A Theory of Similarity Functions for Clustering

Maria-Florina Balcan[*] **Avrim Blum**[*] **Santosh Vempala**[†]

July 2007
CMU-CS-07-143

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[*] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. {ninamf,avrim}@cs.cmu.edu
[†]College of Computing, Georgia Institute of Technology. vempala@cc.gatech.edu.

Research supported in part by NSF grants CCR-0105488, CCR-0122581, and IIS-0121678.

Keywords: Clustering Framework, Similarity Functions, List Clustering, Tree Clustering, Clustering Complexity, Linkage Based Algorithms.

Abstract

Problems of clustering data from pairwise similarity information are ubiquitous in Computer Science. Theoretical treatments typically 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 true goal is to cluster the points correctly rather than to optimize any specific graph property. In this work, we initiate a theoretical study of the design of similarity functions for clustering from this perspective. In particular, motivated by recent work in learning theory that asks "what natural properties of a similarity 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?"

We develop a notion of the *clustering complexity* of a given property (analogous to notions of *capacity* in learning theory), that characterizes its information-theoretic usefulness for clustering. We then analyze this complexity for several natural game-theoretic and learning-theoretic properties, as well as design efficient algorithms that are able to take advantage of them. We consider two natural clustering objectives: (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 desired clustering is some pruning of this tree (which a user could navigate). 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 [18] and [4].

1 Introduction

Clustering problems arise in many different fields, from data mining to computer vision to VLSI design to computational biology. In the Algorithms literature, clustering is typically studied by posing some objective function, suck 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 [13, 25, 21, 22]. That is, the graph is viewed as "ground truth" and then one considers algorithms to optimize various objectives on this data. On the other hand, for 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 a heuristic based on some knowledge of the general problem: for example, cosine-similarity for clustering documents or a Smith-Waterman score in computational biology. In other words, in many real-world applications the goal is really to produce a clustering that gets the data correct, not necessarily to optimize some specific graph property.

In this work, we imagine a domain expert with a large set of data that she would like to cluster (perhaps documents clustered by topic, or proteins clustered by function). Based on the task at hand, the domain expert comes up with a pairwise similarity function $\mathcal{K}$ that is related to the desired goal. If this function were extremely good, say $\mathcal{K}(x, y) > 1/2$ for all pairs x and y that should be in the same cluster, and $\mathcal{K}(x, y) < 1/2$ for all pairs x and y that should be in different clusters, then she wouldn't need the help of an additional clustering algorithm: she could just use it to assign clusters directly.[1] However, what if she cannot construct a similarity function that is *that* good: what natural and much weaker properties would be sufficient for it to still be useful for producing a good clustering? What kind of advice can we provide to the domain expert in terms of *desiderata* for a similarity function?[2] Moreover, given a property she believes her similarity function has with respect to the ground truth, what *algorithms* would guarantee a low-error solution? In particular, motivated by work on learning with kernel functions that asks "what natural properties of a given kernel (or similarity) function are sufficient to allow one to *learn* well?" [7, 8, 36, 34, 20, 23, 1] we ask the question "what natural properties of a similarity function are sufficient to allow one to *cluster* well?" Our approach can be thought of as defining a *PAC model for clustering*, though the basic object of study, rather than a concept class, is a *property* that effectively can be viewed as a set of (concept, distribution) pairs. We expand on this further in Section 1.1.

The main reason there has not been so much work in this direction is that if one defines success as outputting a close approximation to the correct clustering, then one needs *very strong* conditions to guarantee this will occur. For example, suppose we weaken the above condition to simply require that all points x are more similar to all points y from their cluster than to any points y from any other clusters. That is, for any data point x, if we sort the data points y by decreasing similarity to x, then x's cluster is some prefix of this ordering. This is still a strong condition and yet it is not sufficient to guarantee one can produce even a good approximation to the correct answer. For instance, in the example in Figure 1, there are multiple clusterings consistent with this property (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 [26] 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.

In our work we get around 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 to alternatively allow the clustering algorithm to produce a *tree* (a hierarchical clustering) such that the correct answer is

[1] Correlation Clustering can be viewed as a relaxation that allows some pairs to fail to satisfy this condition, and the algorithms of [10, 14, 38, 3] shows that this is sufficient to cluster well if the number of pairs that fail is small. *Planted partition* models [5, 30, 15] allow for many failures so long as they occur at *random*. We will be interested in much more drastic relaxations, however.

[2] Of course, given some clustering algorithm $\mathcal{A}$, one could simply tell the domain expert to create a graph structure such that algorithm $\mathcal{A}$ will find the correct answer, but this is not particularly enlightening.

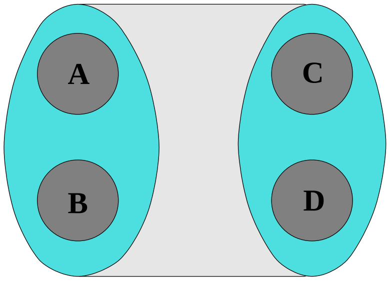

Figure 1: Suppose that $\mathcal{K}(x, y) = 1$ if x and y belong to the same dark shaded region (A, B, C, or D), $\mathcal{K}(x, y) = 1/2$ if $x \in A$ and $y \in B$ or if $x \in C$ and $y \in D$, and $\mathcal{K}(x, y) = 0$ otherwise. Even assuming that all points are more similar to other points in their own cluster than to any point in any other cluster, there are still multiple consistent clusterings, including two consistent 3-clusterings ($\{A, B, C \cup D\}$ and $\{A \cup B, C, D\}$). However, there is a single hierarchical decomposition such that any consistent clustering is a pruning of this tree.

(approximately) some pruning of this tree. For instance, the example in Figure 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, given a hierarchical clustering of web-pages, a user could start at the top (a single cluster with all objects in it) and then "click" on any cluster that is too broad to break it apart into its children in the tree (in fact, Yahoo directories are organized this way). 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.

1.1 Perspective

There has been significant work both in machine learning and theoretical computer science on defining and comparing notions of what it means to produce a *good clustering* of a given set of data points – e.g, of a given weighted graph or of a given set of points in R^n. That work is primarily focused on the objective function, for example presenting graphs or sets of points in R^n for which one objective produces better-looking results than another [12, 31, 32, 25], or on issues such as the stability of clustering algorithms [33, 9]. In this work we flip the perspective around, viewing the problem as effectively one of learning from unlabeled data via similarity functions. That is, our goal is to achieve *low true error* (an approximation to the correct clustering) and we then ask what natural properties might we want a pairwise similarity function to satisfy that would allow us to get at this ground truth, either by producing a tree such that some pruning is approximately correct or through a small list of candidates.

Our approach can also be viewed as developing a *PAC model for clustering*. In the PAC model for learning [39], 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 the *property*, which can be viewed as a set of concept-distribution pairs (i.e., the pairs for which the data and target concept 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 the more common 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 learning of that type (e.g., when does some heuristic like ID3 work well), our interest is in understanding which properties are sufficient for clustering, and then ideally the simplest algorithm to cluster given that property.

1.2 Connections to other related work

There has been substantial and continuing work in recent years in the machine learning community analyzing the use of pairwise similarity functions (especially kernel functions) for learning [1, 7, 8, 20, 23, 36, 34].

Much of this work examines properties that allow a similarity function to be useful for learning from *labeled* examples. The clustering problem is more difficult because even in the relaxations we consider, the forms of feedback allowed are much weaker.

There has also been significant work both in the algorithms and in the machine learning community on learning mixtures of distributions for the case that examples lie in R^n [2, 6, 24, 41, 16]. This 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: what properties of an embedding into R^n would allow a point set to cluster well? However, unlike our focus, the types of assumptions made are distributional and in that sense are much stronger than the types of properties we will be considering. Abstractly speaking, this view of clustering parallels the "generative" classification setting [17], while the framework we propose parallels the "discriminative" classification setting (i.e. the PAC model of Valiant [39] and the Statistical Learning Theory framework of Vapnik [40]).

Another line of research related to ours is work on *planted* partitions in graphs [5, 30, 15]. This work also has a notion of a "correct answer", but as with the distributional models above, makes strong probabilistic assumptions about the similarity function.

1.3 Our Results

Broadly speaking, we provide a general unified framework for analyzing what properties of a similarity function are sufficient to allow it to be useful for clustering, under different levels of relaxation of the clustering objective. We illustrate our framework by analyzing several natural game-theoretic and learning-theoretic classes of properties. Specifically:

- We consider a family of stability-based properties. For example, we show that a natural generalization of the "stable marriage" property (no two subsets $A \subset C$, $A' \subset C'$ of clusters C, C' in the correct clustering are both more similar on average to each other than to the rest of their own clusters) is sufficient to produce a hierarchical clustering via a natural average-linkage algorithm (Theorems 6 and 9). Moreover, a significantly weaker notion of stability is also sufficient to produce a hierarchical clustering, but requires a more involved algorithm (Theorem 8).

- We show that a weaker "average-attraction" property (which we show is provably not enough to produce a single correct tree) is sufficient to produce a small list of clusterings (Theorem 3), and give several generalizations to even weaker conditions that generalize the notion of large-margin kernel functions, using and extending recent results in learning theory (Theorem 5).

- 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 [11, 17, 40] and it provides a formal measure of the inherent usefulness of a similarity function property.

- We also show how these 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 delicate arguments using regularity-type results of [18] and [4].

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.

2 Notation, 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 $\ell(x)$ 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 $err(h) = \min_{\sigma \in \mathcal{S}_k} [\Pr_{x \in S} [\sigma(h(x)) \neq \ell(x)]]$. 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 $\mathcal{K}(x, x')$ that given two examples outputs a number in the range $[-1, 1]$.[3] We will say that $\mathcal{K}$ is a symmetric similarity function if $\mathcal{K}(x, x') = \mathcal{K}(x', x)$ for all x, x'.

Our goal is to develop a set of natural properties sufficient for a similarity function $\mathcal{K}$ to be *good* for a clustering problem (S, ℓ) that (ideally) are intuitive, broad, and imply that such a similarity function results in the ability to *cluster well*. As mentioned above, 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 (human) expert or from an oracle. 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. It effectively corresponds to a clustering algorithm saying "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 them for you." Note that any given tree has at most 2^{2k} prunings of size k [27], so this model is at least as strict as the list model.

Transductive vs Inductive. Clustering is typically posed as a "transductive" 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. This is closely connected to the notion of sample complexity in learning [40], as well as the notion of property testing [19]. For clarity, for most of this paper we will focus on the transductive setting. In Appendix B we show how our algorithms can be adapted to the inductive setting.

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 $C_{\ell(x)}$ to which point x belongs. For $A \subseteq X, B \subseteq X$, let $\mathcal{K}(A, B) = \mathbf{E}_{x \in A, x' \in B}[\mathcal{K}(x, x')]$. We call this the *average attraction* of A to B. Let $\mathcal{K}_{max}(A, B) = \max_{x \in A, x' \in B} \mathcal{K}(x, x')$; we call this *maximum attraction* of A to B. Given two clusterings g and h we define the distance $d(g, h) = \min_{\sigma \in \mathcal{S}_k} [\Pr_{x \in S} [\sigma(h(x)) \neq g(x)]]$.

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 ordering property. A 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, $\mathcal{K}(x, C(x) - \{x\}) > \mathcal{K}(x, C_i)$ for all $C_i \neq C(x)$. We will also consider intermediate "stability" properties

[3] That is, the input to the clustering algorithm is just a weighted graph. However, we still want to conceptually view $\mathcal{K}$ as a *function* over abstract objects, much like the notion of a kernel function in learning theory.

such as that for any two clusters C_i, C_j, for any $A \subset C_i, B \subset C_j$ we have $\mathcal{K}(A, C_i - A) > \mathcal{K}(A, B)$. 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 1 *Given a property $\mathcal{P}$ and similarity function $\mathcal{K}$, define the (ϵ, k)-***clustering complexity** *of the pair $(\mathcal{P}, \mathcal{K})$ 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.*[4] *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 $\mathcal{K}$.*

In the following sections we analyze the clustering complexity of several natural properties and provide efficient algorithms to take advantage of such functions. To illustrate the definitions we start by analyzing the strict ordering property in Section 3. We then analyze a much weaker average attraction property in Section 4 that has close connections to large margin properties studied in Learning Theory [1, 7, 8, 20, 23, 36, 34]. 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 5.

3 A simple property: strict ordering

To illustrate the setup we begin with the simple strict ordering property mentioned in the introduction.

Property 1 *The similarity function $\mathcal{K}$ satisfies the* **strict ordering** *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 ordering property, we can efficiently construct a tree such that the ground-truth clustering is a pruning of this tree (Theorem 2). As mentioned earlier, 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 1 *For $\epsilon < \frac{1}{2k}$, the strict ordering property has (ϵ, k)-clustering complexity at least $2^{k/2}$.*

Proof: The similarity function is a generalization of the picture in Figure 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 $\mathcal{K}(x, x') = 1$ if x and x' belong to the same R_i, $\mathcal{K}(x, x') = 1/2$ if x and x' belong to two subsets in the same pair, and $\mathcal{K}(x, x') = 0$ otherwise. Notice that in this setting 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 2 *Let $\mathcal{K}$ be a similarity function satisfying the strict ordering property. Then we can efficiently construct a tree such that the ground-truth clustering is a pruning of this tree.*

Proof: If $\mathcal{K}$ 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 $\mathcal{K}_{max}(C, C')$. This maintains the invariant that at each step the current clustering is laminar with respect to the ground-truth. Specifically, if the algorithm merges two clusters C and C', and C is strictly

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

contained in some cluster C_r of the ground truth, then by the strict ordering property we must have $C' \subset C_r$ as well. If $\mathcal{K}$ is not symmetric, then single linkage may fail.[5] However, in this case, the following "Boruvka-ish" algorithm can be used. Starting with n clusters of size 1, draw a directed edge from each cluster C to the cluster C' maximizing $\mathcal{K}_{max}(C, C')$. 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 ordering 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 ordering property is quite strong, a similarity function satisfying this property can still fool a top-down spectral clustering approach. See Figure 2 in Appendix C.

4 A weaker property: average attraction

A much weaker property to ask of a similarity function is just that most points are noticeably more similar on average to other points in their own cluster than to points in any other cluster. Specifically, we define:

Property 2 *A similarity function $\mathcal{K}$ satisfies the (ν, γ)-***average attraction** *property for the clustering problem (S, ℓ) if a $1 - \nu$ fraction of examples x satisfy:*

$$\mathcal{K}(x, C(x)) \geq \mathcal{K}(x, C_i) + \gamma \text{ for all } i \in Y, i \neq \ell(x).$$

This is a fairly natural property to ask of a similarity function. In addition, it also has a game-theoretic interpretation: if one thinks of the data points as players in a game in which they each choose their own label with payoff equal to their average attraction to others of the same label, then for $\nu = 0$ this property says that the similarity function should be such that the correct clustering is a γ-strict Nash equilibrium (a Nash equilibrium where each player has at least γ-disincentive to deviate).

The following is a simple clustering algorithm that given a similarity function $\mathcal{K}$ satisfying the average attraction property produces a list of clusterings of size that depends only on ϵ, k, and γ. Specifically,

Algorithm 1 Sampling Based Algorithm, List Model

```
    Input: Data set $S$, similarity function $\mathcal{K}$, parameters $\gamma, \epsilon > 0$, $k \in Z^+$; $N(\epsilon, \gamma, k)$, $s(\epsilon, \gamma, k)$.

  • Set $\mathcal{L} = \emptyset$.

  • Repeat $N(\epsilon, \gamma, k)$ times
      For $k' = 1, \ldots, k$ do
          Pick a set ${R_S}^{k'}$ of random examples from $S$ of size $s(\epsilon, \gamma, k)$.
          Let $h$ be the average-nearest neighbor hypothesis induced by the sets ${R_S}^i$, $i = 1, \ldots, k'$.
          That is, for any point $x \in S$, define $h(x) = \text{argmax}_{i \in \{1, \ldots k'\}}[\mathcal{K}(x, {R_S}^i)]$. Add $h$ to $\mathcal{L}$.

  • Output the list $\mathcal{L}$.
```

Theorem 3 *Let $\mathcal{K}$ be a similarity function satisfying the (ν, γ)-average attraction property for the clustering problem (S, ℓ). Using Algorithm 1 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*

[5] Consider 3 points x, y, and z where the correct clustering is $\{\{x\}, \{y, z\}\}$. If $\mathcal{K}(x, y) = 1$, $\mathcal{K}(y, z) = \mathcal{K}(z, y) = 1/2$, and $\mathcal{K}(y, x) = \mathcal{K}(z, x) = 0$, then this is consistent with strict ordering and yet the algorithm will incorrectly merge x and y in its first step.

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: See Appendix A. ■

Note that Theorem 3 immediately implies a corresponding upper bound on the (ϵ, k)-clustering complexity of the $(\epsilon/2, \gamma)$-average attraction property. 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 *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 $\mathcal{K}(x, x') = 1$ if x and x' belong to the same region R_i and $\mathcal{K}(x, x') = 0$, otherwise. Notice that in this setting all the k-way partitions of the set $\{R_1, \ldots, R_{1/\gamma}\}$ are consistent with Property 2 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.

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 $\mathcal{K}$ being a legal kernel function with the large margin property [7, 37, 40, 35].

Property 3 *A similarity function* $\mathcal{K}$ *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:*

$$\mathbf{E}_{x' \in C(x)}[w(x')\mathcal{K}(x, x')] \geq \mathcal{K}_{x' \in C_r}[w(x')\mathcal{K}(x, x')] + \gamma \text{ for all } r \in Y, r \neq \ell(x).$$

Property 3 can, for instance, model a natural k-median style property, where we ask that each cluster contain a non-negligible α fraction of plausible cluster centers (points x' of weight 1) such that each data point is at least β more similar to its own cluster centers than to those of any other cluster (in this case, $\gamma = \alpha\beta$).

If we have $\mathcal{K}$ 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 which we present in Appendix A. Formally we can show that:

Theorem 5 *Let* $\mathcal{K}$ *be a similarity function satisfying the* (ν, γ)*-average weighted attraction property for the clustering problem* (S, ℓ)*. Using Algorithm 4 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 5 to Appendix A. While the proof follows from ideas in [7] (in the context of classification), we are able to get substantially better bounds by a more careful analysis and by taking advantage of attribute-efficient learning algorithms with good L_1-margin guarantees [28, 42].

4.1 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 $\mathcal{K}(C_i, C_i) \geq \mathcal{K}(C_i, C_j) + \gamma$; 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 $\mathcal{K}(x, x') = 1$ for x, x' in the same region and $\mathcal{K}(x, x') = 0$ 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.

5 Stability-based Properties

The properties in Section 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 3 and 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 are natural analogs of the "stable-marriage" property to clustering:

Property 4 *The similarity function $\mathcal{K}$ 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 $A \subset C_r$, for all $A' \subseteq C_{r'}$ we have*

$$\mathcal{K}(A, C_r \setminus A) > \mathcal{K}(A, A').$$

Property 5 *The similarity function $\mathcal{K}$ satisfies the* **weak stability** *property for the clustering problem (S, ℓ) if for all C_r, $C_{r'}$, $r \neq r'$, for all $A \subseteq C_r$, $A' \subseteq C_{r'}$, we have:*

- *If $A \subset C_r$ and $A' \subset C_{r'}$ then either $\mathcal{K}(A, C_r \setminus A) > \mathcal{K}(A, A')$ or $\mathcal{K}(A', C_{r'} \setminus A') > \mathcal{K}(A', A)$.*
- *If $A = C_r$ then $\mathcal{K}(A', C_{r'} \setminus A') > \mathcal{K}(A', A)$.*
- *If $A' = C_{r'}$ then $\mathcal{K}(A, C_r \setminus A) > \mathcal{K}(A, A')$.*

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 stable-marriage condition. Strong stability asks that *both* be more attracted to their true clusters. Note that if we take the example from Figure 1 and set a small fraction of the edges inside each dark-shaded region to 0, then with high probability this would still satisfy strong stability with respect to the natural clusters even though it no longer satisfies strict ordering. We show now that strong stability is sufficient to produce a hierarchical clustering and leave the proof for weak stability to Appendix A (see Theorem 9).

Theorem 6 *Let $\mathcal{K}$ be a symmetric similarity function satisfying Property 4. Then we can efficiently construct a binary tree such that the ground-truth clustering is a pruning of this tree.*

Proof Sketch: We will show that Algorithm 2 (Average Linkage) will produce the desired result. Note that the algorithm uses $\mathcal{K}(C, C')$ rather than $\mathcal{K}_{max}(C, C')$ as in single linkage; in fact in Figure 3 (Appendix C) 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

Algorithm 2 Average Linkage, Tree Model

```
Input: Data set $S$, similarity function $\mathcal{K}$.
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.
```

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 4, $\mathcal{K}(C', C_r - C') > \mathcal{K}(C', x)$ for all $x \notin C_r$, and so in particular we have $\mathcal{K}(C', C_r - C') > \mathcal{K}(C', C)$. Furthermore, $\mathcal{K}(C', C_r - C')$ is a weighted average of the $\mathcal{K}(C', C'')$ over the sets $C'' \subseteq C_r - C'$ in our current clustering and so at least one such C'' must satisfy $\mathcal{K}(C', C'') > \mathcal{K}(C', C)$. However, this contradicts the specification of the algorithm, since by definition it merges the pair C, C' such that $\mathcal{K}(C', C)$ is greatest. ■

While natural, Properties 4 and 5 are still somewhat brittle: in the example of Figure 1, for instance, if one adds a small number of edges with similarity 1 going left to right then the properties are no longer satisfied for the natural clusters (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 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 6 *The similarity function $\mathcal{K}$ 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*

$$\mathcal{K}(A, C_r \setminus A) > \mathcal{K}(A, A') + \gamma.$$

The idea of how we can use this property is we will first run an algorithm for the list model much like Algorithm 1, 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 f-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 3, below.

Theorem 7 *Let $\mathcal{K}$ be a similarity function satisfying the (s, γ)-strong stability of large subsets property 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 using Algorithm 3 with parameters satisfying $s + f \leq g$, $f \leq g\gamma/10$ and $\alpha > 5g + 2f$ we get a tree such that the ground-truth clustering is αk-close to a pruning of this tree.*

Proof Sketch: Let k' be the number of "big" ground-truth clusters, i.e. the clusters of size at least αn; without loss of generality assume that C_1, ..., $C_{k'}$ are the big clusters.

Algorithm 3 Testing Based Algorithm, Tree Model.

Input: Data set S, similarity function $\mathcal{K}$, parameters $\gamma > 0$, $k \in Z^+$, $f, g, s, \alpha > 0$. A list of clusters $\mathcal{L}$ with the property that any cluster C in the ground-truth is f-close to one of them.

Output: A tree on subsets.

- Throw out all clusters of size at most αn. For every pair of clusters C_r, $C_{r'}$ in our list $\mathcal{L}$ of clusters that are sufficiently "non-laminar" with respect to each other: i.e. we have $|C_r \setminus C_{r'}| \geq gn$, $|C_{r'} \setminus C_r| \geq gn$ and $|C_r \cap C_{r'}| \geq gn$, compute $\mathcal{K}(C_r \cap C_{r'}, C_r \setminus C_{r'})$ and $\mathcal{K}(C_r \cap C_{r'}, C_{r'} \setminus C_r)$. Throw out whichever one does worse: i.e., throw out C_r if the first similarity is smaller; throw out $C_{r'}$ is the second similarity is smaller. Let $\mathcal{L}'$ be the remaining list of clusters at the end of the process.
- 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_r, $C_{r'}$ are approximately equal if $|C_r \setminus C_{r'}| \leq gn$, $|C_{r'} \setminus C_r| \leq gn$ and $|C_{r'} \cap C_r| \geq gn$. Let $\mathcal{L}''$ be the list remaining.
- Construct a forest on the remaining list $\mathcal{L}''$. C_r becomes a child of $C_{r'}$ in this forest if $C_{r'}$ approximately contains C_r, i.e. $|C_r \setminus C_{r'}| \leq gn$, $|C_{r'} \setminus C_r| \geq gn$ and $|C_{r'} \cap C_r| \geq gn$.
- Complete the forest arbitrarily into a tree.

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 6 and Lemma 1 (stated below), we know that after Step 1 (the "testing of clusters" step) all the clusters C'_1, ...,$C'_{k'}$ survive; furthermore, one can show we have three types of relations between the remaining clusters. Specifically:

(a) C_r and $C_{r'}$ are approximately equal; this happens if $|C_r \setminus C_{r'}| \leq gn$, $|C_{r'} \setminus C_r| \leq gn$ and $|C_{r'} \cap C_r| \geq gn$.

(b) C_r and $C_{r'}$ are approximately disjoint; this happens if $|C_r \setminus C_{r'}| \geq gn$, $|C_{r'} \setminus C_r| \geq gn$ and $|C_{r'} \cap C_r| \leq gn$.

(c) $C_{r'}$ approximately contains C_r; this happens if $|C_r \setminus C_{r'}| \leq gn$, $|C_{r'} \setminus C_r| \geq gn$ and $|C_{r'} \cap C_r| \geq gn$.

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", and 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. So, $C''_1, ..., C''_{k'}$ indeed forms a pruning of this tree. This then implies that the ground-truth is $\alpha \cdot k$-close to a pruning of this tree. ■

Lemma 1 *Let $\mathcal{K}$ 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 $\mathcal{K}(I, C \setminus I) > \mathcal{K}(I, C' \setminus I)$.*

Proof: See Appendix A. ■

Theorem 8 *Let $\mathcal{K}$ 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 3 with parameters $\alpha = O(\epsilon/k)$, $g = O(\epsilon^2/k^2)$, $f = O(\epsilon^2\gamma/k^2)$, together with Algorithm 1 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 $O(k/\epsilon)$.*

Proof Sketch: First, we run Algorithm 1 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 running Procedure 3 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 7). To complete the proof we only need to show that this tree has $O(\epsilon/k)$ 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 2). ■

To better understand the specifics of our properties and of the linkage-based algorithms, we present a few interesting examples in Appendix C.

6 Inductive Setting

Our algorithms can also be extended to 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 view each node in the tree as inducing 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.

Our specific results for this model appear in Appendix B. While most of our analyses can be adapted in a reasonably direct way, adapting the average-linkage algorithm for the strong stability property while maintaining its computational efficiency is substantially more involved, as it requires showing that sampling preserves the stability property. See Theorem 11.

7 Conclusions and Open Questions

In this paper we provide a generic framework for analyzing what properties of a similarity function are sufficient to allow it to be useful for clustering, under different levels of relaxation 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 web-pages by topic) and in terms of advice about what *algorithms* to use given certain beliefs about the relation of the similarity function to the clustering task. Abstractly speaking, our notion of a *property* parallels that of a data-dependent concept class [40] (such as large-margin separators) in the context of classification.

Our work also provides the first formal framework for analyzing clustering with limited (non-interactive) feedback. A concrete implication of our work is 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.

Open questions: It would be interesting to further explore and 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 2) 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 6), our algorithm produces hierarchy but has 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? More generally, it would be interesting to determine whether these stability properties can be further weakened and still admit a hierarchical clustering.

References

[1] *http://www.kernel-machines.org/.*

[2] D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *Proceedings of the Eighteenth Annual Conference on Learning Theory*, 2005.

[3] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. In *STOC*, pages 684–693, 2005.

[4] 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.

[5] N. Alon and N. Kahale. A spectral technique for coloring random 3-colorable graphs. *SIAM Journal on Computing*, 26(6):1733 – 1748, 1997.

[6] S. Arora and R. Kannan. Learning mixtures of arbitrary gaussians. In *ACM Symposium on Theory of Computing*, 2005.

[7] M.-F. Balcan and A. Blum. On a theory of learning with similarity functions. In *International Conference on Machine Learning*, 2006.

[8] M.-F. Balcan, A. Blum, and S. Vempala. On kernels, margins and low-dimensional mappings. *Machine Learning Journal*, 2006.

[9] S. Ben-David, U. von Luxburg, and D. Pal. A sober look at stability of clustering. In *Proceedings of the Annual Conference on Computational Learning Theory*, 2006.

[10] A. Blum, N. Bansal, and S. Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004.

[11] S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: a survey of recent advances. *ESAIM: Probability and Statistics*, 9:9:323–375, 2005.

[12] M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 1999.

[13] 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.

[14] 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.

[15] A. Dasgupta, J. E. Hopcroft, R. Kannan, and P. P. Mitra. Spectral clustering by recursive partitioning. In *ESA*, pages 256–267, 2006.

[16] S. Dasgupta. Learning mixtures of gaussians. In *Fortieth Annual IEEE Symposium on Foundations of Computer Science*, 1999.

[17] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.

[18] A. Frieze and R. Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.

[19] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653 – 750, 1998.

[20] R. Herbrich. *Learning Kernel Classifiers*. MIT Press, Cambridge, 2002.

[21] P. Indyk. Sublinear time algorithms for metric space problems. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, 1999.

[22] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of ACM*, 48(2):274 – 296, 2001.

[23] T. Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*. Kluwer, 2002.

[24] R. Kannan, H. Salmasian, and S. Vempala. The spectral method for general mixture models. In *Proceedings of the Eighteenth Annual Conference on Learning Theory*, 2005.

[25] R. Kannan, S. Vempala, and A. Vetta. On clusterings: good, bad and spectral. *J. ACM*, 51(3):497–515, 2004.

[26] J. Kleinberg. An impossibility theorem for clustering. In *NIPS*, 2002.

[27] D. E. Knuth. *The Art of Computer Programming*. Addison-Wesley, 1997.

[28] N. Littlestone. Learning when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1987.

[29] N. Littlestone. From online to batch learning. In *Proc. 2nd Annual ACM Conference on Computational Learning Theory*, pages 269–284, 1989.

[30] F. McSherry. Spectral parititioning of random graphs. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 529–537, 2001.

[31] M. Meila. Comparing clusterings by the variation of information. In *Proceedings of the Annual Conference on Computational Learning Theory*, 2003.

[32] M. Meila. Comparing clusterings – an axiomatic view. In *International Conference on Machine Learning*, 2005.

[33] A. Rakhlin and A. Caponnetto. Stability of k-means clustering. In *Neural Information Processing Systems Conference*, 2006.

[34] B. Scholkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*. MIT Press, 2004.

[35] 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.

[36] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[37] N. Srebro. How good is a kernel as a similarity function? In *Proceedings of the 20th The Twentieth Annual Conference on Learning Theory*, 2007.

[38] C. Swamy. Correlation clustering: Maximizing agreements via semidefinite programming. In *Proceedings of the Symposium on Discrete Algorithms*, 2004.

[39] L.G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

[40] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons Inc., 1998.

[41] S. Vempala and G. Wang. A spectral algorithm for learning mixture models. *J. Comp. Sys. Sci.*, 68(2):841–860, 2004.

[42] T. Zhang. Regularized winnow methods. In *NIPS*, 2001.

A Proofs

Theorem 3 *Let $\mathcal{K}$ be a similarity function satisfying the (ν, γ)-average attraction property for the clustering problem (S, ℓ). Using Algorithm 1 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 $(\nu + \epsilon)$-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 $i \leq k'$. Therefore 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

$$\mathcal{K}(x, C(x)) \geq \mathcal{K}(x, C_j) + \gamma \text{ for all } j \in Y, j \neq \ell(x).$$

By assumption and from the previous observations, $\Pr_{x\sim S}[x \in \mathsf{Good}] \geq 1 - \nu - \epsilon/2$. Now, fix $x \in \mathsf{Good}$. Since $\mathcal{K}(x,x') \in [-1,1]$, by Hoeffding bounds we have that over the random draw of ${R_S}^j$, conditioned on ${R_S}^j \subseteq C_j$,

$$\Pr_{{R_S}^j}\left(\left|\mathbf{E}_{x'\sim {R_S}^j}[\mathcal{K}(x,x')] - \mathcal{K}(x,C_j)\right| \geq \gamma/2\right) \leq 2e^{-2|{R_S}^j|\gamma^2/4},$$

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 $x \in \mathsf{Good}$, 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 $x \in \mathsf{Good}$, it implies that the *expected* error of this procedure, over $x \in \mathsf{Good}$, 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. ■

Theorem 9 *Let $\mathcal{K}$ 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 6 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 $\mathcal{K}(C,C')$ 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, $\mathcal{K}(C', C_{r'} - C') > \mathcal{K}(C', C)$. Furthermore, $\mathcal{K}(C', C_{r'} - C')$ is a weighted average of the $\mathcal{K}(C', C'')$ over the sets $C'' \subseteq C_{r'} - C'$ in our current clustering and so at least one such C'' must satisfy $\mathcal{K}(C', C'') > \mathcal{K}(C', C)$. However, this contradicts the specification of the algorithm, since by definition it merges the pair C, C' such that $\mathcal{K}(C', C)$ 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 $\mathcal{K}(C, C_r - C) > \mathcal{K}(C, C')$ or $\mathcal{K}(C', C_{r'} - C') > \mathcal{K}(C, C')$. 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, $\mathcal{K}(C', C_{r'} - C') > \mathcal{K}(C', C_i)$, for $i = 1, \ldots k'$, and so $\mathcal{K}(C', C_{r'} - C') > \mathcal{K}(C', C)$. This again leads to a contradiction as argued above. ■

Lemma 1 Let $\mathcal{K}$ 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 $\mathcal{K}(I, C \setminus I) > \mathcal{K}(I, C' \setminus I)$.

Proof: Let $I^* = I \cap C^*$. So, $I^* = C \cap C' \cap C^*$. We prove first that

$$\mathcal{K}(I, C \setminus I) > \mathcal{K}(I^*, C^* \setminus I^*) - \gamma/2. \tag{1}$$

Since $\mathcal{K}(x, x') \geq -1$, we have

$$\mathcal{K}(I, C \setminus I) \geq (1 - p_1)\mathcal{K}(I \cap C^*, (C \setminus I) \cap C^*) - p_1,$$

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 $|C \setminus I| \cap \bar{C^*} \leq |C \setminus C^*| \leq gn$. 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:

$$\mathcal{K}(I, C \setminus I) \geq (1 - p)\mathcal{K}(I^*, (C \setminus I) \cap C^*) - p. \tag{2}$$

Let $A = (C^* \setminus I^*) \cap C$ and $B = (C^* \setminus I^*) \cap \bar{C}$. We have

$$\mathcal{K}(I^*, C^* \setminus I^*) = (1 - \alpha)\mathcal{K}(I^*, A) - \alpha\mathcal{K}(I^*, B), \tag{3}$$

where $1 - \alpha = \frac{|A|}{|C^* \setminus I^*|}$. Notice that

$$|(C \setminus I) \cap C^*| = |(C \setminus C') \setminus (C \setminus (C' \cap C^*))| \geq |C \setminus C'| - |C \setminus (C' \cap C^*)| \geq |C \setminus C'| - |C \setminus C^*| \geq gn - fn.$$

We also have $|B| = |(C^* \setminus I^*) \cap \bar{C}| \geq |(C^* \setminus C|$. These imply that $1 - \alpha = \frac{|A|}{|A| + |B|} = \frac{1}{1 + \frac{|B|}{|A|}} \geq \frac{g-f}{g}$, and furthermore $\frac{\alpha}{1-\alpha} = -1 + \frac{1}{1-\alpha} \leq \frac{f}{g-f}$.

Equation (3) implies $\mathcal{K}(I^*, A) = \frac{1}{1-\alpha}\mathcal{K}(I^*, C^* \setminus I^*) - \frac{\alpha_1}{1-\alpha_1}\alpha_1\mathcal{K}(I^*, B)$ and since $\mathcal{K}(x, x') \leq 1$, we obtain:

$$\mathcal{K}(I^*, A) \geq \mathcal{K}(I^*, C^* \setminus I^*) - \frac{f}{g - f}. \tag{4}$$

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 $A = (C \setminus I) \cap C^*$. Overall, combining (2) and (4) we obtain: $\mathcal{K}(I, C \setminus I) \geq (1 - p)\left[\mathcal{K}(I^*, C^* \setminus I^*) - \frac{f}{g-f}\right] - p$, so

$$\mathcal{K}(I, C \setminus I) \geq \mathcal{K}(I^*, C^* \setminus I^*) - 2p - (1 - p)\frac{f}{g - f}.$$

We prove now that $2p + (1-p)\frac{f}{g-f} \leq \gamma/2$, which finally implies relation (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 $\mathcal{K}$ 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

$$\mathcal{K}(I^*, C^* \setminus I^*) \geq \mathcal{K}(I^*, C' \setminus (I^* \cup C^*)) + \gamma. \tag{5}$$

We finally prove that

$$\mathcal{K}(I^*, C' \setminus (I^* \cup C^*)) \geq \mathcal{K}(I, C' \setminus I) - \gamma/2. \tag{6}$$

The proof is similar to the proof of statement (1). First note that

$$\mathcal{K}(I, C' \setminus I) \leq (1 - p_2)\mathcal{K}(I^*, (C' \setminus I) \cap \bar{C^*}) + p_2,$$

where $1 - p_2 = \frac{|I^*|}{|I|} \cdot \frac{\left|(C' \setminus I) \cap \bar{C^*}\right|}{|C' \setminus I|}$. We know from above that $\frac{|I^*|}{|I|} \geq \frac{g-f}{g}$, and we can also show $\frac{\left|(C' \setminus I) \cap \bar{C^*}\right|}{|C' \setminus I|} \geq \frac{g-f}{g}$. 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 (1), (5) and (6) together imply the desired result, namely that $\mathcal{K}(I, C \setminus I) > \mathcal{K}(I, C' \setminus I)$. ■

Lemma 2 *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 $s \leq 2\frac{k}{\nu}$. ■

Algorithm 4 Sampling Based Algorithm, List Model

Input: Data set S, similarity function $\mathcal{K}$, parameters $\gamma, \epsilon > 0$, $k \in Z^+$; $d_1(\epsilon, \gamma, k, \delta)$, $d_2(\epsilon, \gamma, k, \delta)$.

- 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 $\rho_U : X \to R^{d_1}$, $\rho_U(x) = (\mathcal{K}(x, x_1), \mathcal{K}(x, x_2), \ldots, \mathcal{K}(x, x_{d_1}))$.
- Pick a set $\tilde{U}$ of d_2 random examples from S where $d_2 = d(\epsilon, \gamma, k, \delta)$ 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 [28, 42] to learn a multiclass linear separator h and add the clustering induced by h to $\mathcal{L}$.
- Output the list $\mathcal{L}$.

Theorem 5 Let $\mathcal{K}$ be a similarity function satisfying the (ν, γ)-average weighted attraction property for the clustering problem (S, ℓ). Using Algorithm 4 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 Sketch: For simplicity we describe the case $k = 2$. The generalization to larger k follows the standard multi-class to binary reduction [40].

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

$$\rho_U(x) = (\mathcal{K}(x, x_1), \mathcal{K}(x, x_2), \ldots, \mathcal{K}(x, x_{d_1}))$$

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 $\tilde{w}$ in the ρ_U space defined as $\tilde{w}_i = \ell(x_i)w(x_i)$, for $x_i \in U^- \cup U^+$ and $\tilde{w}_i = 0$ otherwise. We show that, with probability at least $(1-\delta)$, $\tilde{w}$ 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,

$$\Pr_U \left(\ell(x)\tilde{w} \cdot \rho_U(x) \geq d\frac{\gamma}{4} \right) \geq 1 - \delta^2.$$

To do so, first notice that d is large enough so that with high probability, at least $1-\delta^2$, we have both:

$$|\mathbf{E}_{x' \in U^+}[w(x')\mathcal{K}(x,x')] - \mathbf{E}_{x' \sim S}[w(x')\mathcal{K}(x,x')|\ell(x') = 1]| \leq \frac{\gamma}{4}$$

and

$$|\mathbf{E}_{x' \in U^-}[w(x')\mathcal{K}(x,x')] - \mathbf{E}_{x' \sim S}[w(x')\mathcal{K}(x,x')|\ell(x') = -1]| \leq \frac{\gamma}{4}.$$

Let's consider now the case when $\ell(x) = 1$. In this case we have $\ell(x)\tilde{w} \cdot \rho_U(x) = d(\frac{1}{d}\sum_{x_i \in U_+} w(x_i)\mathcal{K}(x,x_i) - \frac{1}{d}\sum_{x_i \in U_-} w(x_i)\mathcal{K}(x,x_i))$, and so combining these facts we have that with probability at least $(1-\delta^2)$ the following holds:

$$\ell(x)\tilde{w} \cdot \rho_U(x) \geq d(\mathbf{E}_{x' \sim S}[w(x')\mathcal{K}(x,x')|\ell(x') = 1] - \gamma/4 - \mathbf{E}_{x' \sim S}[w(x')\mathcal{K}(x,x')|\ell(x') = -1] - \gamma/4).$$

This then implies that $\ell(x)\tilde{w} \cdot \rho_U(x) \geq d\gamma/2$. Finally, since $w(x') \in [-1,1]$ for all x', and since $\mathcal{K}(x,x') \in [-1,1]$ for all pairs x, x', we have that $||\tilde{w}||_1 \leq d$ and $||\rho_U(x)||_\infty \leq 1$, which implies

$$\Pr_U \left(\ell(x)\frac{\tilde{w} \cdot \rho_U(x)}{||\tilde{w}||_1 ||\rho_U(x)||_\infty} \geq \frac{\gamma}{4} \right) \geq 1 - \delta^2.$$

The same analysis applies for the case that $\ell(x) = -1$.

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 $\tilde{w}$ 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 $\mathcal{K}$ 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 $\tilde{w}$ 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 [29], 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. ■

B 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 part (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. The algorithms we have presented for our most general properties, the average attraction property (Property 2) and the average weighted attraction property (Property 3) are inherently transductive. The number of unlabeled examples needed are as specified by Theorems 3 and 5.

The strict ordering property. We can adapt the algorithm in Theorem 2 to the inductive setting as follows. We first draw a set S of $n = O\big(\frac{k}{\epsilon}\ln\big(\frac{k}{\delta}\big)\big)$ unlabeled examples. We run the algorithm described in Theorem 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 $\mathrm{argmax}_{q \in Q} \mathcal{K}(x, q)$; 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 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 our strict ordering 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 to the inductive setting Algorithm 3 we have presented for the Property 6. The main difference in the inductive setting is that we have to *estimate* (rather than *compute*) the $|C_r \setminus C_{r'}|$, $|C_{r'} \setminus C_r|$, $|C_r \cap C_{r'}|$, $\mathcal{K}(C_r \cap C_{r'}, C_r \setminus C_{r'})$ and $\mathcal{K}(C_r \cap C_{r'}, C_{r'} \setminus C_r)$ for any two clusters C_r, $C_{r'}$ in the list $\mathcal{L}$. We can easily do that with only $\mathrm{poly}(k, 1/\epsilon, 1/\gamma, 1/\delta)\log(|\mathcal{L}|))$ unlabeled points, where $\mathcal{L}$ is the input list in Algorithm 3 (whose size depends on $1/\epsilon$, $1/\gamma$ and k only). Specifically, using a modification of the proof in Theorem 8 and standard concentration inequalities (e.g. the McDiarmid inequality [17]) we can show that:

Theorem 10 *Let $\mathcal{K}$ 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 3 with parameters $\alpha = O(\epsilon/k)$, $g = O(\epsilon^2/k^2)$, $f = O(\epsilon^2\gamma/k^2)$, together with Algorithm 1 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 $O\big(\frac{k}{\gamma^2}\ln\big(\frac{k}{\epsilon\delta}\big)\cdot\big(\frac{k}{\epsilon}\big)^{\frac{4k}{\gamma^2}\ln\left(\frac{k}{\epsilon\delta}\right)}\ln\big(\frac{1}{\delta}\big)\big)$ unlabeled points in the first phase and $O\big(\frac{1}{\gamma^2}\frac{1}{g^2}\frac{k}{\gamma^2}\log\frac{1}{\epsilon}\log\frac{k}{\delta f}\log k\big)$ unlabeled 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.[6] Specifically:

Property 7 *The similarity function $\mathcal{K}$ satisfies the γ-**strong stability** property for the clustering problem (D,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*

$$\mathcal{K}(A, C_r \setminus A) > \mathcal{K}(A, A') + \gamma.$$

[6] To see why this is necessary consider the following example. Suppose all $\mathcal{K}(x,x')$ values are equal to $1/2$, except for a special single center point x_i in each cluster C_i with $\mathcal{K}(x_i, x) = 1$ for all x in C_i. This satisfies strong-stability since for every $A \subset C_i$ we have $\mathcal{K}(A, C_i \setminus A)$ is strictly larger than $1/2$. Yet it is impossible to cluster in the inductive model.

For this property, we could always run the algorithm for Theorem 10, 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 2 to the inductive setting as in the case of the strict order property. However, the proof here is substantially more involved.

Algorithmically, 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. Let Q be the set of leaves of this tree. We 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 $\mathrm{argmax}_{q \in Q} \mathcal{K}(x, q)$; if u appears on the path from $q(x)$ to the root, then set $f_u(x) = 1$, otherwise set $f_u(x) = 0$.

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, . Remember that $\mathcal{C} = \{C_1, \ldots, C_k\}$ is the correct 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 11 below implies that with high probability the clusters C'_i corresponding to the large ground-truth clusters satisfy our property with a gap $\gamma/2$. 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 6 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 $\mathcal{K}(x, C(x))$ is within $\gamma/2$ of $\mathcal{K}(x, C'(x))$ 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 $\mathcal{C}$.

We prove in the following that for a sufficiently large value of n sampling preserves stability.

Theorem 11 *Let $C_1, C_2, \ldots, C_k$ be a partition of a set X with N elements such that for any $S \subseteq C_i$ and any $x \notin C_i$,*

$$K(S, C_i \setminus S) \geq K(S, x) + \gamma.$$

Let C'_i be a random subset of n elements of C_i. Then, $n = poly(1/\gamma, 1/\delta)$ is sufficient so that with probability $1 - \delta$, for any $S \subset C'_i$ and any $x \in C' \setminus C'_i$,

$$K(S, C'_i \setminus S) \geq K(S, x) + \frac{\gamma}{2}.$$

In the rest of this section we sketch a proof which follows closely ideas from [18] and [4].

For a real matrix A, a subset of rows S and subset of columns T, let $A(S, T)$ denote the sum of all entries of A in the submatrix induced by S and T. The cut norm $||A||_C$ is the maximum of $|A(S, T)|$ over all choices of S and T.

We use two lemmas that are closely related to the regularity lemma but more convenient for our purpose.

Theorem 12 *[18][Cut decomposition] For any $m \times n$ real matrix A and any $\varepsilon > 0$, there exist matrices $B_1, \ldots B_s$ with $s \leq 1/\varepsilon^2$ such that each B_l is defined by a subset R_l of rows of A and a subset C_l of columns of A as $B^l_{ij} = d_l$ if $i \in R_l$ and $j \in C_l$ and $B^l_{ij} = 0$ otherwise, and $W = A - (B^1 + \ldots + B^s)$ satisfies: for any subset S of rows of A and subset T of columns of A,*

$$|W(S, T)| \leq \varepsilon \sqrt{|S||T|} ||A||_F \leq \varepsilon \sqrt{|S||T|mn} ||A||_\infty.$$

The proof is straightforward: we build the decomposition iteratively, if the current W violates the required condition, we define the next cut matrix using the violating pair S, T and set entries in the induced submatrix to be $W(S, T)/|S||T|$.

When A is the adjacency matrix of an n-vertex graph, we get $|W(S, T)| \leq \varepsilon n \sqrt{|S||T|} \leq \varepsilon n^2$.

Theorem 13 *[4][Random submatrix] For $\varepsilon, \delta > 0$, and any B be an $N \times N$ real matrix with $||B||_C \leq \varepsilon n^2$, $||B||_\infty \leq 1/\varepsilon$ and $||B||_F \leq n$, let S be a random subset of the rows of B with $q = |S|$ and H be the $q \times q$ submatrix of B corresponding to S. For $q > (c_1/\varepsilon^4\delta^5)\log(2/\varepsilon)$, with probability at least $1-\delta$,*

$$||H||_C \leq c_2 \frac{\varepsilon}{\sqrt{\delta}} q^2$$

where c_1, c_2 are absolute constants.

Proof: The proof follows essentially from Theorem 1 of [4]. We sketch it here. Fix C_1, we'll apply the argument to each C_i separately. Fix also an integer $1 \leq t \leq |C_1|$. For a set $S \subseteq C_1$ with $|S| = t$ and a point $x \in C \setminus C_1$, we consider the function

$$f(S) = \frac{1}{t(|C_1|-t)} \sum_{i \in S, j \in C_1 \setminus S} K(i,j) - \frac{1}{t}\sum_{i \in S} K(i,x).$$

We can rewrite this as follows. Let $A(i,j) = K(i,j)/t(|C_1|-t)$ for $i,j \in C_1$ and $A(i,j) = -K(i,j)/t$ for $i \in C_1, j \in C \setminus C_1$ and $A(i,j) = 0$ otherwise. We see that,

$$\min f(S) = \min\{\sum_{i,j \in C_1} A(i,j)y_i(1-y_j) + \sum_{i \in C_1, j \in C \setminus C_1} A(i,j)y_iy_j \mid y \in \{0,1\}^{|C|}, \sum_{i \in C_1} y_i = t, \sum_{i \in C \setminus C_1} y_i = 1\}$$

The techniques of [4] show that this minimum is approximated by the minimum over a random subset of C. In what follows, we sketch the main ideas.

Let A be the similarity matrix for C. Fix a cut decomposition $B^1, \ldots B^s$ of A. Let $W = A - (B^1 + \ldots + B^s)$ and we have $|W(S,T)| \leq \varepsilon N^2$ (Theorem 12).

For convenience, assume A is symmetric with $0-1$ entries. Then each B^l is also symmetric and induced by some subset R_l of C with $|R_l| \geq \varepsilon_1 N$. Let the induced cut decomposition for the sample A' corresponding to the sample C' be $B' = B'^1 + B'^2 + \ldots + B'^s$ and each B'^i is induced by a set $R_i' = R_i \cap C'$. By Theorem 13, we know that this induced cut decomposition gives a good approximation to $A'(S,T)$, i.e., if we set $W' = A' - B'$, then

$$|W'(S,T)| \leq 2c_2\varepsilon n^2.$$

We now briefly sketch the proof of Theorem 11. First, it holds for singletons subsets S with high probability using a Chernoff bound. In fact, we get good estimates of $K(x, C_i)$ for every x and i. This implies that the condition is also satisfied for every subset of size at least $\gamma n/2$. It remains to prove this for large subsets. To do this, observe that it suffices to prove it using B' as the similarity matrix rather than A' (for a slightly larger threshold).

The last step is to show that the condition is satisfied by B' which is a sum of s cut matrices. There are two ideas here: first, the weight in B of the edges of any cut of C is given by knowing only the *sizes* of intersections of the shores of the cut with each of the subsets inducing the cut matrices. Next, the minimum value attained by any set is approximated by a linear program and the sub-LP induced by a random subset of variables has its optimum close to that of the full LP. Thus, the objective value over the sample is also large. ■

C Examples

Strict ordering and Spectral partitioning Figure 2 shows that it is possible for a similarity function to satisfy the strict ordering property for a given clustering problem for which Theorem 2 gives a good algorithm, but nonetheless to fool a straightforward spectral clustering approach.

Linkage-based algorithms and strong stability Figure 3 (a) gives an example of a similarity function that does not satisfy the strict ordering property, but for large enough m, w.h.p. will satisfy the strong stability property.[7] However, single-linkage using $\mathcal{K}_{max}(C, C')$ would still work well here. Figure 3 (b) extends

[7] This is because there are at most m^k subsets A of size k, and each one has failure probability only $e^{-O(mk)}$.

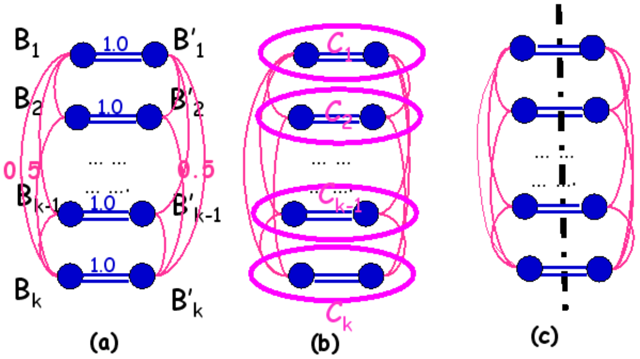

Figure 2: Consider $2k$ blobs $B_1, B_2, \ldots, B_k$, $B'_1, B'_2, \ldots, B'_k$ of equal probability mass. Points inside the same blob have similarity 1. Assume that $\mathcal{K}(x, x') = 1$ if $x \in B_i$ and $x' \in B'_i$. Assume also $\mathcal{K}(x, x') = 0.5$ if $x \in B_i$ and $x' \in B_j$ or $x \in B'_i$ and $x' \in B'_j$, for $i \neq j$; let $\mathcal{K}(x, x') = 0$ 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)).

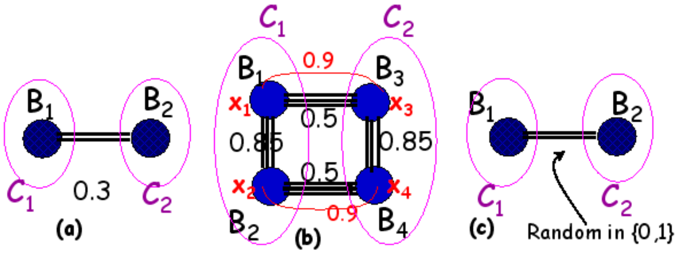

Figure 3: Part (a): Consider two blobs B_1, B_2 with m points each. Assume that $\mathcal{K}(x, x') = 0.3$ if $x \in B_1$ and $x' \in B_2$, $\mathcal{K}(x, x')$ is random in $\{0, 1\}$ if $x, x' \in B_i$ for all i. Clustering C_1, C_2 does not satisfy Property 1, but for large enough m, w.h.p. will satisfy Property 4. Part (b): Consider four blobs B_1, B_2, B_3, B_4 of m points each. Assume $\mathcal{K}(x, x') = 1$ if $x, x' \in B_i$, for all i, $\mathcal{K}(x, x') = 0.85$ if $x \in B_1$ and $x' \in B_2$, $\mathcal{K}(x, x') = 0.85$ if $x \in B_3$ and $x' \in B_4$, $\mathcal{K}(x, x') = 0$ if $x \in B_1$ and $x' \in B_4$, $\mathcal{K}(x, x') = 0$ if $x \in B_2$ and $x' \in B_3$. Now $\mathcal{K}(x, x') = 0.5$ 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 $\mathcal{K}(x_1, x_3) = 0.9$. Similarly $\mathcal{K}(x, x') = 0.5$ 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 $\mathcal{K}(x_2, x_4) = 0.9$. For large enough m, clustering C_1, C_2 satisfies Property 4. 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 $\{0, 1\}$.

this to an example where single-linkage using $\mathcal{K}_{max}(C, C')$ fails. Figure 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 1 will succeed.