal{G}_{k^*})\right),$$

as desired. ■

7.3.3 Improving the Bounds

The results above say, in essence, that if we have enough bidders so that the optimal profit is large compared to $\frac{h}{\epsilon^2}\log(|\mathcal{G}|)$, then our mechanism will perform nearly as well as the best offer in $\mathcal{G}$. In these bounds, one should think of $\log(|\mathcal{G}|)$ as a measure of the complexity of the offer class $\mathcal{G}$; for instance, it can be thought of as the number of bits needed to describe a typical offer in that class. However, in many cases one can achieve a better bound by adapting techniques developed for analyzing generalization performance in machine learning theory. In this section, we discuss a number of such methods that can produce better bounds. These include both *analysis* techniques (such as using appropriate forms of *covering numbers*), where we do not change the mechanism but instead provide a stronger guarantee, and *design* techniques (like *discretizing*), where we modify the mechanism to produce a better bound.

Discretizing

Notation: Given a class of offers $\mathcal{G}$, define $\mathcal{G}_\alpha$ to be the set of offers induced by rounding all prices down to the nearest power of $(1+\alpha)$.

In many cases, we can greatly reduce $|\mathcal{G}|$ without much affecting $\mathrm{OPT}_\mathcal{G}$ by performing some type of discretization. For instance, for auctioning a digital good, there are infinitely many offers induced by all take-it-or-leave-it prices but only $\log_{1+\alpha} h \approx \frac{1}{\alpha}\ln h$ offers induced by the discretized prices at powers of $1+\alpha$. Also, since rounding down the optimal price to the nearest power of $1+\alpha$ can reduce revenue for this auction by at most a factor of $1+\alpha$, the optimal offer in the discretized class must be close, in terms of total profit, to the optimal offer in the original class. More generally, if we can find a smaller offer class $\mathcal{G}'$ such that $\mathrm{OPT}_{\mathcal{G}'}$ is guaranteed to be close to $\mathrm{OPT}_\mathcal{G}$, then we can instruct our algorithm $\mathcal{A}$ to optimize over $\mathcal{G}'$ instead of $\mathcal{G}$ to get better bounds. We consider the discretization $\mathcal{G}_\alpha$ in our refined analysis of the digital good auction problem (Section 7.4) and in our consideration of attribute auctions (Section 7.5). Further, in Section 7.6 we discuss an interesting alternative discretization for item-pricing in combinatorial auctions.

Counting Possible Outputs

Suppose we can argue that our algorithm $\mathcal{A}$, run on a subset of S, will only ever output offers from a restricted set $\mathcal{G}_\mathcal{A} \subseteq \mathcal{G}$. For example, for the problem of auctioning a digital good, if $\mathcal{A}$ picks the offer based on the optimal take-it-or-leave-it price over its input then this price must be one of the bids, so $|\mathcal{G}_\mathcal{A}| \leq n$. Then, we can simply replace $|\mathcal{G}|$ with $|\mathcal{G}_\mathcal{A}|$ (or $|\mathcal{G}_\mathcal{A}|+1$ if the optimal offer is not in $\mathcal{G}_\mathcal{A}$) in all the above arguments. Formally we can say that:

Observation 7.3.7 *If algorithm $\mathcal{A}$, run on any subset of S, only output offers from a restricted set $\mathcal{G}_\mathcal{A} \subseteq \mathcal{G}$, then all the bounds in Sections 7.3.1 and 7.3.2 hold with $|\mathcal{G}|$ replaced by $|\mathcal{G}_\mathcal{A}|+1$.*

Using Covering Numbers

The main idea of these arguments is the following. Suppose $\mathcal{G}$ has the property that there exists a much smaller class $\mathcal{G}'$ such that every $g \in \mathcal{G}$ is "close" to some $g' \in \mathcal{G}'$, with respect to the given set of bidders S. Then one can show that if all offers in $\mathcal{G}'$ perform similarly on S_1 as they do on S_2, then this will be true for all offers in $\mathcal{G}$ as well. These kind of arguments are quite often used in machine learning (see for instance [18, 72, 103, 203]), but the main challenge is to define the right notion of "close" for our mechanism design setting to get good and meaningful bounds. Specifically, we will consider L_1 multiplicative γ-covers which we define as follows:

Definition 7.3.1 *$\mathcal{G}'$ is an L_1 multiplicative γ-cover of $\mathcal{G}$ with respect to S if for every $g \in \mathcal{G}$ there exists $g' \in \mathcal{G}'$ such that*

$$\sum_{i \in S} |g(i) - g'(i)| \quad \leq \quad \gamma g(S).$$

In the following we present bounds based on L_1 multiplicative γ-covers. We start by proving the following structural lemma characterizing these L_1 covers.

Lemma 7.3.8 *If $\sum_{i \in S} |g(i) - g'(i)| \leq \gamma g(S)$ and $|g'(S_1) - g'(S_2)| \leq \epsilon' \max[g'(S), p]$ then we have*

$$|g(S_1) - g(S_2)| \leq \epsilon' \max[g'(S), p] + \gamma g(S).$$

This further implies that

$$|g(S_1) - g(S_2)| \leq \big(\gamma + \epsilon'(1+\gamma)\big) \max[g(S), p].$$

Proof: We will first prove that $g(S_1) \geq g(S_2) - \epsilon' \max[g'(S), p] - \gamma g(S)$. Note that this clearly implies

$$g(S_1) \geq g(S_2) - \big(\gamma + \epsilon'(1+\gamma)\big) \max[g(S), p],$$

since the first assumption in the lemma implies that $|g(S) - g'(S)| \leq \gamma g(S)$. Let us define

$$\vec{\Delta}_{g_1 g_2}(S) = \sum_{i \in S} \max(g_1(i) - g_2(i), 0)$$

and consider

$$\Delta_{gg'}(S) = \vec{\Delta}_{gg'}(S) + \vec{\Delta}_{g'g}(S) = \sum_{i \in S} |g(i) - g'(i)|.$$

Clearly, for any $S' \subseteq S$ we have $\vec{\Delta}_{gg'}(S) \geq \vec{\Delta}_{gg'}(S')$ and likewise $\Delta_{gg'}(S) \geq \Delta_{gg'}(S')$. Also, for any subset $S' \subseteq S$ we have $g(S') - g'(S') \leq \vec{\Delta}_{gg'}(S)$ and $g'(S') - g(S') \leq \vec{\Delta}_{g'g}(S)$. Now, from $g'(S_1) \geq g'(S_2) - \epsilon' \max[g'(S), p]$ we obtain that

$$g(S_1) + \vec{\Delta}_{g'g}(S) \geq g'(S_2) - \epsilon' \max[g'(S), p] \geq g(S_2) - \vec{\Delta}_{gg'}(S) - \epsilon' \max[g'(S), p].$$

Therefore we have

$$g(S_1) \geq g(S_2) - \Delta_{gg'}(S) - \epsilon' \max[g'(S), p],$$

which implies

$$g(S_1) \geq g(S_2) - \epsilon' \max[g'(S), p] - \gamma g(S),$$

as desired. Using the same argument with S_1 replaced by S_2 yields the theorem. ■

Using Lemma 7.3.8, we can now get the following bound:

Theorem 7.3.9 *Given the offer class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$, then with probability at least $1 - \delta$, the profit of $RSO_{(\mathcal{G},\mathcal{A})}$ is at least $(1-\epsilon)\mathrm{OPT}_{\mathcal{G}}/\beta$ so long as*

$$\mathrm{OPT}_{\mathcal{G}} \geq \beta \tfrac{72h}{\epsilon^2} \ln\Big(\tfrac{2|\mathcal{G}'|}{\delta}\Big),$$

for some L_1 multiplicative $\frac{\epsilon}{12}$-cover $\mathcal{G}'$ of $\mathcal{G}$ with respect to S.

Proof: Let $p = \frac{72h}{\epsilon^2} \ln\left(\frac{2|\mathcal{G}'|}{\delta}\right)$. By Lemma 7.3.4, applying the union bound, we have that with probability $1-\delta$, every $g' \in \mathcal{G}'$ satisfies $|g'(S_1) - g'(S_2)| \leq \frac{\epsilon}{6} \max[g'(S), p]$. Using Lemma 7.3.8, with ϵ' set to $\frac{\epsilon}{6}$ and γ set to $\frac{\epsilon}{12}$, we obtain that with probability $1-\delta$, every $g \in \mathcal{G}$ satisfies $|g(S_1) - g(S_2)| \leq \frac{\epsilon}{3} \max[g(S), p]$. Finally, proceeding as in the proof of Theorem 7.3.1 we obtain the desired result. ■

Notice that Theorem 7.3.9 implies that:

Corollary 7.3.10 *Given the offer class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$, then with probability at least $1-\delta$, the profit of RSO$_{(\mathcal{G},\mathcal{A})}$ is at least $(1-\epsilon)\mathrm{OPT}_{\mathcal{G}}/\beta$, so long as $\mathrm{OPT}_{\mathcal{G}} \geq n$ and the number of bidders satisfies*

$$n \geq \frac{72h\beta}{\epsilon^2} \ln\left(\frac{2|\mathcal{G}'|}{\delta}\right)$$

for some L_1 multiplicative $\frac{\epsilon}{12}$-cover $\mathcal{G}'$ of $\mathcal{G}$ with respect to S.

We will demonstrate the utility of L_1 multiplicative covers in Section 7.4 by showing the existence of L_1 covers of size $o(n)$ for the digital good auction. It is worth noting that a straightforward application of analogous ϵ-cover results in learning theory [18] (which would require an additive, rather than multiplicative gap of ϵ for every bidder) would add an extra factor of h into our sample-size bounds.

7.4 The Digital Good Auction

We now consider applying the results in Section 7.3 to the problem of auctioning a digital good to indistinguishable bidders. In this section we define $\mathcal{G}$ to be the natural class of offers induced by the set of all take-it-or-leave-it prices (see for instance [124]). Clearly in this case, it is trivial to solve the underlying optimization problem optimally: given a set of bidders, just output the offer induced by the constant price that maximizes the price times the number of bidders with bids at least as high as the price. Also, it is easy to see that this price will be one of the bid values. Thus, applying Theorem 7.3.7 with the bound on $|\mathcal{G}_{\mathcal{A}}| = n$, we get an approximately optimal auction with convergence rate $O(h \log n)$.

We can obtain better results using L_1 multiplicative-cover arguments and Theorem 7.3.9 as follows. Let $b_1, \ldots, b_n$ be the bids of the n bidders sorted from highest to lowest. Define $\mathcal{G}'$ as the offer class induced by $\{b_i : i = \lfloor (1+\gamma)^j \rfloor \text{ for some } j \in \mathbb{Z}\} \cup \{(1+\gamma)^i : i \in \{1, \ldots, \log_{1+\gamma} h\}\}$. Consider $g \in \mathcal{G}$ and find the $g' \in \mathcal{G}'$ that offers the largest price less than the offer price of g. Notice first that all the winners in S on g also win in g'. Second, the offer price of g' is within a factor of $1+\gamma$ of the offer price of g. Third, g' has at most a factor of $1+\gamma$ more winners than g. The first two facts above imply that $\vec{\Delta}_{gg'}(S) \leq \gamma g(S)$. The third fact implies that $\vec{\Delta}_{g'g}(S) \leq \gamma g(S)$. Thus, $\Delta_{gg'} \leq 2\gamma g(S)$ and therefore, $\mathcal{G}'$ is a 2γ-cover of $\mathcal{G}$ (see the proof of Lemma 7.3.8 for definitions of $\Delta_{gg'}$ and $\vec{\Delta}_{gg'}$). Since $|\mathcal{G}'|$ is $O(\log hn)$, the additive loss of RSO$_{(\mathcal{G},\mathcal{A})}$ is $O(h \log\log nh)$.[4]

We can also apply the discretization technique by defining $\mathcal{G}_\alpha$ to be the set of offers induced by the set of all constant-price functions whose price $v \in [1, h]$ is a power of $(1+\alpha)$ and $\alpha = \frac{\epsilon}{2}$. Clearly, if we can get revenue at least $(1-\frac{\epsilon}{2})$ times the optimal in this class, we will be within $(1-\epsilon)$ of the optimal fixed price overall. For example, Corollary 7.3.2 ($\mathcal{A}$ can trivially find the best offer in $\mathcal{G}'$ by simply trying all of them) shows that with probability $1-\delta$ we get at least $1-\epsilon$ times the revenue of the optimal take-it-or-leave-it offer so long as the number of bidders n is at least $\frac{72h}{\epsilon^2} \ln(\frac{4 \ln h}{\epsilon\delta}) = O(h \log\log h)$.

[4] It is interesting to contrast these results with that of [121] which showed that RSO over the set of constant-price functions is near 6-competitive with the promise that $n \gg h$.

7.4.1 Data Dependent Bounds

We can use the high level idea of our structural risk minimization reduction in order to get a better *data dependent* bound for the digital good auction. In particular, we can replace the "h" term in the additive loss with the actual sale price used by the optimal take-it-or-leave-it offer (in fact, even better, the lowest sales price needed to generate near-optimal revenue), yielding a much better bound when most of the profit to be made is from the low bids. The idea is that rather than penalizing the "complexity" of the offer in the usual sense, we instead penalize the use of higher prices.

Let $q_i = (1+\alpha)^i$ and offer g_i be the take-it-or-leave-it price of q_i. Define $\bar{\mathcal{G}} = \{g_1\}, \{g_2\}, \ldots$ and consider the auction RSO-SRM$_{\bar{\mathcal{G}},\mathsf{pen}}$ with $\mathsf{pen}(\{g_i\})$ specified from Section 7.3.2 to be $\frac{8q_i}{\epsilon^2}\ln\left(\frac{8i^2}{\delta}\right)$. The following is an a corollary of of Theorem 7.3.6.

Corollary 7.4.1 *For any given value of n, ϵ, and δ, with probability $1-\delta$, the revenue of RSO-SRM$_{(\bar{\mathcal{G}},\mathsf{pen})}$ is at least $\max_i\left[(1-\epsilon)g_i(S) - 2\mathsf{pen}(\{g_i\})\right]$, where $\mathsf{pen}(\{g_i\}) = \frac{8q_i}{\epsilon^2}\ln\left(\frac{8i^2}{\delta}\right)$.*

In other words, if the optimal take-it-or-leave-it offer has a sale price of p, then RSO-SRM$_{(\bar{\mathcal{G}},\mathsf{pen})}$ has convergence rate bounded by $O(p\log\log h)$ instead of $O(h\log\log h)$ as provided by our generic analysis of RSO$_{(\mathcal{G},\mathcal{A})}$.

7.4.2 A Special Purpose Analysis for the Digital Good Auction

In this section we present a refined data independent analysis for the digital good auction. Specifically, we can show for an optimal algorithm $\mathcal{A}$, that:

Theorem 7.4.2 *For $\delta < \frac{1}{2}$, with probability $1-\delta$, RSO$_{(\mathcal{G}_\alpha,\mathcal{A})}$ obtains profit at least*

$$\mathrm{OPT}_{\mathcal{G}_\alpha} - 8\sqrt{h\,\mathrm{OPT}_{\mathcal{G}_\alpha}\log\left(\tfrac{1}{\alpha\delta}\right)}.$$

Corollary 7.4.3 *For $\delta < \frac{1}{2}$ and $\alpha = \frac{\epsilon}{2}$, so long as $\mathrm{OPT}_{\mathcal{G}_\alpha} \geq (\frac{16}{\epsilon})^2 h\log\left(\frac{2}{\epsilon\delta}\right)$, then with probability at least $1-\delta$, the profit of RSO$_{(\mathcal{G}_\alpha,\mathcal{A})}$ is at least $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}$.*

The above corollary improves over our basic discretization results using Theorem 7.3.1 by an $O(\log\log h)$ factor in the convergence rate.

To prove Theorem 7.4.2, let us introduce some notation. For the offer g_v induced by the take-it-or-leave-it offer of price v, let n_v denote the number of winners (bidders whose value is at least v), and let $r_v = v\cdot n_v$ denote the profit of g_v on S. Denote by $\hat{r}_v$ the observed profit of g_v on S_1 (and so $\hat{r}_v = v\cdot\hat{n}_v$, where $\hat{n}_v$ is the number of winners in S_1 for g_v). So, we have $\mathbf{E}[\hat{r}_v] = \frac{r_v}{2}$. We now begin with the following lemma.

Lemma 7.4.4 *Let $\epsilon < 1$ and $\delta < \frac{1}{2}$. With probability at least $1-\delta$ we have that, for every $g_v \in \mathcal{G}_\alpha$ the observed profit on S_1 satisfies:*

$$\left|\hat{r}_v - \frac{r_v}{2}\right| \leq \max\left(\frac{h\log\left(\frac{1}{\alpha\delta}\right)}{\epsilon}, \epsilon r_v\right).$$

Proof: First for a given price v let $a_{n,v}$ be $|\hat{n}_v - \frac{n_v}{2}|$. To prove our lemma we will use the consequence of Chernoff bound we present in Appendix A.3, Theorem A.3.2. For any v and $j \geq 1$ we consider $n' = \frac{(1+\alpha)^j\log\left(\frac{1}{\alpha\delta}\right)}{\epsilon^2}$, and so we get

$$\Pr\left\{a_{n,v} \geq \epsilon\max\left(n_v, \frac{(1+\alpha)^j\log\left(\frac{1}{\alpha\delta}\right)}{\epsilon^2}\right)\right\} \leq 2e^{-2(1+\alpha)^j\log\left(\frac{1}{\alpha\delta}\right)}.$$

This further implies that we have $a_{n,v} \geq \epsilon \max\left(n_v, \frac{(1+\alpha)^j \log\left(\frac{1}{\alpha\delta}\right)}{\epsilon^2}\right)$ with probability at most $2(\alpha\delta)^{2(1+\alpha)^j}$. Therefore for $v = \frac{h}{(1+\alpha)^j}$ we have

$$\Pr\left\{\left|\hat{r}_v - \frac{r_v}{2}\right| \geq \max\left(\frac{h\log\left(\frac{1}{\alpha\delta}\right)}{\epsilon}, \epsilon r_v\right)\right\} \leq 2(\alpha\delta)^{2(1+\alpha)^j},$$

and so the probability that there exists a $g_v \in \mathcal{G}_\alpha$ such that $\left|\hat{r}_v - \frac{r_v}{2}\right| \geq \max\left(\frac{h}{\epsilon}, \epsilon r_v\right)$ is at most $2\sum_j (\alpha\delta)^{2(1+\alpha)^j} \leq 2\sum_{j'} \frac{1}{\alpha}(\alpha\delta)^{2\cdot 2^{j'}} \leq \delta$. This implies that with high probability, at least $1-\delta$, we have that simultaneously, for every $g_v \in \mathcal{G}_\alpha$ the observed revenue on S_1 satisfies:

$$\left|\hat{r}_v - \frac{r_v}{2}\right| \leq \max\left(\frac{h\log\left(\frac{1}{\alpha\delta}\right)}{\epsilon}, \epsilon r_v\right),$$

as desired. ■

Proof of Theorem 7.4.2: Assume now that it is the case that for every $g_v \in \mathcal{G}_\alpha$ we have

$$\left|\hat{r}_v - \frac{r_v}{2}\right| \leq \max\left(\frac{H}{\epsilon}, \epsilon r_v\right),$$

where $H = h\log\left(\frac{1}{\alpha\delta}\right)$. Let v^* be the optimal price level among prices in $\mathcal{G}_\alpha$, and let $\tilde{v}^*$ be the price that looks best on S_1. Obviously, our gain on S_2 is $r_{\tilde{v}^*} - \hat{r}_{\tilde{v}^*}$. We have

$$\hat{r}_{v^*} \geq \frac{r^*_v}{2} - \frac{H}{\epsilon} - \epsilon r_{v^*} = r_{v^*}\frac{1-2\epsilon}{2} - \frac{H}{\epsilon},$$

$$\hat{r}_{\tilde{v}^*} \geq \hat{r}_{v^*}, \text{ and } \hat{r}_{\tilde{v}^*} \leq \frac{r_{\tilde{v}^*}}{2} + \frac{H}{\epsilon} + \epsilon r_{\tilde{v}^*} \leq \frac{r_{\tilde{v}^*}}{2} + \frac{H}{\epsilon} + \epsilon r_{v^*},$$

and therefore $r_{\tilde{v}^*} - \hat{r}_{\tilde{v}^*} \geq \hat{r}_{\tilde{v}^*} - \frac{H}{\epsilon} - \epsilon r_{v^*}$, which finally implies that

$$r_{\tilde{v}^*} - \hat{r}_{\tilde{v}^*} \geq r_{v^*}\left(\frac{1}{2} - 2\epsilon\right) - 2\frac{H}{\epsilon}.$$

This implies that with probability at least $1 - \frac{\delta}{2}$ our gain on S_2 is at least $r_{v^*}\left(\frac{1}{2} - 2\epsilon\right) - 2\frac{H}{\epsilon}$, and similarly our gain on S_1 is at least $r_{v^*}\left(\frac{1}{2} - 2\epsilon\right) - 2\frac{H}{\epsilon}$. Therefore, with probability $1-\delta$, our revenue is

$$\mathrm{OPT}_{\mathcal{G}_\alpha}(1-4\epsilon) - 4\frac{h\log\left(\frac{1}{\alpha\delta}\right)}{\epsilon}.$$

Optimizing the bound we set $\epsilon = \sqrt{\frac{h\log\left(\frac{1}{\alpha\delta}\right)}{OPT_{\mathcal{G}_\alpha}}}$ and get a revenue of

$$OPT_{\mathcal{G}_\alpha} - 8\sqrt{h\, OPT_{\mathcal{G}_\alpha} \;\; \log\left(\frac{1}{\alpha\delta}\right)},$$

which completes the proof. ■

7.5 Attribute Auctions

We now consider applying our general bounds (Section 7.3) to attribute auctions. For attribute auctions an offer is a function from the publicly observable attribute of an agent to a take-it-or-leave-it price. As such, we identify such an offer with its *pricing function*. We begin by instantiating the results in Section 7.3 for market pricing auctions, in which we consider pricing functions that partition the attribute space into market segments and offer a fixed price in each. We show how one can use standard combinatorial dimensions in learning theory, e.g. the Vapnik-Chervonenkis (VC) dimension [18, 69, 103, 149, 203], in order to bound the complexity of these classes of offers. We then give an analysis for very general offer classes induced by general pricing functions over the attribute space that uses the notion of covers defined in Section 7.3.3.

7.5.1 Market Pricing

For attribute auctions, one natural class of pricing functions are those that segment bidders into *markets* in some simple way and then offer a single sale price in each market segment. For example, suppose we define $\mathcal{P}_k$ to be the set of functions that choose k bidders $b_1, \ldots, b_k$; use these as cluster centers to partition S into k markets based on distance to the nearest center in attribute space; and then offer a single price in each market. In that case, if we discretize prices to powers of $(1+\epsilon)$, then clearly the number of functions in the offer class $\mathcal{G}_k$ induced by the pricing class $\mathcal{P}_k$, is at most $n^k(\log_{1+\epsilon} h)^k$, so Corollary 7.3.2 implies that so long as $n \geq \frac{18h}{\epsilon^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln n + k\ln\left(\log_{1+\epsilon} h\right)\right]$ and assuming we can solve the optimization problem, then with probability at least $1-\delta$, we can get profit at least $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}_k}$.

We can also consider more general ways of defining markets. Let C be any class of subsets of $\mathcal{X}$, which we will call *feasible markets*. For k a positive integer, we consider $F_{k+1}(C)$ to be the set of all pricing functions of the following form: pick k disjoint subsets $\mathcal{X}_1$,...,$\mathcal{X}_k \subseteq \mathcal{X}$ from C, and $k+1$ prices p_0,...,p_k discretized to powers of $1+\epsilon$. Assign price p_i to bidders in $\mathcal{X}_i$, and price p_0 to bidders not in any of $\mathcal{X}_1$,...,$\mathcal{X}_k$. For example, if $\mathcal{X} = \mathcal{R}^d$ a natural C might be the set of axis-parallel rectangles in $\mathcal{R}^d$. The specific case of $d = 1$ was studied in [60]. One can envision more complex partitions, using the membership of a bidder in $\mathcal{X}_i$ as a basic predicate, and constructing any function over it (e.g., a decision list).

We can apply the results in Section 7.3 by using the machinery of VC-dimension to count the number of distinct such functions over any given set of bidders S. In particular, let $D = \mathrm{VCdim}(C)$ be the VC-dimension of C and assume $D < \infty$. Define $C[S]$ to be the number of distinct subsets of S induced by C. Then, from Sauer's Lemma $C[S] \leq \left(\frac{en}{D}\right)^D$, and therefore the number of different pricing functions in $F_k(C)$ over S is at most $\left(\log_{1+\epsilon} h\right)^k \left(\frac{en}{D}\right)^{kD}$. Thus applying Corollary 7.3.2 here we get:

Corollary 7.5.1 *Given a β-approximation algorithm $\mathcal{A}$ for optimizing over the offer class $\mathcal{G}_k$ induced by the class of pricing functions $F_k(C)$, then so long as $\mathrm{OPT}_{\mathcal{G}_k} \geq n$ and the number of bidders n satisfies*

$$n \geq \frac{18h\beta}{\epsilon^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{1}{\epsilon}\ln h\right) + kD\ln\left(\frac{ne}{D}\right)\right],$$

then with probability at least $1-\delta$, the profit of $\mathrm{RSO}_{\mathcal{G}_k,\mathcal{A}}$ is at least $(1-\epsilon)\frac{\mathrm{OPT}_{\mathcal{G}_k}}{\beta}$.

The above lemma has "n" on both sides of the inequality. Simple algebra yields:

Corollary 7.5.2 *Given a β-approximation algorithm $\mathcal{A}$ for optimizing over the offer class $\mathcal{G}_k$ induced by the class of pricing functions $F_k(C)$, then so long as $\mathrm{OPT}_{\mathcal{G}_k} \geq n$ and the number of bidders n satisfies*

$$n \geq \frac{36h\beta}{\epsilon^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{1}{\epsilon}\ln h\right) + kD\ln\left(\frac{36kh\beta}{\epsilon^2}\right)\right],$$

then with probability at least $1-\delta$*, the profit of* $\text{RSO}_{\mathcal{G}_k,\mathcal{A}}$ *is at least* $(1-\epsilon)\frac{\text{OPT}_{\mathcal{G}_k}}{\beta}$.

Proof: Since $\ln a \leq ab - \ln b - 1$ for all $a, b > 0$, we obtain: $\frac{18kDh\beta}{\epsilon^2}\ln n \leq \frac{n}{2} + \frac{18kDh\beta}{\epsilon^2}\ln\left(\frac{36kDh\beta}{e\epsilon^2}\right)$. Therefore, it suffices to have:

$$n \geq \frac{n}{2} + \frac{18h\beta}{\epsilon^2}\Big[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{1}{\epsilon}\ln h\right) + kD\ln\left(\frac{36kh\beta}{\epsilon^2}\right)\Big],$$

so

$$n \geq \frac{36h\beta}{\epsilon^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{1}{\epsilon}\ln h\right) + kD\ln\left(\frac{36kh\beta}{\epsilon^2}\right)\right]$$

suffices. ■

For certain classes C we can get better bounds. In the following, denote by C_k the concept class of unions of at most k sets from C, and let L be $\lceil \log_{1+\epsilon} h \rceil$. If C is the class of intervals on the line, then the VC-dimension of C_k is $2k$, and so the number of different pricing functions in $F_k(C)$ over S is at most $L^k\left(\frac{en}{2k}\right)^{2k}$; also, if C is the class of all axis parallel rectangles in d dimensions, then the VC-dimension of C_k is $O(kd)$ [107]. In these cases we can remove the $\log k$ term in our bounds, which is nice because it means we can interpret our results (e.g., Corollary 7.5.2) as charging OPT a penalty for each market it creates. However, we do not know how to remove this $\log k$ term in general, since in general the VC-dimension of C_k can be as large as $2Dk\log(2Dk)$ (see [57, 102]).

Corollary 7.5.2 gives a guarantee in the revenue of $\text{RSO}_{\mathcal{G}_k,\mathcal{A}}$ so long as we have enough bidders. In the following, for $k \geq 0$ let $\text{OPT}_k = \text{OPT}_{\mathcal{G}_k}$. We can also use Corollaries 7.3.5 and 7.5.2 to show a bound that holds for all n, but with an additive loss term.

Theorem 7.5.3 *For any given value of* n, k, ϵ*, and* δ*, with probability at least* $1-\delta$*, the revenue of* $\text{RSO}_{\mathcal{G}_k,\mathcal{A}}$ *is*

$$\tfrac{1}{\beta}\left[(1-\epsilon)\,\text{OPT}_k - h \cdot r_F(k, D, h, \epsilon, \delta)\right],$$

where $r_F(k, D, h, \epsilon, \delta) = O\left(\frac{kD}{\epsilon^2}\ln\left(\frac{kDh}{\epsilon\delta}\right)\right)$.

Proof: For simplicity, we show the proof for $\beta = 1$, the general case is similar. We prove the bound with the "$(1-\epsilon)$" term replaced by the term $\min\left(\frac{(1-\epsilon')^2}{1+\epsilon'}, 1-2\epsilon'\right)$, which then implies our desired result by simply using $\epsilon' = \frac{\epsilon}{3}$. If

$$n \geq \frac{36h}{\epsilon'^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{1}{\epsilon'}\ln h\right) + kD\ln\left(\frac{36kh}{\epsilon'^2}\right)\right],$$

then the desired statement follows directly from Corollary 7.5.2. Otherwise, consider first the case when we have

$$\text{OPT}_k \geq \frac{4h}{\epsilon'^2(1-\epsilon')}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{1}{\epsilon'}\ln h\right) + kD\ln\left(\frac{ne}{D}\right)\right].$$

Let g_i be the optimal offer in $\mathcal{G}_k$ over S_i, for $i = 1, 2$, and let g_{OPT} be the optimal offer in $\mathcal{G}_k$ over S (and so $g_i(S_i) \geq g_{\text{OPT}}(S_i)$). From Corollary 7.3.5, we have

$$g_{\text{OPT}}(S_i) \geq \frac{2h}{\epsilon'^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{1}{\epsilon'}\ln h\right) + kD\ln\left(\frac{ne}{D}\right)\right] \quad \text{for} \quad i = 1, 2.$$

So,

$$g_i(S_i) \geq \frac{2h}{\epsilon'^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{1}{\epsilon'}\ln h\right) + kD\ln\left(\frac{ne}{D}\right)\right].$$

Using again Corollary 7.3.5, we obtain $g_i(S_j) \geq \frac{1-\epsilon'}{1+\epsilon'} g_i(S_i)$ for $j \neq i$, which then implies the desired result. To complete the proof notice that if both

$$\mathrm{OPT}_k \leq \frac{4h}{\epsilon'^2(1-\epsilon')}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{1}{\epsilon'}\ln h\right) + kD\ln\left(\frac{ne}{D}\right)\right]$$

and

$$n \leq \frac{4h}{\epsilon'^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{2}{\epsilon'}\ln h\right) + kD\ln\left(\frac{4kh}{\epsilon'^2}\right)\right],$$

then we easily get the desired statement. ■

Finally, as in Theorem 7.3.6 we can extend our results to use structural risk minimization, where we want the algorithm to optimize over k, by viewing the additive loss term, $h \cdot r_F(\cdot)$, as a penalty function.

Theorem 7.5.4 *Let $\bar{\mathcal{G}}$ be the sequence $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n$ of offer classes induced by the sequence of classes of pricing functions $F_1(C), F_2(C), \ldots, F_n(C)$. Then for any value of n, ϵ and δ with probability $1-\delta$ the revenue of RSO-SRM$_{\bar{\mathcal{G}},\mathsf{pen}}$ is*

$$\max_k \left((1-\epsilon)\,\mathrm{OPT}_k - h \cdot r_F(k, D, h, \epsilon, \delta)\right),$$

where $\mathsf{pen}(F_k(C)) = \frac{h}{2} \cdot r_F(k, D, h, \epsilon, \delta) = O\left(\frac{kD}{\epsilon^2}\ln\left(\frac{kDh}{\epsilon\delta}\right)\right)$.

To illustrate the tightness of Theorem 7.5.3, notice that even for the special case of pricing using interval functions (the case of $d = 1$ studied in [60]), the following lower bound holds.

Theorem 7.5.5 *Let $\mathcal{X} = \mathcal{R}$ and let C_k be the class of k intervals over $\mathcal{X}$. Then there is no incentive compatible mechanism whose expected revenue is at least $\frac{3}{4}\,\mathrm{OPT}_k - o(kh)$.*

That is, an additive loss linear in kh is necessary in order to achieve a multiplicative ratio of at least $3/4$.

Proof: Consider $\frac{kh}{2}$ bidders with distinct attributes (for instance, say bidder i has attribute i), each of whom independently has a $\frac{1}{h}$ probability of having valuation h and a $1-\frac{1}{h}$ probability of having valuation 1. Then, any incentive-compatible mechanism has expected profit at most $\frac{kh}{2}$ because for any given bidder and any given proposed price, the expected profit (over randomization in the bidder's valuation) is at most 1. However, there is at least a 50% chance we will have at least $\frac{k}{2}$ bidders of valuation h, and in that case OPT_k can give $\frac{k}{2}-1$ of those bidders a price of h and the rest a price of 1 for an expected profit of $\left(\frac{k}{2}-1\right)h + \left(\frac{kh}{2} - \frac{k}{2} + 1\right)1 = kh - h - \frac{k}{2} + 1$. On the other hand even if that does not occur, we always have $\mathrm{OPT}_k \geq \frac{kh}{2}$. So, the expected profit of OPT_k is at least $3\frac{kh}{4} - \frac{h}{2} - \frac{k}{4}$. Thus, the profit of the incentive-compatible mechanism is at most $\frac{3}{4}\,\mathrm{OPT}_k - \frac{kh}{16} + o(kh)$. ■

We note that a similar lower bound holds for most base classes. Also for the case of intervals on the line, both our auction and the auction in [60] match this lower bound up to constant factors.

7.5.2 General Pricing Functions over the Attribute Space

In this section we generalize the results in Section 7.5.1 in two ways: we consider general classes of pricing functions (not just piecewise-constant functions defined over markets), and we remove the need to discretize by instead using the covering arguments discussed in Section 7.3.3. This allows us to consider offers based on linear or quadratic functions of the attributes, or perhaps functions that divide the attribute space into markets and use pricing functions are linear in the attributes (rather than constant) in each market. The key point of this section is that we can bound the size of the L_1 multiplicative cover in an attribute auction in terms of natural quantities.

Assume in the following that $\mathcal{X} \subseteq \mathcal{R}^d$, let $\mathcal{P}$ be a fixed class of pricing functions over the attribute space $\mathcal{X}$ and let $\mathcal{G}$ be the induced class of offers. Let $\mathcal{P}_d$ be the class of decision surfaces (in $\mathcal{R}^{d+1}$) induced by $\mathcal{P}$: that is, to each $q \in \mathcal{P}$ we associate the set of all $(x, v) \in \mathcal{X} \times [1, h]$ such that $q(x) \leq v$. Also, let us denote by D the VC-dimension of class $\mathcal{P}_d$. We can then show that:

Theorem 7.5.6 *Given the offer class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$, then so long as $\mathrm{OPT}_{\mathcal{G}} \geq n$ and the number of bidders n satisfies*

$$n \geq \frac{154h\beta}{\epsilon^2}\left[\ln\left(\frac{2}{\delta}\right) + D\ln\left(\frac{154h\beta}{\epsilon^2}\left(\frac{12}{\epsilon}\ln h + 1\right)\right)\right],$$

then with probability at least $1-\delta$, the profit of $\mathrm{RSO}_{(\mathcal{G},\mathcal{A})}$ is at least $(1-\epsilon)\frac{\mathrm{OPT}_{\mathcal{G}}}{\beta}$.

The key to the proof is to exhibit an L_1 multiplicative cover of $\mathcal{G}$ whose size is exponential in D only, and then to apply Corollary 7.3.10.

Proof: Let $\alpha = \frac{\epsilon}{12}$. For each bidder (x, v) we conceptually introduce $O(\frac{1}{\alpha}\ln h)$ "phantom bidders" having the same attribute value x and bid values $1, (1+\alpha), (1+\alpha)^2, \cdots, h$. Let S^* be the set S together with the set of all phantom bidders; let $n^* = |S^*|$. Let Split be the set of possible splittings of S^* with surfaces from $\mathcal{P}_d$. We clearly have $|\mathsf{Split}| \leq \mathcal{P}_d[S^*]$. For each element $s \in \mathsf{Split}$ consider a representative function in $\mathcal{G}$ that induces splitting s in terms of its winning bidders, and let $\mathsf{Split}_{\mathcal{G}}$ be the set of these representative functions. Let $\mathcal{G}'$ be the offer class induced by the pricing class $\mathsf{Split}_{\mathcal{G}}$. Notice that $\mathcal{G}'$ is actually an L_1 multiplicative α-cover for $\mathcal{G}$ with respect to S, since for every offer in $\mathcal{G}$ there is a offer in $\mathcal{G}'$ that extracts nearly the same profit from every bidder; i.e., for every offer in $g \in \mathcal{G}$, there exists $g' \in \mathcal{G}'$ such that for every $(x, v) \in S$, we have both

$$g'((x,v)) \leq (1+\alpha)g((x,v)) \quad \text{and} \quad g((x,v)) \leq (1+\alpha)g'((x,v)).$$

From Sauer's lemma we know $|\mathsf{Split}_{\mathcal{G}}| \leq \left(\frac{n^* e}{D}\right)^D$, and applying Corollary 7.3.10, we finally get the desired statement by using simple algebra as in Corollary 7.5.2. ■

The above theorem is the analog of Corollary 7.3.2. Using it and Theorem 7.3.9, it is easy to derive a bound that holds for all n (i.e., the analog of Theorem 7.5.3). One can further easily extend these results to get bounds for the corresponding SRM auction (as done in Theorem 7.5.4).

7.5.3 Algorithms for Optimal Pricing Functions

There has been relatively little work on the algorithmic question of computing optimal pricing functions in general attribute spaces. However, for single-dimentional attributes and piece-wise constant pricing functions [60] discusses an optimal polynomial time dynamic program. For single-dimentional attributes and monotone pricing functions, [9] gives a polynomial time dynamic program. The problem of computing the optimal of linear pricing function over m-dimentional attributes generalizes the problem of item-pricing (m distinct items) for single-minded combinatorial consumers (see Section 7.6.4) that has been shown to be hard to approximate to better than a $\log^{\delta}(m)$ factor for some $\delta > 0$ [101].

7.6 Combinatorial Auctions

Combinatorial auctions have received much attention in recent years because of the difficulty of merging the algorithmic issue of computing an optimal outcome with the game-theoretic issue of incentive compatibility. To date, the focus primarily has been on the problem of optimizing social welfare: partitioning

a limited supply of items among bidders to maximize the sum of their valuations. We consider instead the goal of profit maximization for the seller in the case that the items for sale are available in unlimited supply.[5] We consider the general version of the combinatorial auction problem as well as the special cases of *unit-demand* bidders (each bidder desires only singleton bundles) and *single-minded* bidders (each bidder has a single desired bundle).

It is interesting to restrict our attention to the case of item-pricing, where the auctioneer intuitively is attempting to set a price for each of the distinct items and bidders then choose their favorite bundle given these prices. Item-pricing is without loss of generality for the unit-demand case, and general bundle-pricing can be realized with an auction with $m' = 2^m$ "items", one for each of possible bundle of the original m items.[6]

First notice that if the set of allowable item pricings are constrained to be integral, $\mathcal{G}_{\mathbb{Z}}$, then clearly there are at most $|\mathcal{G}_{\mathbb{Z}}| = (h+1)^m$ possible item pricings. By Corollary 7.3.2 we get that $\tilde{O}\left(\frac{hm}{\epsilon^2}\right)$ bidders are sufficient to achieve profit close to $\mathrm{OPT}_{\mathcal{G}_{\mathbb{Z}}}$. Generally it is possible to do much better if non-integral item-pricings are allowed, i.e., $\mathrm{OPT}_{\mathcal{G}}(S) \gg \mathrm{OPT}_{\mathcal{G}_{\mathbb{Z}}}(S)$. In these settings we can still get good bounds following the guidelines established in Section 7.3.3, by either considering an offer class $\mathcal{G}'$ induced by discretization (see Section 7.6.1), or from counting possible outcomes in $\mathcal{G}_{\mathcal{A}}$ (see Section 7.6.2). A summary of our results is given in Table 7.6.

Table 7.1: Size of offer classes for combinatorial auctions.

	general	unit-demand	single-minded
$\lvert\mathcal{G}'\rvert$	$O(\log^m_{1+\epsilon 2} \frac{nm}{\epsilon})$	$O(\log^m_{1+\epsilon 2} \frac{n}{\epsilon})$	$O(\log^m_{1+\epsilon} \frac{nm}{\epsilon})$
$\lvert\mathcal{G}_{\mathcal{A}}\rvert$	$n^m 2^{2m2}$	$n^m(m+1)^{2m}$	$(n+m)^m$

We can apply Theorem 7.3.1 and Corollary 7.3.2 to the sizes of the offer classes in Table 7.6 to get bounds on the profit of random sampling auctions for combinatorial item pricing. In particular, using Corollary 7.3.2 we get that $\tilde{O}\left(\frac{hm^2}{\epsilon^2}\right)$ bidders are sufficient to achieve revenue close to the optimum item-pricing in the general case, and $\tilde{O}\left(\frac{hm}{\epsilon^2}\right)$ bidders are sufficient for the unit-demand case. Also, by using Theorem 7.3.1 instead of Corollary 7.3.2 we can replace the condition on the number of bidders with a condition on $\mathrm{OPT}_{\mathcal{G}}$, which gives a factor of m improvement on the bound given by [119].

As before we let $h = \max_{g\in\mathcal{G}, i\in S} g(i)$. In particular, this implies that $\mathrm{OPT}_{\mathcal{G}} \geq h$ which will be important later in this section.

7.6.1 Bounds via Discretization

As shown in Section 7.3.3, we can obtain good bounds if we are willing to optimize over a set $\mathcal{G}'$ of offers induced by a small set of discretized prices satisfying that $\mathrm{OPT}_{\mathcal{G}'}$ is close to $\mathrm{OPT}_{\mathcal{G}}$. Prior to this work, [131] shows how to construct discretized classes $\mathcal{G}'$ with $\mathrm{OPT}_{\mathcal{G}'} \geq \frac{1}{1+\epsilon}\mathrm{OPT}_{\mathcal{G}}$ and size $O(m^m \log^m_{1+\epsilon} \frac{n}{\epsilon})$ for the unit-demand case and size $O(\log^m_{1+\epsilon} \frac{nm}{\epsilon})$ for the single-minded case. Nisan [178] gives the basic argument necessary to generalize these results to obtain the result in Theorem 7.6.1 which applies to combinatorial auctions in general. We note in passing that Theorem 7.6.1 allows for generalization and improvement of the computational results of [131]. The discretization results we obtain are

[5]Other work focusing on profit maximization in combinatorial auctions include Goldberg and Hartline [119], Hartline and Koltun [131], Guruswami et al. [129], Likhodedov and Sandholm [161], and Balcan et al. [37].

[6]We make the assumption that all desired bundles contain at most one of each item. This assumption can be easily relaxed and our results apply given any bound on the number of copies of each item that are desired by any one consumer. Of course, this reduction produces an exponential blowup in the number of items.

summarized in the first row of Table 7.6.

Let $\mathbf{p} = (p_1, \ldots, p_m)$ be an item-pricing of the m items. Let $g_{\mathbf{p}}$ correspond to the offering pricing $\mathbf{p}$. The following is the main result of this section.

Theorem 7.6.1 *Let k be the size of the maximum desired bundle. Let $\mathbf{p}'$ be the optimal discretized price vector that uses item prices equal to 0 or powers of $(1+\epsilon)$ in the range $\left[\frac{h\epsilon}{nk}, h\right]$ and let $\mathbf{p}^*$ be the optimal price vector. Then we have:*

$$g_{\mathbf{p}'}(S) \geq (1 - 2\sqrt{\epsilon}) g_{\mathbf{p}^*}(S).$$

Proof: Let $\delta = \sqrt{\epsilon}$. For the optimal price vector $\mathbf{p}^*$ with item j priced at p_j^* (i.e., $g_{\mathbf{p}^*}(S) = \mathrm{OPT}_{\mathcal{G}}$), consider a price vector $\mathbf{p}$ with p_j in $[(1-\delta)p_j^*, (1-\delta+\delta^2)p_j^*]$ if $p_j^* \geq \frac{h\delta^2}{nk}$ and 0 otherwise, where $p_j = (1+\epsilon)^k$ for some integer k (note that such a price vector always exists). We show now that $g_{\mathbf{p}}(S) \geq (1 - 2\sqrt{\epsilon}) g_{\mathbf{p}^*}(S)$, which clearly implies the desired result.

Let J be a multi-set of items and $\text{Profit}(J) = \sum_{j \in J} p_j^*$ be the payment necessary to purchase bundle J under pricing $\mathbf{p}^*$. Define $R_j = p_j^* - p_j$. Thus we have:

$$(\delta - \delta^2)p_j^* \leq R_j \leq \max\{\delta p_j^*, \tfrac{\delta^2 h}{nk}\} \leq \delta p_j^* + \tfrac{\delta^2 h}{nk}.$$

This implies that for any multiset J with $|J| \leq k$, we have the following upper and lower bounds:

$$\sum_{j \in J} R_j \quad \geq \quad (\delta - \delta^2)\text{Profit}(J)\,, \tag{7.6.1}$$

$$\sum_{j \in J'} R_j \quad \leq \quad \delta\text{Profit}(J') + \tfrac{h\delta^2}{n}. \tag{7.6.2}$$

Let J_i^* and J_i be the bundles that bidder i prefers under pricing $\mathbf{p}^*$ and $\mathbf{p}$, respectively. Consider bidder i who switches from bundle J_i^* to bundle J_i when the item prices are decreased from $\mathbf{p}^*$ to $\mathbf{p}$. This implies that:

$$\sum_{j \in J_i^*} R_j \leq \sum_{j \in J_i} R_j.$$

Combining this with equations (7.6.1) and (7.6.2) and canceling a common factor of δ we see that:

$$(1-\delta)\text{Profit}(J_i^*) \leq \text{Profit}(J_i) + \tfrac{h\delta}{n}.$$

Summing over all bidders i, we see that the total profit under our new pricing $\mathbf{p}$ is at least $(1-\delta)\,\mathrm{OPT}_{\mathcal{G}} - h\delta$. Since $\mathrm{OPT}_{\mathcal{G}} \geq h$, we finally obtain that the profit under $\mathbf{p}$ is at least $(1-2\delta)\,\mathrm{OPT}_{\mathcal{G}}$. ■

Note that we can now apply Theorem 7.6.1 by letting $\mathcal{G}'$ be the offer class induced by the class of item prices equal to 0 or powers of $(1+\epsilon)$ in the range $\left[\frac{h\epsilon}{nk}, h\right]$ (where k bounds the maximum size of a bundle). Using Theorem 7.3.1 we obtain the following guarantee:

Corollary 7.6.2 *Given a β-approximation algorithm $\mathcal{A}$ optimizing over $\mathcal{G}'$, then with probability at least $1-\delta$, the profit of $\mathrm{RSO}_{\mathcal{G}',\mathcal{A}}$ is at least $(1-3\epsilon)\mathrm{OPT}_{\mathcal{G}}/\beta$ so long as*

$$\mathrm{OPT}_{\mathcal{G}'} \geq \tfrac{18h\beta}{\epsilon^2}\left(m\ln(\log_{1+\epsilon^2} nk) + \ln\left(\tfrac{2}{\delta}\right)\right).$$

7.6.2 Bounds via Counting

We now show how to use the technique of counting possible outcomes (See Section 7.3.3) to get a bound on the performance of the random sampling auction with an algorithm $\mathcal{A}$ for item-pricing. This approach calls for bounding $|\mathcal{G}_{\mathcal{A}}|$, the number of different pricing schemes $\text{RSO}_{(\mathcal{G},\mathcal{A})}$ can possibly output. Our results for this approach are summarized in the second row of Table 7.6.

Recall that bidder i's utility for a bundle J given pricing $\mathbf{p}$ is $u_i(J,\mathbf{p}) = v_i(J) - \sum_{j\in J} p_j$. We now make the following claim about the regions of the space of possible pricings, $\mathcal{R}_+^m$, in which bidder i's most desired bundle is fixed.

Claim 1 *Let $P_i(J) = \{\mathbf{p} \mid \forall J', u_i(J,\mathbf{p}) \geq u_i(J',\mathbf{p})\}$. The set $P_i(J,\mathbf{p})$ is a polytope.*

Proof: This follows immediately from the observation that the region $P_i(J)$ is convex and the only way to pack convex regions into space is if they are polytopes.

To show that $P_i(J)$ is convex, suppose the allocation to a particular bidder for $\mathbf{p}$ and $\mathbf{p}'$ are the same, J. Then for any other bundle J' we have:

$$v_i(J) - \sum_{j\in J} p_j \geq v_i(J') - \sum_{j\in J'} p_j$$

and

$$v_i(J) - \sum_{j\in J} p'_j \geq v_i(J') - \sum_{j\in J'} p'_j.$$

If we now consider any price vector $\alpha\mathbf{p} + (1-\alpha)\mathbf{p}'$, for $\alpha \in [0,1]$, these imply:

$$v_i(J) - \sum_{j\in J} (\alpha p_j + (1-\alpha)p'_j) \geq v_i(J') - \sum_{j\in J'} (\alpha p_j + (1-\alpha)p'_j).$$

This clearly implies that this agent prefers allocation J on any convex combination of $\mathbf{p}$ and $\mathbf{p}'$. Hence the region of prices for which the agent prefers bundle J is convex. ■

The above claim shows that we can divide the space of pricings into polytopes based on an agent's most desirable bundle. Consider fixing an outcome, i.e., the bundles $J_1,\ldots,J_n$, obtained by agents $1,\ldots,n$, respectively. This outcome occurs for pricings in the intersection $\bigcap_{i\in S} P_i(J_i)$.

Definition 7.6.1 *For a set of agents S, let Verts_S denote the set of vertices of the polytopes that partition the space of prices by the allocation produced. I.e., $\mathsf{Verts}_S = \{\mathbf{p}$ such that $\mathbf{p}$ is a vertex of the polytope containing $\bigcap_{i\in S'} P_i(J_i)$ for some $i \in S' \subset S$ and bundles $J_i\}$.*

Claim 2 *For $S' \subseteq S$ we have $\mathsf{Verts}_{S'} \subseteq \mathsf{Verts}_S$.*

Proof: Follows immediately from the definition of Verts_S and basic properties of polytopes. ■

Now we consider optimal pricings. Note that when fixing an allocation $J_1,\ldots,J_n$ we are looking for an optimal price point within the polytope that gives this allocation. Our objective function for this optimization is linear. Let n_j be the number of copies of item j allocated by the allocation. The seller's payoff for prices $\mathbf{p} = (p_1,\ldots,p_m)$ is $\sum_j p_j n_j$. Thus, all optimal pricings of this allocation lie on facets of the polytope and in particular there is an optimal pricing that is at a vertex of the polytope. Over the space of all possible allocations, all optimal pricings are on facets of the allocation defining polytopes and there exists an optimal pricing that is at a vertex of one of the polytopes.

Lemma 7.6.3 *Given an algorithm $\mathcal{A}$ that always outputs a vertex of the polytope then $\mathcal{G}_{\mathcal{A}} \subseteq \mathsf{Verts}_S$.*

Proof: This follows from the fact that $\text{RSO}_{(\mathcal{G},\mathcal{A})}$ runs $\mathcal{A}$ on a subset S' of S which has $\mathsf{Verts}_{S'} \subseteq \mathsf{Verts}_S$. $\mathcal{A}$ must pick a price vector from $\mathsf{Verts}_{S'}$. By Claim 2 this price vector must also be in Verts_S. This gives the lemma. ■

We now discuss getting a bound on Verts_S for n agents, m distinct items, and various types of preferences.

Theorem 7.6.4 *We have the following upper bounds on* $|\mathsf{Verts}_S|$*:*

1. $(n+m)^m$ *for single-minded preferences.*
2. $n^m(m+1)^{2m}$ *for unit-demand preferences.*
3. $n^m 2^{2m^2}$ *for arbitrary preferences.*

Proof: We consider how many possible bundles, M, an agent might obtain as a function of the pricing. An agent with single-minded preferences will always obtain one of $M_s = 2$ bundles: either their desired bundle or nothing (the empty bundle). An agent with unit-demand preferences receives one of the m items or nothing for a total of $M_u = m + 1$ possible bundles. An agent with general preferences receives one of the $M_g = 2^m$ possible bundles.[7]

We now bound the number of hyperplanes necessary to partition the pricing space into M convex regions (e.g., that specify which bundle the agent receives). For convex regions, each pair of regions can meet in at most one hyperplane. Thus, the total number of hyperplanes necessary to partition the pricing space into regions is at most $\binom{M}{2}$. Of course we wish to restrict our pricings to be non-negative, so we must add m additional hyperplanes at $p_j = 0$ for all j.

For all n agents, we simply intersect the regions of all agents. This does not add any new hyperplanes. Furthermore, we only need to count the m hyperplanes that restrict to non-negative pricings once. Thus, the total number of hyperplanes necessary for specifying the regions of allocation for n agents with M convex regions each, is $K = n\binom{M}{2} + m$. Thus, $K_s = n + m$, $K_u \leq n\binom{m+1}{2} + m \leq n(m+1)^2$, and $K_g \leq n\binom{2^m}{2} + m \leq n2^{2m}$ (for $m \geq 2$).

Of course, K hyperplanes in m dimensional space intersect in at most $\binom{K}{m} \leq K^m$ vertices. Not all of these intersections are vertices of polytopes defining our allocation, still K^m is an upper bound on the size of Verts_S. Plugging this in gives us the desired bounds of $(n+m)^m$, $n^m(m+1)^{2m}$, and $n^m 2^{2m^2}$ respectively for single-minded, unit-demand, and general preferences. ■

We note that are above arguments apply to approximation algorithms that always output a price corresponding to the vertex of a polytope as well. Though we do not consider this direction here, it is entirely possible that it is not computationally difficult to post-process the solution of an algorithm that is not a vertex of a polytope to get a solution that is on a vertex of a polytope.[8] This would further motivate the analysis above. If for some reason, restricting to algorithms that return vertices is undesirable, it is possible to use cover arguments on the set of vertices we obtain when we add additional hyperplanes corresponding to the discretization of the preceding section.

7.6.3 Combinatorial Auctions: Lower Bounds

We show in the following an interesting lower bound for combinatorial auctions.[9] Notice that our upper bounds and this lower bound are quite close.

[7] Here we make the assumption that desired bundles are simple sets. If they are actually multi-sets with bounded multiplicity k, then the agent could receive one of at most $M_g = (k+1)^m$ bundles.

[8] Notice that this is not immediate because of the complexity of representing an agent's combinatorial valuation.

[9] This proof follows the standard approach for lower bounds for revenue maximizing auctions that was first given by Goldberg et al. in [123].

Theorem 7.6.5 *Fix m and h. There exists a probability distribution on unit-demand single-minded agents such that the expected revenue of any incentive compatible mechanism is at most $\frac{mh}{2}$ whereas the expected revenue of* OPT *is at least* $0.7mh$.

Thus, this theorem states that in order to achieve a close multiplicative ratio with respect to OPT, one must have additive loss $\Omega(mh)$.

Proof: Consider the following probability distribution over valuations of agents preferences. Assume we have $n = \frac{mh}{2}$ agents in total, and $\frac{h}{2}$ agents desire item j only, $j \in \{1, \cdots m\}$.[10] Each of these agents has valuation h with probability $\frac{1}{h}$ and valuation 1 with probability $1 - \frac{1}{h}$.

Notice now any incentive-compatible mechanism has expected profit at most n. To see this, note that for each bidder, any proposed price has expected profit (over the randomization in the selection of his valuation) of at most 1. Moreover, the expected profit of $\text{OPT}_{\mathcal{G}}$ is at least $n + \frac{mh}{8}$. For each item j, there is a $1 - (1 - \frac{1}{h})^{h/2} \approx 0.4$ probability that some bidder has valuation h. For those items, $\text{OPT}_{\mathcal{G}}$ gets at least a profit of h. For the rest, $\text{OPT}_{\mathcal{G}}$ gets a profit of $\frac{h}{2}$. So, overall, $\text{OPT}_{\mathcal{G}}$ gets an expected profit of at least $0.4mh + 0.6m(h/2) = 0.7h$. All these together imply the desired result. ■

7.6.4 Algorithms for Item-pricing

Given standard complexity assumptions, most item-pricing problems are not polynomial time solvable, even for simple special cases. We review these results here. We focus our attention to the unlimited supply special case, though some of the work we mention also considers limited supply item-pricing. Algorithmic pricing problems in this form were first posed by Guruswami et al. [129] though item-pricing for unit-demand consumers with several alternative payment rules (i.e., rules that do not represent quasi-linear utility maximization) were independently considered by Aggarwal et al. [10].

For consumers with single-minded preferences, [129] gives a simple $O(\log mn)$ approximation algorithm. Demaine et al. [101] show the problem to be hard to approximate to better than a $\log^{\delta}(m)$ factor for some $\delta > 0$. Both Briest and Krysta [73] and Grigoriev et al. [126] proved that optimal pricing is weakly NP-hard for the special case known as "the highway problem" where there is a linear order on the items and all desired bundles are for sets of consecutive items (actually this hardness result follows for the more specific case where the desired bundles for any two agents, S_i and $S_{i'}$, satisfy one of: $S_i \subseteq S_{i'}$, $S_{i'} \subseteq S_i$, or $S_i \cup S_{i'} = \emptyset$). In the case when the cardinality of the desired bundles are bounded by k, Briest and Krysta [73] give an $O(k^2)$ approximation algorithm. In our work [24] we have improved this, by giving a simpler and better $O(k)$ approximation. Finally, when the number of distinct items for sale, m, is constant, Hartline and Koltun [131] show that it is possible to improve on the trivial $O(n^m)$ algorithm by giving a near-linear time approximation scheme. Their approximation algorithm is actually an exact algorithm for the problem of optimizing over a discretized set of item prices $\mathcal{G}'$ which is directly applicable to our auction $\text{RSO}_{(\mathcal{G}',\mathcal{A})}$, discussed above.

For consumers with unit-demand preferences, [129] (and [10] essentially) give a trivial logarithmic approximation algorithm and show that the optimization problem is APX-hard (meaning that standard complexity assumptions imply that there does not exist a polynomial time approximation scheme (PTAS) for the problem). Again, Hartline and Koltun [131] show how to improve on the trivial $O(n^m)$ algorithm in the case where the number of distinct items for sale, m, is constant. They give a near-linear time approximation scheme that is based on considering a discretized set of item prices; however, the discretization of Nisan [178] that we discussed above gives a significant improvement on their algorithm and also generalizes it to be applicable to the problem of item-pricing for consumers with general combinatorial preferences.

[10] Notice that these preferences are both unit-demand and single-minded.

7.7 Conclusions and Discussion

In this work we have made an explicit connection between machine learning and mechanism design. In doing so, we obtain a *unified* approach to considering a variety of profit maximizing mechanism design problems including many that have been previously considered in the literature.

Some of our techniques give suggestions for the *design* of mechanisms and others for their *analysis*. In terms of design, these include the use of discretization to produce smaller function classes, and the use of structural-risk-minimization to choose an appropriate level of complexity of the mechanism for a given set of bidders. In terms of analysis, these include both the use of basic sample-complexity arguments, and the notion of multiplicative covers for better bounding the true complexity of a given class of offers.

Our results substantially generalize the previous work on random sampling mechanisms by both broadening the applicability of such mechanisms and by simplifying the analysis. Our bounds on random sampling auctions for digital goods not only show how the auction profit approaches the optimal profit, but also weaken the required assumptions of [121] by a constant factor. Similarly, for random sampling auctions for multiple digital goods, our unified analysis gives a bound that weakens the assumptions of [119] by a factor of more than m, the number of distinct items. This multiple digital good auction problem is a special case of the a more general unlimited supply combinatorial auction problem for which we obtain the first positive worst-case results by showing that it is possible to approximate the optimal profit with an incentive-compatible mechanism. Furthermore, unlike the case for combinatorial auctions for social welfare maximization, our incentive-compatible mechanisms can be easily based on approximation algorithms instead of exact ones.

We have also explored the attribute auction problem that was proposed in [60] for 1-dimensional attributes in a much more general setting: the attribute values can be multi-dimensional and the target pricing functions considered can be arbitrarily complex. We bound the performance of random sampling auctions as a function of the complexity of the target pricing functions.

Our random sampling auctions assume the existence of exact or approximate pricing algorithms. Solutions to these pricing problem have been proposed for several of our settings. In particular, optimal item-pricings for combinatorial auctions in the single-minded and unit-demand special cases have been considered in [24, 73, 129, 131]. On the other hand for attribute auctions, many of the clustering and market-segmenting pricing algorithms have yet to be considered at all.

Chapter 8

Bibliography

[1] *http://www.kernel-machines.org/*. 1.1.2, 3.1

[2] *6th Kernel Machines Workshop*. NIPS, 2002. http://www-stat.ucdavis.edu/ nello/nips02.html. 1.1.2

[3] *The Seventh Workshop on Kernel Machines*. COLT, 2003. http://learningtheory.org/colt2003. 1.1.2

[4] *Workshop Graphical Models and Kernels*. NIPS, 2004. http://users.rsise.anu.edu.au/ smola/workshops/nips04. 1.1.2

[5] *Kernel Methods and Structured Domains*. NIPS, 2005. http://nips2005.kyb.tuebingen.mpg.de. 1.1.2

[6] 1.1.1

[7] D. Achlioptas. Database-friendly random projections. *Journal of Computer and System Sciences*, 66(4): 671–687, 2003. 6.1, 6.4

[8] D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *COLT*, 2005. 1.1.3, 4.1, 4.1.1

[9] G. Aggarwal and J. Hartline. Knapsack Auctions. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, 2006. 7.1, 7.5.3

[10] G. Aggarwal, T. Feder, R. Motwani, and A. Zhu. Algorithms for multi-product pricing. In *Proceedings of the International Colloquium on Automata, Languages, and Programming*, pages 72–83, 2004. 7.6.4

[11] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. In *STOC*, pages 684–693, 2005. 1

[12] P. Alimonti and V. Kann. Hardness of approximating problems on cubic graphs. In *Algorithms and Complexity*, 1997. 4.8.1

[13] N. Alon and N. Kahale. A spectral technique for coloring random 3-colorable graphs. *SIAM J. Computing*, 26(6):1733 – 1748, 1997. 1, 4.1.1

[14] N. Alon, W. Fernandez de la Vega, R. Kannan, and M. Karpinski. Random sampling and approximation of max-csps. *Journal of Computer and Systems Sciences*, 67(2):212–243, 2003. 1.1.3, 4, 4.1.2, 4.6, 4.6.4

[15] M.-R. Amini, O. Chapelle, and R. Ghani, editors. *Learning with Partially Classified Training Data*. Workshop, ICML'05, 2005. 2.1

[16] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1998. 5.1.1, 1

[17] D. Angluin. Queries revisited. *Theoretical Computer Science*, 313(2):175–194, 2004. 5.1.1, 1

[18] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999. 1.1.2, 1.1.4, 3.1, 3.2, 3.3.3, 6, 7.1, 7.3.3, 7.3.3, 7.5

[19] S. Arora and R. Kannan. Learning mixtures of arbitrary gaussians. In *ACM Symposium on Theory of Computing*, 2005. 1.1.3, 4.1, 4.1.1

[20] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54:317 – 331, 1997. 3.2, 3.3.3

[21] R. I. Arriaga and S. Vempala. An algorithmic theory of learning, robust concepts and random projection. pages 616–623, 1999. 6.1, 6.4, 6.4.1

[22] B. Awerbuch, Y. Azar, and A. Meyerson. Reducing truth-telling online mechanisms to online optimization. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, 2003. 7.1

[23] M.-F. Balcan and A. Blum. A PAC-style model for learning from labeled and unlabeled data. In *Proceedings of the Annual Conference on Computational Learning Theory*, 2005. 1.2

[24] M.-F. Balcan and A. Blum. On a theory of learning with similarity functions. In *International Conference on Machine Learning*, 2006. 1.2, 3.1, 3.3, 4, 3.3.3, 3.4, 3.4.2, 3.4.4, 9, 3.4.6, 3.6, 4.1, 4.1.3, 4.2, 4.4, 4.4, 7.6.4, 7.7

[25] M.-F. Balcan and A. Blum. Approximation Algorithms and Online Mechanisms for Item Pricing. *TOC*, 2007. 1.1.4, 1.2

[26] M.-F. Balcan and A. Blum. Approximation Algorithms and Online Mechanisms for Item Pricing. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, 2006. 1.1.4, 1.2

[27] M.-F. Balcan and A. Blum. An augmented PAC-model for semi-supervised learning. Book chapter in "Semi-Supervised Learning", O. Chapelle, B. Schlkopf, and A. Zien, eds., MIT press, 2006. 1, 1.2

[28] M. F. Balcan, A. Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. In *Advances in Neural Information Processing Systems*, 2004. 1.2, 2.3.2, 2.4.2

[29] M.-F. Balcan, A. Blum, J. Hartline, and Y. Mansour. Mechanism Design via Machine Learning. In *46th Annual IEEE Symposium on Foundations of Computer Science*, 2005. 1.2

[30] M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *International Conference on Machine Learning*, 2006. 1.2, 2.5.4, 7, 5, 2, 1, 5.2.1, 5.2.1, 5.2.2, 5.2.2

[31] M.-F. Balcan, A. Blum, and S. Vempala. On kernels, margins and low-dimensional mappings. *Machine Learning Journal*, 2006. 1.2, 3.3, 3.6, 4.1, 4.1.3, 4.2, 4.4

[32] M.-F. Balcan, A. Blum, H. Chan, and M.T. Hajiaghayi. A theory of loss-leaders: Making money by pricing below cost. In *Proc. 3rd International Workshop on Internet and Network Economics*. Lecture Notes in Computer Science, 2007. 1.1.4, 1.2

[33] M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Proceedings of the 20th Annual Conference on Computational Learning Theory (COLT)*, 2007. 1.2, 2.5.4, 5, 5.1.1, 5.1.1, 5.1.4, 5.1.4, 5.2.2, 5.2.3

[34] M.-F. Balcan, E. Even-Dar, S. Hanneke, M. Kearns, Y. Mansour, and J. Wortman. Asymptotic active learning. In *Workshop on Principles of Learning Design Problem.* In conjunction with the 21st Annual Conference on Neural Information Processing Systems (NIPS), 2007. 2.5.4, 5, 3, 5.3

[35] M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 2008. 1.2, 5, 5.2.2

[36] M.-F. Balcan, A. Blum, J. Hartline, and Y. Mansour. Reducing mechanism design to algorithm design via machine learning. *Journal of Computer and System Sciences*, 2008. to appear. 1.2

[37] M.-F. Balcan, A. Blum, and Y. Mansour. Item Pricing for Revenue Maximization. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, 2008. 1.2, 5

[38] M.-F. Balcan, A. Blum, and N. Srebro. A theory of learning with similarity functions. *Machine Learning Journal*, 2008. 1.2, 3.1, 3.3, 4, 3.4.4

[39] M.-F. Balcan, A. Blum, and N. Srebro. Improved guarantees for learning via similarity functions. In *COLT*, 2008. 1.2, 3.1, 4

[40] M.-F. Balcan, A. Blum, and S. Vempala. A discriminative framework for clustering via similarity functions. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, 2008. 1.2, 3.6

[41] M.-F. Balcan, S. Hanneke, and J. Wortman. The true sample complexity of active learning. In *Proceedings of the 21st Annual Conference on Computational Learning Theory (COLT)*, 2008. 1.2, 2.5.4, 5, 3, 5.3

[42] M.-F. Balcan, A. Blum, and A. Gupta. Approximate clustering without the approximation. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2009. 1.2, 4.7, 4.7

[43] P. Bartlett and S. Mendelson. Rademacher and Gaussian Complexities Risk Bounds and Structural Results. *Journal of Machine Learning Research*, 54(3):463–482, 2002. 2.1.1, 2.3.1, 2.5.4, 3.2

[44] P. Bartlett and J. Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In *Advances in Kernel Methods: Support Vector Learning*. MIT Press, 1999. 6.1, 6.3, 6.4.1

[45] P. Bartlett, S. Boucheron, and G. Lugosi. Model selection and error estimation. In *Proceedings of the 13th Annual Conference on Computational Learning Theory*. 2.3.1

[46] E. Baum and K. Lang. Query learning can work poorly when a human oracle is used. In *International Joint Conference on Neural Networks*, 1993. 5.1.1

[47] E. B. Baum. Polynomial time algorithms for learning neural nets. In *Proceedings of the third annual workshop on Computational learning theory*, pages 258 – 272, 1990. 2.1

[48] S. Ben-David. A priori generalization bounds for kernel based learning. In *NIPS Workshop on Kernel Based Learning*, pages 991 – 998, 2001. 6.1

[49] S. Ben-David. A framework for statistical clustering with constant time approximation for k-means clustering. *Machine Learning Journal*, 2007. 4.1.3

[50] S. Ben-David, N. Eiron, and H.-U. Simon. Limitations of learning via embeddings in euclidean half-spaces. *The Journal of Machine Learning Research*, 3:441 – 461, 2003. 3.4, 3.4.3, 6.1

[51] A. Ben-Israel and T.N.E. Greville. *Generalized Inverses: Theory and Applications*. Wiley, New York, 1974. 6.3

[52] G.M. Benedek and A. Itai. Learnability by fixed distributions. In *Proc. 1st Workshop Computat. Learning Theory*, pages 80–90, 1988. 3.4.3

[53] G.M. Benedek and A. Itai. Learnability with respect to a fixed distribution. *Theoretical Computer Science*, 86:377–389, 1991. 2.1, 2.1.1, 2.5.4

[54] K. P. Bennett and C. Campbell. Support vector machines: hype or hallelujah? *SIGKDD Explor. Newsl.*, 2(2): 1–13, 2000. 3.4, 3.4.2

[55] T. De Bie and N. Cristianini. Convex methods for transduction. In *Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems*, volume 16, 2003. 2.2

[56] T. De Bie and N. Cristianini. Convex transduction with the normalized cut. Internal Report 04-128, ESAT-SISTA, K.U.Leuven, 2004. 2.5.1

[57] C.L. Blake and C. J. Merz. UCI Repository of Machine Learning Databases. 1998. http://www.ics.uci.edu/mlearn/MLRepository.html. 7.5.1

[58] A. Blum. Machine learning theory. *Essay*, 2007. 1.1.1

[59] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proc. ICML*, pages 19–26, 2001. 1.1.1, 2.2, 2.3.2, 2.5.1

[60] A. Blum and J. Hartline. Near-Optimal Online Auctions. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms*, pages 1156 – 1163, 2005. 7.1, 7.3.1, 7.5.1, 7.5.1, 7.5.1, 7.5.3, 7.7

[61] A. Blum and R. Kannan. Learning an intersection of k halfspaces over a uniform distribution. *Journal of Computer and Systems Sciences*, 54(2):371–380, 1997. 2.1

[62] A. Blum and T. M. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998. 1.1.1, 2.1, 2.1.2, 2.2, 2.3.2, 2.4.2, 2.4.2, 1, 2.4.2, 2.5.4

[63] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using fourier analysis. In *Proceedings of the 26th Annual ACM Symposium on*

Theory of Computing, pages 253–262, 1994. 3.4.3

[64] A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22:35–52, 1998. 2.1.2, 2.4.2, 2.4.2, 2.4.2, 2.4.2

[65] A. Blum, V. Kumar, A. Rudra, and F. Wu. Online Learning in Online Auctions. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms*, pages 137 – 146, 2003. 7.1

[66] A. Blum, N. Bansal, and S. Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004. 1

[67] A. Blum, J. Lafferty, R. Reddy, and M. R. Rwebangira. Semi-supervised learning using randomized mincuts. In *ICML '04*, 2004. 2.3.2, 2.5.1

[68] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987. 5.1.2, 5.1.3

[69] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989. 2.1, 2.1.1, 7.1, 7.5

[70] C. Borgs, J. T. Chayes, N. Immorlica, M. Mahdian, and A. Saberi. Multi-unit auctions with budget-constrained bidders. In *Proceedings of the 6th ACM Conference on Electronic Commerce*, pages 44–51, 2005. 1.1.4, 7.2.4

[71] S. Boucheron, G. Lugosi, and P. Massart. A sharp concentration inequality with applications. *Random Structures and Algorithms*, 16:277–292, 2000. 2.3.1, A.1.1

[72] O. Bousquet, S. Boucheron, and G. Lugosi. Theory of Classification: A Survey of Recent Advances. *ESAIM: Probability and Statistics*, 2005. 1.1.3, 2.1.1, 2.3.1, 2.3.1, 4.1.2, 4.2, 5.1.2, 7.3.3

[73] P. Briest and P. Krysta. Single-Minded Unlimited Supply Pricing on Sparse Instances. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, 2006. 7.6.4, 7.7

[74] N. H Bshouty and N. Eiron. Learning monotone dnf from a teacher that almost does not answer membership queries. *Journal of Machine Learning Research*, 3:49–57, 2002. 5.1.1, 1

[75] M. Burnashev and K. Zigangirov. An interval estimation problem for controlled observations. *Problems in Information Transmission*, 10:223–231, 1974. 5.1.1, 5.1.4

[76] V. Castelli and T.M. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 16: 105–111, 1995. 2.1, 1, 2.5.2

[77] V. Castelli and T.M. Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory*, 42(6):2102–2117, 1996. 2.1, 1, 2.5.2

[78] R. Castro and R. Nowak. Minimax bounds for active learning. In *Proceedings of the 20th Annual Conference on Computational Learning Theory (COLT)*, 2007. 5.1.1, 5.1.1

[79] R. Castro, R. Willett, and R. Nowak. Faster rates in regression via active learning. In *Advances in Neural Information Processing Systems*, volume 18, 2006. 5.1.1, 5.1.1

[80] Rui M. Castro and Robert D. Nowak. Upper and lower error bounds for active learning. In *The 44th Annual Allerton Conference on Communication, Control and Computing*, 2006. 2, 5.2.2, 5.2.2, 5.2.2

[81] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Tenth International Workshop on Artificial Intelligence and Statistics*, 2005. 2.2

[82] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. URL `http://www.kyb.tuebingen.mpg.de/ssl-book`. 2.1, 2.3.1, 2.4, 2.6.1, 3.4.1

[83] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoy. A constant-factor approximation algorithm for the k-median problem. In *ACM Symposium on Theory of Computing*, 1999. 1.1.3, 4.1, 4.1.1

[84] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science*, pages 524–533, 2003. 1

[85] D. Cheng, R. Kannan, S. Vempala, and G. Wang. A divide-and-merge methodology for clustering. *ACM*

Trans. Database Syst., 31(4):1499–1525, 2006. 4.9

[86] D. Cohen, L. Atlas, and R. Ladner. Improving generalzation with active learning. *Machine Learning*, 15(2): 201–221, 1994. 1.1.1, 2.5.4, 7, 5, 5.1.1, 5.1.3, 5.2, 1, 5.2.1, 5.2.1

[87] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 189–196, 1999. 2.2

[88] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273 – 297, 1995. 1.1.2, 3.1

[89] N. Cristianini, J. Shawe-Taylor, Andre Elisseeff, and J. Kandola. On kernel target alignment. In *Advances in Neural Information Processing Systems*, 2001. 1.1.2, 3.1

[90] J. Czyzowicz, D. Mundici, and A. Pelc. Ulam's searching game with lies. *Journal of Combinatorial Theory, Series A*, 52:62–76, 1989. 5.1.1

[91] A. Dasgupta, J. Hopcroft, J. Kleinberg, and M. Sandler. On learning mixtures of heavy-tailed distributions. In *46th IEEE Symposium on Foundations of Computer Science*, 2005. 1.1.3, 4.1

[92] A. Dasgupta, J. E. Hopcroft, R. Kannan, and P. P. Mitra. Spectral clustering by recursive partitioning. In *ESA*, pages 256–267, 2006. 1, 4.1.1

[93] S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems*, 2004. 5.1.1

[94] S. Dasgupta. Coarse sample complexity bounds for active learning. In *Proceedings of the Nineteenth Annual Conference on Neural Information Processing Systems*, 2005. 1.1.1, 2.5.4, 5, 5.1.1, 5.1.1, 5.1.4, 5.3

[95] S. Dasgupta. Learning mixtures of gaussians. In *Fortieth Annual IEEE Symposium on Foundations of Computer Science*, 1999. 1.1.3, 4.1, 4.1.1

[96] S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss Lemma. *Random Structures & Algorithms*, 22(1):60–65, 2002. 6, 6.1, 6.4

[97] S. Dasgupta, M. L. Littman, and D. McAllester. Pac generalization bounds for co-training. In *Advances in Neural Information Processing Systems 14*, 2001. 2.3.2

[98] S. Dasgupta, A. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *Proceedings of the Eighteenth Annual Conference on Learning Theory*, 2005. 5, 5.1.1, 5.1.1, 5.1.4, 5.1.4, 6, 1, 5.2.1

[99] S. Dasgupta, D.J. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. *Advances in Neural Information Processing Systems*, 20, 2007. 2, 5.1.5, 5.1.6, 5.2

[100] W. Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. Approximation schemes for clustering problems. In *STOC*, 2003. 4.1.1

[101] E. Demaine, U. Feige, M.T. Hajiaghayi, and M. Salavatipour. Combination Can Be Hard: Approximability of the Unique Coverage Problem . In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, 2006. 7.5.3, 7.6.4

[102] L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation*. Springer-Verlag, 2001. 7.5.1

[103] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996. 1.1.3, 2.1.1, 2.3.1, 4.1.1, 4.1.2, 4.2, 4.6, 7.3.3, 7.5, A.1.1, A.1.1, A.1.1, A.3.1

[104] J. Dunagan and S. Vempala. Optimal outlier removal in high-dimensional spaces. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, 2001. 2.4.2, 2.4.2

[105] A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Inf. and Comput*, 82:246–261, 1989. 2.4.1

[106] A. Fiat, A. Goldberg, J. Hartline, and A. Karlin. Competitive Generalized Auctions. In *Proceedings 34th ACM Symposium on the Theory of Computing*, pages 72 – 81, 2002. 7.1

[107] P. Fische and S. Kwek. Minimizing Disagreement for Geometric Regions Using Dynamic Programming, with Applications to Machine Learning and Computer Graphics. 1996. 7.5.1

[108] A. Flaxman. Personal communication, 2003. 2.4.2

[109] J. Forster and H.-U. Simon. On the smallest possible dimension and the largest possible margin of linear arrangements representing given concept classes. *Theoretical Computer Science*, 350(1):40–48, 2006. 3.4, 3.4.3

[110] Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277 – 296, 1999. 3.1, 3.3.4

[111] Y. Freund and R.E. Schapire. Large margin classification using the Perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999. 6.4.1

[112] Y. Freund, H.S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997. 5, 5.1.1, 1

[113] A. Frieze and R. Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999. 1.1.3, 4, 4.1.2, 4.6

[114] K. Ganchev, J. Graca, J. Blitzer, and B. Taskar. Multi-view learning over structured and non-identical outputs. In *Proceedings of The 24th Conference on Uncertainty in Artificial Intelligence*, 2008. 2.6.1

[115] C. Gentile and M. K. Warmuth. Linear hinge loss and average margin. In *Proceedings of the 1998 conference on Advances in neural information processing systems*, 1988. 3.4.2

[116] R. Ghani. Combining labeled and unlabeled data for text classification with a large number of categories. In *Proceedings of the IEEE International Conference on Data Mining*, 2001. 2.2

[117] R. Ghani, R. Jones, and C. Rosenberg, editors. *The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*. Workshop ICML'03, 2003. 2.1

[118] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10(6):1455–1480, 1998. 3.4.2

[119] A. Goldberg and J. Hartline. Competitive Auctions for Multiple Digital Goods. In *Proceedings of the 9th Annual European Symposium on Algorithms*, pages 416 – 427, 2001. 7.1, 7.6, 5, 7.7

[120] A. Goldberg and J. Hartline. Competitiveness via Consensus. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms*, pages 215 – 222, 2003. 7.3

[121] A. Goldberg, J. Hartline, and A. Wright. Competitive Auctions and Digital Goods. In *Proceeding of the 12th ACM-SIAM Symposium on Discrete Algorithms*, pages 735–744, 2001. 1.1.4, 7.1, 7.3, 7.3, 4, 7.7

[122] A. Goldberg, J. Hartline, A. Karlin, M. Saks, and A. Wright. Competitive Auctions and Digital Goods. *Games and Economic Behavior*, 2002. Submitted for publication. An earlier version available as InterTrust Technical Report STAR-TR-99.09.01. 1

[123] A. Goldberg, J. Hartline, A. Karlin, and M. Saks. A Lower Bound on the Competitive Ratio of Truthful Auctions. In *Proceedings 21st Symposium on Theoretical Aspects of Computer Science*, pages 644–655, 2004. 9

[124] A. Goldberg, J. Hartline, A. Karlin, M. Saks, and A. Wright. Competitive Auctions and Digital Goods. *Games and Economic Behavior*, 2006. 1.1.4, 7.1, 7.3, 7.4

[125] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4): 792–807, 1986. 6.5

[126] A. Grigoriev, J. van Loon, R. Sitters, and M. Uetz. How to Sell a Graph: Guideliness for Graph Retailers. Meteor Research Memorandum RM/06/001, Maastricht University, 2005. 7.6.4

[127] V. Guigue, A. Rakotomamonjy, and S. Canu. Kernel basis pursuit. In *Proceedings of the 16th European Conference on Machine Learning (ECML'05)*, 2005. 3.4, 3.4.2

[128] S. R. Gunn and J. S. Kandola. Structural modelling with sparse kernels. *Mach. Learn.*, 48(1-3):137–163, 2002. ISSN 0885-6125. 3.4.2

[129] V. Guruswami, J. Hartline, A. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On Profit-Maximizing Envy-Free Pricing. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms*, pages 1164 – 1173,

2005. 1.1.4, 7.1, 7.2.4, 5, 7.6.4, 7.7

[130] S. Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th Annual International Conference on Machine Learning (ICML)*, 2007. 5.1.5, 5.1.5, 5.1.5, 5.1.5, 5.1.6

[131] J. Hartline and V. Koltun. Near-Optimal Pricing in Near-Linear Time. In *Proceedings of the 9th Workshop on Algorithms and Data Structures*, pages 422–431, 2005. 7.6.1, 5, 7.6.4, 7.7

[132] B. Heisele, P. Ho, and T. Poggio. Face recognition with support vector machines: Global versus component-based approach. In *International Conference on Computer Vision*, 2001. 1.1.2

[133] R. Herbrich. *Learning Kernel Classifiers*. MIT Press, Cambridge, 2002. 1.1.2, 3.1, 4.1, 4.1.3, 4.2, 4.4

[134] R. Hettich and K. O. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Rev.*, 35(3):380–429, 1993. 3.3.5

[135] R. Hwa, M. Osborne, A. Sarkar, and M. Steedman. Corrected co-training for statistical parsers. In *ICML-03 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, Washington D.C., 2003. 1.1.1

[136] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 604–613, 1998. 6.4

[137] J. Jackson. An efficient membership-query algorithm for learning dnf with respect to the uniform distribution. *Journal of Computer and System Sciences*, 57(3):414–440, 1995. 5.1.1, 1

[138] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *JACM*, 48(2):274 – 296, 2001. 1.1.3, 4.1, 4.1.1

[139] T. Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*. Kluwer, 2002. 1.1.2, 3.1

[140] T. Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2003. 2.3.2, 2.5.1

[141] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. ICML*, pages 200–209, 1999. 1.1.1, 2.1, 2.2, 2.5.4

[142] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in Modern Analysis and Probability*, pages 189–206, 1984. 6.1, 6.4

[143] M. Kaariainen. On active learning in the non-realizable case. In *ALT*, 2006. 5.1.1, 5.1.4, 5.1.4

[144] M. Kaariainen. Generalization error bounds using unlabeled data. In *Proceedings of the 18th Annual Conference on Learning Theory*, pages 127–142, 2005. 2.1, 2.5.4

[145] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *Proceedings of the 46th Annual Symposium on the Foundations of Computer Science*, 2005. 3.3.3, 5.2.3

[146] R. Kannan, S. Vempala, and A. Vetta. On clusterings: good, bad and spectral. *J. ACM*, 51(3):497–515, 2004. 1.1.3, 4.1, 4.1.1

[147] R. Kannan, H. Salmasian, and S. Vempala. The spectral method for general mixture models. In *Proc. 18th Annual Conference on Learning Theory*, 2005. 1.1.3, 4.1, 4.1.1

[148] M. Kearns. Efficient noise-tolerant learning from statistical queries. In *Journal of the ACM (JACM)*, pages 983 – 1006, 1998. 6

[149] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994. 2.1, 2.1.1, 2.3.1, 2.4.2, 6, 3.3, 7.1, 7.5

[150] J. Kleinberg. Detecting a network failure. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 231–239, 2000. 2.5.1

[151] J. Kleinberg. An impossibility theorem for clustering. In *NIPS*, 2002. 1.1.3, 4.1

[152] J. Kleinberg, M. Sandler, and A. Slivkins. Network failure detection and graph connectivity. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 231–239, 2004. 2.5.1

[153] A. R. Klivans, R. O'Donnell, and R. Servedio. Learning intersections and thresholds of halfspaces. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 177–186, 2002. 2.1

[154] D. E. Knuth. *The Art of Computer Programming*. Addison-Wesley, 1997. 2

[155] V. Koltchinskii. Rademacher Penalties and Structural Risk Minimization. *IEEE Transactions of Information Theory*, 54(3):1902–1914, 2001. 2.1.1, 2.3.1, 2.5.4

[156] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proc. ICML*, 2002. 4.1.3

[157] G. R. G. Lanckriet, N. Cristianini, P. L. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, (5):27–72, 2004. 1.1.2, 3.1, 3.3.4

[158] B. Leskez. The value of agreement: A new boosting algorithm. In *Proceedings of the Annual Conference on Computational Learning Theory*, 2005. 2.2, 2.5.4

[159] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *Proc. 9th Int. Conf. Computer Vision*, pages 626–633, 2003. 1.1.1, 2.1, 2.2

[160] L. Liao and W. S. Noble. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *Journal of Computational Biology*, 10(6):857–868, 2003. 3.2, 3.6

[161] A. Likhodedov and T. Sandholm. Approximating Revenue-Maximizing Combinatorial Auctions. In *The Twentieth National Conference on Artificial Intelligence (AAAI)*, pages 267–274, 2005. 5

[162] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, fourier transform, and learnability. In *Proceedings of the Thirtieth Annual Symposium on Foundations of Computer Science*, pages 574–579, Research Triangle Park, North Carolina, October 1989. 2.1, 3.4.3

[163] N. Littlestone. From online to batch learning. In *Proc. 2nd Annual ACM Conference on Computational Learning Theory*, pages 269–284, 1989. 3.4.2, 4.10

[164] N. Littlestone. Learning when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1987. 6

[165] P. M. Long. On the sample complexity of PAC learning halfspaces against the uniform distribution. *IEEE Transactions on Neural Networks*, 6(6):1556–1559, 1995. 5.1.4, 5.2.1

[166] R. Luss and A. d'Aspremont. Support vector machine classification with indefinite kernels. In *Advances in Neural Information Processing Systems*, 2007. 3.2

[167] Y. Mansour. Learning boolean functions via the fourier transform. In *Theoretical Advances in Neural Computation and Learning*, pages 391–424. 1994. 3.4.3

[168] D. McAllester. Simplified pac-bayesian margin bounds. In *Proceedings of the 16th Conference on Computational Learning Theory*, 2003. 3.2

[169] F. McSherry. Spectral parititioning of random graphs. In *Proc. 43rd Symp. Foundations of Computer Science*, pages 529–537, 2001. 1, 4.1.1

[170] S. Mendelson and P. Philips. Random subclass bounds. In *Proceedings of the 16th Annual Conference on Computational Learning Theory (COLT)*, 2003. 2.5.3

[171] R. Meyerson. Optimal Auction Design. *Mathematics of Opperations Research*, 6:58–73, 1983. 1.1.4, 7.1

[172] T. Mitchell. The discipline of machine learning. *CMU-ML-06 108*, 2006. 1, 1.1.1, 3.4.1

[173] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12:181 – 201, 2001. 1.1.2, 3.1

[174] R. Myerson. Optimal Auction Design. *Mathematics of Operations Research*, 6:58–73, 1981. 1

[175] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of Co-training. In *Proc. ACM CIKM Int. Conf. on Information and Knowledge Management*, pages 86–93, 2000. 2.2

[176] K. Nigam, A. McCallum, S. Thrun, and T.M. Mitchell. Text classification from labeled and unlabeled docu-

ments using EM. *Mach. Learning*, 39(2/3):103–134, 2000. 1.1.1, 2.1

[177] N. Nisan, T. Rougharden, E. Tardos, and V. Vazirani (Eds.). *Algorithmic Game Theory*. 2006. To appear. 1.1.4, 7.2.2

[178] N. Nissan. Personal communication, 2005. 7.6.1, 7.6.4

[179] E. E. Osuna and F. Girosi. Reducing the run-time complexity in support vector machines. In *Advances in kernel methods: support vector learning*, pages 271–283. 1999. ISBN 0-262-19416-3. 3.4.2

[180] S. Park and B. Zhang. Large scale unstructured document classification using unlabeled data and syntactic information. In *PAKDD 2003*, LNCS vol. 2637, pages 88–99. Springer, 2003. 2.2

[181] S.-B. Park and B.-T. Zhang. Co-trained support vector machines for large scale unstructured document classification using unlabeled data and syntactic information. *Information Processing and Management*, 40(3):421 – 439, 2004. 1.1.1

[182] D. Pierce and C. Cardie. Limitations of Co-Training for natural language learning from large datasets. In *Proc. Conference on Empirical Methods in NLP*, pages 1–9, 2001. 2.2

[183] J. Ratsaby and S. Venkatesh. Learning from a mixture of labeled and unlabeled examples with parametric side information. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, pages 412–417, 1995. 2.5.2

[184] D. Rosenberg and P. Bartlett. The Rademacher Complexity of Co-Regularized Kernel Classes. In *Proceedings of Artificial Intelligence & Statistics*, 2007. 2.2, 2.5.4, 2.6.1

[185] V. Roth. Sparse kernel regressors. In *ICANN '01: Proceedings of the International Conference on Artificial Neural Networks*, 2001. 3.4, 3.4.2

[186] B. Scholkopf and A. J. Smola. *Learning with kernels. Support Vector Machines, Regularization, Optimization, and Beyond*. MIT University Press, Cambridge, 2002. 3.2, 3.3

[187] B. Scholkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*. MIT Press, 2004. 1.1.2, 3.1, 4.1, 4.1.3, 4.2, 4.4

[188] R.E. Shapire. The strength of weak learnability. *Machine Learning*, (5):197–227, 1990. 3.3.2

[189] J. Shawe-Taylor. Rademacher Analysis and Multi-View Classification. 2006. http://www.gla.ac.uk/external/RSS/RSScomp/shawe-taylor.pdf. 2.2, 2.6.1

[190] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. 1.1.2, 3.1, 4.1, 4.1.3, 4.2, 4.4

[191] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998. 1.1.2, 2.1.3, 2.5, 2.5.3, 3.1, 4.4, 6, 6.1, 6.3, 6.4.1

[192] Y. Singer. Leveraged vector machines. In *Advances in Neural International Proceedings System 12*, 2000. 3.4, 3.4.2

[193] A. J. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans. *Advances in Large Margin Classifiers*. MIT Press, 2000. 1.1.2, 3.1

[194] N. Sokolovska, O. Capp, and F. Yvon. The asymptotics of semi-supervised learning in discriminative probabilistic models. In *Proceedings of the 25th International Conference on Machine Learning*, 2008. 2.5.4

[195] N. Srebro. How good is a kernel as a similarity function? In *Proc. 20th Annual Conference on Learning Theory*, 2007. 3.3, 3.4.1, 3.4.4, 3.4.4, 3.4.4, 4.4

[196] K. Sridharan and S. M. Kakade. An information theoretic framework for multi-view learning. In *Proceedings of the 21st Annual Conference on Learning Theory*, 2008. 2.2, 3, 2.6.1

[197] C. Swamy. Correlation clustering: Maximizing agreements via semidefinite programming. In *SODA*, 2004. 1

[198] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244,

2001. ISSN 1533-7928. 3.4, 3.4.2

[199] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 4:45–66, 2001. 1, 5.2

[200] A. Tsybakov. Optimal aggregation of classifiers in statistical learning. *Annals of Statistics*, 2004. 2, 5.2.2

[201] L.G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984. (document), 1, 2.1, 3.3, 4.1.1, 4.8.3

[202] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971. 5.1.2, 5.1.3

[203] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons Inc., 1998. (document), 1, 1.1.2, 1.1.3, 1.1.4, 2.1, 3.1, 4.1.1, 4.1.2, 4.2, 4.2, 4.4, 4.9, 4.10, 5.1.3, 7.1, 7.3.3, 7.5

[204] S. Vempala. A random sampling based algorithm for learning the intersection of half-spaces. In *Proceedings of the 38th Symposium on Foundations of Computer Science*, pages 508–513, 1997. 2.1

[205] S. Vempala and G. Wang. A spectral algorithm for learning mixture models. *J. Comp. Sys. Sci.*, 68(2): 841–860, 2004. 1.1.3, 4.1, 4.1.1

[206] K. A. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *COLT*, pages 314–326, 1990. 2.1

[207] P. Vincent and Y. Bengio. Kernel matching pursuit. *Machine Learning*, 48(1-3):165–187, 2002. 3.4.2

[208] L. Wang, C. Yang, and J. Feng. On learning with dissimilarity functions. In *Proceedings of the 24th international conference on Machine learning*, pages 991 – 998, 2007. 3.1, 3.6

[209] M. K. Warmuth and S. V. N. Vishwanathan. Leaving the span. In *Proceedings of the Annual Conference on Learning Theory*, 2005. 3.4

[210] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics*, pages 189–196, 1995. 2.1, 2.2

[211] T. Zhang. Covering number bounds of certain regularized linear function classes. *J. Mach. Learn. Res.*, 2: 527–550, 2002. ISSN 1533-7928. 3.4.2, 3.4.2

[212] T. Zhang. Regularized winnow methods. In *NIPS*, 2001. 6

[213] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schlkopf. Learning with local and global consistency. In *NIPS*, 2004. 2.5.1

[214] X. Zhu. Semi-Supervised Learning Literature Survey. 2006. Computer Sciences TR 1530 University of Wisconsin - Madison. 2.1

[215] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. ICML*, pages 912–912, 2003. 1.1.1, 2.1, 2.2, 2.5.1

[216] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning: From gaussian fields to gaussian processes. Technical report, Carnegie Mellon University, 2003. 2.5.1

Appendix A

Additional Proof and Known Results

A.1 Appendix for Chapter 2

A.1.1 Standard Results

We state in the following a few known generalization bounds and concentration results used in our proofs. We start with a classic result from [103].

Theorem A.1.1 *Suppose that $\mathcal{C}$ is a set of functions from X to $\{-1,1\}$ with finite VC-dimension $D \geq 1$. Let D be an arbitrary, but fixed probability distribution over $X \times \{-1,1\}$. For any ϵ, $\delta > 0$, if we draw a sample from D of size*

$$m(\epsilon, \delta, D) = \frac{64}{\epsilon^2}\left(2D\ln\left(\frac{12}{\epsilon}\right) + \ln\left(\frac{4}{\delta}\right)\right),$$

then with probability at least $1-\delta$, we have $\left|\mathrm{err}(h) - \hat{L}(h)\right| \leq \epsilon$ for all $f \in \mathcal{C}$.

We present now another classic results from [103].

Theorem A.1.2 *Suppose that $\mathcal{C}$ is a set of functions from X to $\{-1,1\}$ with finite VC-dimension $D \geq 1$. Let D be an arbitrary, but fixed probability distribution over $X \times \{-1,1\}$. Then*

$$\Pr_S\left[\sup_{f\in C, \hat{L}(f)=0}\left|\mathrm{err}(f) - \hat{L}(f)\right| \geq \epsilon\right] \quad \leq \quad 2\mathcal{C}[2m, D]e^{-m\epsilon/2}.$$

So, for any ϵ, $\delta > 0$, if we draw a sample from D of size

$$m \geq \frac{2}{\epsilon}\left(2\ln\left(\mathcal{C}[2m, D]\right) + \ln\left(\frac{2}{\delta}\right)\right),$$

then with probability at least $1-\delta$, we have that all functions with $\hat{L}(f) = 0$ satisfy $\mathrm{err}(f) \leq \epsilon$.

We present now another classic results from [103].

Theorem A.1.3 *Suppose that $\mathcal{C}$ is a set of functions from X to $\{-1,1\}$ with finite VC-dimension $D \geq 1$. Let D be an arbitrary, but fixed probability distribution over $X \times \{-1,1\}$. Then*

$$\Pr_S\left[\sup_{f\in C}\left|\mathrm{err}(f) - \hat{L}(f)\right| \geq \epsilon\right] \quad \leq \quad 8\mathcal{C}[2m, D]e^{-m\epsilon^2/8}.$$

So, for any ϵ, $\delta > 0$, if we draw from D a sample satisfying

$$m \geq \frac{8}{\epsilon^2}\left(\ln\left(\mathcal{C}[m, D]\right) + \ln\left(\frac{8}{\delta}\right)\right),$$

then with probability at least $1-\delta$ all functions f satisfy $\left|\mathrm{err}(f) - \hat{L}(f)\right| \geq \epsilon$.

We now state a result from [71].

Theorem A.1.4 *Suppose that $\mathcal{C}$ is a set of functions from X to $\{-1,1\}$. Let D be an arbitrary, but fixed probability distribution over $X \times \{-1,1\}$. Then for any target $f \in \mathcal{C}$ and for any i.i.d. sample of S of size m from D, let f_m be the function that minimizes the empirical error over S. Then for any $\delta > 0$, the probability that*

$$\mathrm{err}(f_m) \leq \hat{L}(f_m) + \sqrt{\frac{6 \ln \mathcal{C}[S]}{m}} + 4\sqrt{\frac{\ln(2/\delta)}{m}}$$

is greater than $1 - \delta$.

Note that in fact the above statement is true even if in the right handside we use $C[S']$ instead of $C[S]$ where S' is another i.i.d sample of size m drawn from D.

Theorem A.1.5 *For any class of functions we have:*

$$\Pr_S \left[\log_2(\mathcal{C}[S]) \geq \mathbf{E}[\log_2(\mathcal{C}[S])] + \alpha\right] \quad \leq \quad \exp\left[-\frac{\alpha^2}{2\mathbf{E}[\log_2(\mathcal{C}[S])] + 2\alpha/3}\right]. \tag{A.1.1}$$

Also,

$$\mathbf{E}[\log_2 \mathcal{C}[S]] \leq \log_2 \mathbf{E}[\mathcal{C}[S]] \leq \frac{1}{\ln 2}\mathbf{E}[\log_2 \mathcal{C}[S]]. \tag{A.1.2}$$

A.1.2 Additional Proofs

Theorem A.1.6 *For any class of functions we have:*

$$\Pr_S \left[\log_2(\mathcal{C}[S]) \geq 2 \log \mathbf{E}[\mathcal{C}[S]] + \alpha\right] \quad \leq \quad e^{-2\alpha}. \tag{A.1.3}$$

Proof: Inequality (A.1.1) implies that:

$$\Pr_S \left[\log_2(\mathcal{C}[S]) \geq 2\mathbf{E}[\log_2(\mathcal{C}[S])] + \alpha\right] \leq \exp\left[-\frac{(\alpha + \mathbf{E}[\log_2(\mathcal{C}[S])])^2}{2\mathbf{E}[\log_2(\mathcal{C}[S])] + 2(\mathbf{E}[\log_2(\mathcal{C}[S])] + \alpha)/3}\right].$$

Since $\frac{(\alpha+a)^2}{2a+2(a+\alpha)/3} \geq \frac{\alpha}{2}$ for any $a \geq 0$ we get

$$\Pr_S \left[\log_2(\mathcal{C}[S]) \geq 2\mathbf{E}[\log_2(\mathcal{C}[S])] + \alpha\right] \leq e^{-\alpha/2}.$$

Combining this together with the following fact (implied by Inequality (A.1.2))

$$\Pr_S \left[\log_2(\mathcal{C}[S]) \geq 2 \log \mathbf{E}[\mathcal{C}[S]] + \alpha\right] \leq \Pr_S \left[\log_2(\mathcal{C}[S]) \geq 2\mathbf{E}[\log_2(\mathcal{C}[S])] + \alpha\right],$$

we get the desired result. ■

A.2 Appendix for Chapter 5

Theorem A.2.1 *Let C be a set of functions from X to $\{-1,1\}$ with finite VC-dimension $D \geq 1$. Let P be an arbitrary, but fixed probability distribution over $X \times \{-1,1\}$. For any $\epsilon, \delta > 0$, if we draw a sample from P of size $N(\epsilon,\delta) = \frac{1}{\epsilon}\left(4D\log\left(\frac{1}{\epsilon}\right) + 2\log\left(\frac{2}{\delta}\right)\right)$, then with probability $1-\delta$, all hypotheses with error $\geq \epsilon$ are inconsistent with the data.*

A.2.1 Probability estimation in high dimensional ball

Consider $x = [x_1, \ldots, x_d] \sim P_x$ uniformly distributed on unit ball in R^d. Let A be an arbitrary set in R^2; we are interested in estimating the probability $\Pr_x((x_1, x_2) \in A)$. Let V_d be the volume of d-dimensional ball; we know

$$V_d = \pi^{d/2}/\Gamma(1 + d/2),$$

where Γ is the Gamma-function. In particular $V_{d-2}/V_d = d/(2\pi)$. It follows that

$$\begin{aligned}\Pr_x((x_1, x_2) \in A) &= \frac{V_{d-2}}{V_d} \int_{(x_1,x_2)\in A} (1 - x_1^2 - x_2^2)^{(d-2)/2} dx_1 dx_2 \\ &= \frac{d}{2\pi} \int_{(x_1,x_2)\in A} (1 - x_1^2 - x_2^2)^{(d-2)/2} dx_1 dx_2 \leq \frac{d}{2\pi} \int_{(x_1,x_2)\in A} e^{-(d-2)(x_1^2+x_2^2)/2} dx_1 dx_2,\end{aligned}$$

where we use the inequality $(1 - z) \leq e^{-z}$.

Lemma A.2.2 *Let $d \geq 2$ and let $x = [x_1, \ldots, x_d]$ be uniformly distributed in the d-dimensional unit ball. Given $\gamma_1 \in [0, 1]$, $\gamma_2 \in [0, 1]$, we have*

$$\Pr_x((x_1, x_2) \in [0, \gamma_1] \times [\gamma_2, 1]) \leq \frac{\gamma_1\sqrt{d}}{2\sqrt{\pi}} e^{-(d-2)\gamma_2^2/2}.$$

Proof: Let $A = [0, \gamma_1] \times [\gamma_2, 1]$. We have

$$\begin{aligned}\Pr_x((x_1, x_2) \in A) &\leq \frac{d}{2\pi} \int\limits_{(x_1,x_2)\in A} e^{-(d-2)(x_1^2+x_2^2)/2} dx_1 dx_2 \leq \frac{\gamma_1 d}{2\pi} \int\limits_{x_2\in[\gamma_2,1]} e^{-(d-2)x_2^2/2} dx_2 \\ &\leq \frac{\gamma_1 d}{2\pi} e^{-(d-2)\gamma_2^2/2} \int\limits_{x\in[0,1-\gamma_2)} e^{-(d-2)x^2/2} dx \leq \frac{\gamma_1 d}{2\pi} e^{-(d-2)\gamma_2^2/2} \min\left[1 - \gamma_2, \sqrt{\frac{\pi}{2(d-2)}}\right].\end{aligned}$$

Note that when $d \geq 2$, $\min(1, \sqrt{\pi/(2(d-2))}) \leq \sqrt{\pi/d}$. ■

Lemma A.2.3 *Assume $x = [x_1, \ldots, x_d]$ is uniformly distributed in the d-dimensional unit ball. Given $\gamma_1 \in [0, 1]$, we have*

$$\Pr_x(x_1 \geq \gamma_1) \leq \frac{1}{2} e^{-d\gamma_1^2/2}.$$

Proof: Let $A = [\gamma_1, 1] \times [-1, 1]$. Using a polar coordinate transform, we have:

$$\begin{aligned}\Pr_x((x_1, x_2) \in A) &= \frac{d}{2\pi} \int\limits_{(x_1,x_2)\in A} (1 - x_1^2 - x_2^2)^{\frac{d-2}{2}} dx_1 dx_2 \\ &= \frac{d}{2\pi} \int\limits_{(r,r\cos\theta)\in[0,1]\times[\gamma_1,1]} (1 - r^2)^{\frac{d-2}{2}} r dr d\theta = \frac{1}{2\pi} \int\limits_{(r,r\cos\theta)\in[0,1]\times[\gamma_1,1]} d\theta d(1 - r^2)^{\frac{d}{2}} \\ &\leq \frac{1}{2\pi} \int\limits_{(r,\theta)\in[\gamma_1,1]\times[-\pi/2,\pi/2]} d\theta d(1 - r^2)^{\frac{d}{2}} = 0.5(1 - \gamma_1^2)^{\frac{d}{2}}.\end{aligned}$$

Using inequality $(1 - z) \leq e^{-z}$, we obtain the desired bound. ■

Lemma A.2.4 *Let $d \geq 4$ and let $x = [x_1, \ldots, x_d]$ be uniformly distributed in the d-dimensional unit ball. Given $\gamma, \beta > 0$, we have:*

$$\Pr_x(x_1 \leq 0, x_1 + \beta x_2 \geq \gamma) \leq \frac{\beta}{2}(1 + \sqrt{-\ln \min(1, \beta)}) e^{-d\gamma^2/(4\beta^2)}.$$

Proof: Let $\alpha = \beta\sqrt{-2d^{-1}\ln\min(1,\beta)}$, we have

$$\begin{aligned}
\Pr_x(x_1 \le 0, x_1 + \beta x_2 \ge \gamma) &\le \Pr_x(x_1 \le -\alpha, x_1 + \beta x_2 \ge \gamma) + \Pr_x(x_1 \in [-\alpha, 0], x_1 + \beta x_2 \ge \gamma) \\
&\le \Pr_x(x_1 \le -\alpha, x_2 \ge (\alpha+\gamma)/\beta) + \Pr_x(x_1 \in [-\alpha, 0], x_2 \ge \gamma/\beta) \\
&\le \frac{1}{2}\Pr_x(x_2 \ge (\alpha+\gamma)/\beta) + \Pr_x(x_1 \in [0, \alpha], x_2 \ge \gamma/\beta) \\
&\le \frac{1}{4}e^{-d(\alpha+\gamma)^2/(2\beta^2)} + \frac{\alpha\sqrt{d}}{2\sqrt{\pi}}e^{-d\gamma^2/(4\beta^2)} \le \\
&\le \left[\frac{1}{4}e^{-\frac{d\alpha^2}{2\beta^2}} + \frac{\alpha\sqrt{d}}{2\sqrt{\pi}}\right]e^{-\frac{d\gamma^2}{4\beta^2}} = \left[\frac{\min(1,\beta)}{4} + \frac{\beta\sqrt{-2\ln\min(1,\beta)}}{2\sqrt{\pi}}\right]e^{-\frac{d\gamma^2}{4\beta^2}}.
\end{aligned}$$

■

Lemma A.2.5 *Let u and v be two unit vectors in R^d, and assume that $\theta(u,v) \le \tilde{\beta} < \pi/2$. Let $d \ge 4$ and let $x = [x_1, \ldots, x_d]$ be uniformly distributed in the d-dimensional unit ball. Consider $C > 0$ arbitrary, let*

$$\gamma = \frac{2\sin\tilde{\beta}}{\sqrt{d}}\sqrt{\ln C + \ln\left(1 + \sqrt{\ln\max(1, \cos\tilde{\beta}/\sin\tilde{\beta})}\right)}.$$

Then

$$\Pr_x\left[(u\cdot x)(w\cdot x) < 0, |w\cdot x| \ge \gamma\right] \le \frac{\sin\tilde{\beta}}{C\cos\tilde{\beta}}.$$

Proof: We rewrite the desired probability as

$$2\Pr_x\left[w\cdot x \ge \gamma, u\cdot x < 0\right].$$

W.l.g., let $u = (1,0,0,...,0)$ and $w = (\cos(\theta), \sin(\theta), 0, 0, ..., 0)$. For $x = [x_1, x_2, ..., x_d]$ we have $u\cdot x = x_1$ and $w\cdot x = \cos(\theta)x_1 + \sin(\theta)x_2$. Using this representation and Lemma A.2.4, we obtain

$$\begin{aligned}
\Pr_x\left[w\cdot x \ge \gamma, u\cdot x < 0\right] &= \Pr_x[\cos(\theta)x_1 + \sin(\theta)x_2 \ge \gamma, x_1 < 0] \\
&\le \Pr_x\left[x_1 + \frac{\sin(\tilde{\beta})}{\cos(\tilde{\beta})}x_2 \ge \frac{\gamma}{\cos(\tilde{\beta})}, x_1 < 0\right] \\
&\le \frac{\sin\tilde{\beta}}{2\cos\tilde{\beta}}\left(1 + \sqrt{\ln\max(1, \frac{\cos\tilde{\beta}}{\sin\tilde{\beta}})}\right)e^{-\frac{d\gamma^2}{4\sin^2\tilde{\beta}}} \\
&= \frac{\sin\tilde{\beta}}{2\cos\tilde{\beta}}C^{-1},
\end{aligned}$$

as desired. ■

A.3 Appendix for Chapter 7

A.3.1 Concentration Inequalities

Here is the McDiarmid inequality (see [103]) we use in our proofs:

Theorem A.3.1 *Let $Y_1, ..., Y_n$ be independent random variables taking values in some set A, and assume that $t : A^n \to R$ satisfies:*

$$\sup_{y_1,\ldots,y_n\in A, \overline{y}_i\in A}|t(y_1, ..., y_n) - t(y_1, ..., y_{i-1}, \overline{y}_i, y_{i+1}, y_n)| \le c_i,$$

for all i, $1 \leq i \leq n$. Then for all $\gamma > 0$ we have:

$$\Pr\left\{|t(Y_1, ..., Y_n) - \mathbf{E}[t(Y_1, ..., Y_n)]| \geq \gamma\right\} \leq 2e^{-2\gamma^2 / \sum_{i=1}^{n} c_i^2}$$

Here is also a consequence of the Chernoff bound that we used in Lemma 7.4.4.

Theorem A.3.2 *Let $X_1, ..., X_n$ be independent Poisson trials such that, for $1 \leq i \leq n$, $\Pr[X_i = 1] = \frac{1}{2}$ and let $X = \sum_{i=1}^{n} X_i$. Then any n' we have:*

$$\Pr\left\{\left|X - \frac{n}{2}\right| \geq \epsilon \max\{n, n'\}\right\} \leq 2e^{-2n'\epsilon^2}$$