Active Search with Complex Actions and Rewards

Yifei Ma

MAY 2017
CMU-ML-17-101

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Jeff Schneider, **Chair**
Roman Garnett
Aarti Singh
Alexander J. Smola
Ryan P. Adams

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

This research was sponsored by: Air Force Research Laboratory awards FA865010C7059 and FA87501420244;
National Science Foundation award IIS0911032; and Highmark DHTI award 1020167.

To my parents, Lijun Zhang and Hailin Ma.

Abstract

Active search studies algorithms that can find all positive examples in an unknown environment by collecting and learning from labels that are costly to obtain. They start with a pool of unlabeled data, act to design queries, and get rewarded by the number of positive examples found in a long-term horizon. Active search is connected to active learning, multi-armed bandits, and Bayesian optimization.

To date, most active search methods are limited by assuming that the query actions and rewards are based on single data points in a low-dimensional Euclidean space. Many applications, however, define actions and rewards in a more complex way. For example, active search may be used to recommend items that are connected by a network graph, where the edges indicate item (node) similarity. The active search reward in environmental monitoring is defined by regions because pollution is only identified by finding an entire region with consistently large measurement outcomes. On the other hand, to efficiently search for sparse signal hotspots in a large area, aerial robots may act to query at high altitudes, taking the average value in an entire region. Finally, active search usually ignores the computational complexity in the design of actions, which is infeasible in large problems.

We develop methods to address the disparate issues in the new problems. In a graph environment, the exploratory queries that reveal the most information about the user models are different than the Euclidean space. We used a new exploration criterion called Σ-optimality, which is motivated by a different objective, active surveying, yet empirically performed better due to a tendency to query cluster centers. We also showed submodularity-based guarantees that justify for greedy application of various heuristics including Σ-optimality and we performed regret analysis for active search with results comparable to existing literature. For active area search for region rewards, we designed an algorithm called APPS, which optimizes for one-step look-ahead expected rewards for finding positive regions with high probability. APPS was initially solved by Monte-Carlo estimates; but for simple objectives, e.g. to find region with large average pollution concentrations, APPS has a closed-form solution called AAS that connects to Bayesian quadrature. For active needle search with region queries using aerial robots, we pick queries to maximize the information gain about possible signal hotspot locations. Our method is called RSI and it reduces to bisection search if the measurements are noiseless and the signal hotspot is unique. Turning to noisy measurements, we showed that RSI has near-optimal expected number of measurements, which is comparable to results from compressive sensing (CS). On the other hand, CS relies on weighted averages, which are harder to realize than our use of plain averages. Finally, to address the scalability challenge, we borrow ideas from Thompson sampling, which approximates near-optimal decisions by drawing from the model uncertainty and using greedy decisions accordingly. Our method is conjugate sampling, which allows true computational benefits when the uncertainty is modeled with sparse or circulant matrices.

Acknowledgments

I would like to thank my advisor Jeff Schneider and my effective co-advisor Roman Garnett for offering me long-term apprenticeship and friendship. They invited me to the topic of Bayesian active search and inspired me with its huge potentials in applications. I could not have proceeded and discovered the true excitements about research without their help.

Next, I would like to thank my coauthors, Dougal Sutherland, Tzu-Kuo Huang, and Andrew Gordon Wilson for their significant contributions in the papers that made up the technical body of this thesis. Appreciations also go to an extended list of collaborators in various projects during my PhD study, including Gus (Guangyu) Xia, Tongbo Huang, Junior Oliva, Ying Yang, and Xiaoqi Yin. Collaborations with peer students bring back some of my best memories.

I am fortunate to have an awesome list of committee members: Aarti Singh, Alex Smola, and Ryan Prescott Adams. They offered me insightful comments, deep understanding of related work, and ideas to make bigger impacts. Thank you.

PhD is a long learning process, where I benefited a lot from discussions with Artur W. Dubrawski, Barnabas Poczos, Eric P. Xing, Geoff Gordon, and Roy Maxion. I was also inspired by a long list of peer students, including Akshay Krishnamurthy, Madalina (Ina) Fiterau, Min Xu, Liang Xiong, Yi Zhang, Siamak Ravanbakhsh, Aaditya Ramdas, Xi Chen, Yisong Yue, and James Sharpnack. CMU is a large family, where I also have many interesting discussions with my friends and labmates, Kumar Avinava Dubey, Sashank J. Reddi, Ahmed Hefny, Xuezhi Wang, Yining Wang, Yuxiang Wang, Kirthevasan Kandasamy, and Willie Neiswanger.

Finally, I would love to appreciate my family for their support. This list of names is mostly from a work-related perspective and not even close to the full set of people whom I must acknowledge for their help both in research and in life.

Contents

List of Figures

List of Tables

1

Introduction

We study the problem of active search for positive instances with desired properties [Garnett et al., 2012, Wang et al., 2013]. Active search is like active learning in binary settings [Settles, 2010], but the objective is to recall all positive instances. It assumes a similar paradigm: First, details about the search domain and the desired properties are provided. Then, an algorithm or autonomous machine will conduct the search iteratively, where for each step, the algorithm or machine will select an instance, obtain feedback by querying human or interacting with the environment at the selected point, and update its internal parameters to improve the next selections. The iterative process continues until the user quits and, while trials and errors are bound to happen, the ultimate goal is to maximize the total number of positive instances found in the end.

Autonomous systems operating under this paradigm may be valuable in many applications. For example, in environmental monitoring, we take samples at various locations to find all polluted areas and identify their the sources. In an email investigation, we want to retain all emails with questionable content in order to provide evidence. In social science, we want to find people who have unique opinions in order to understand them. In search and rescue operations, we want to locate all human survivors of a disaster in a large area. Active search can help by making decisions about where to inspect in order to find all relevant information, in a similar manner to human expert investigations.

Active search focuses on collecting and learning from feedback in a sequential application of open-loop search. For example, in environmental monitoring by fan-boat, the information from each search query (i.e., taking measurement at any location) is limited at the chosen location and maybe its adjacent locations. Therefore, to find all positive readings that indicate pollutions, we need to actively plan for the next locations to take measurements after obtaining results at the previous locations. This is different from the *passive* search in information retrieval context

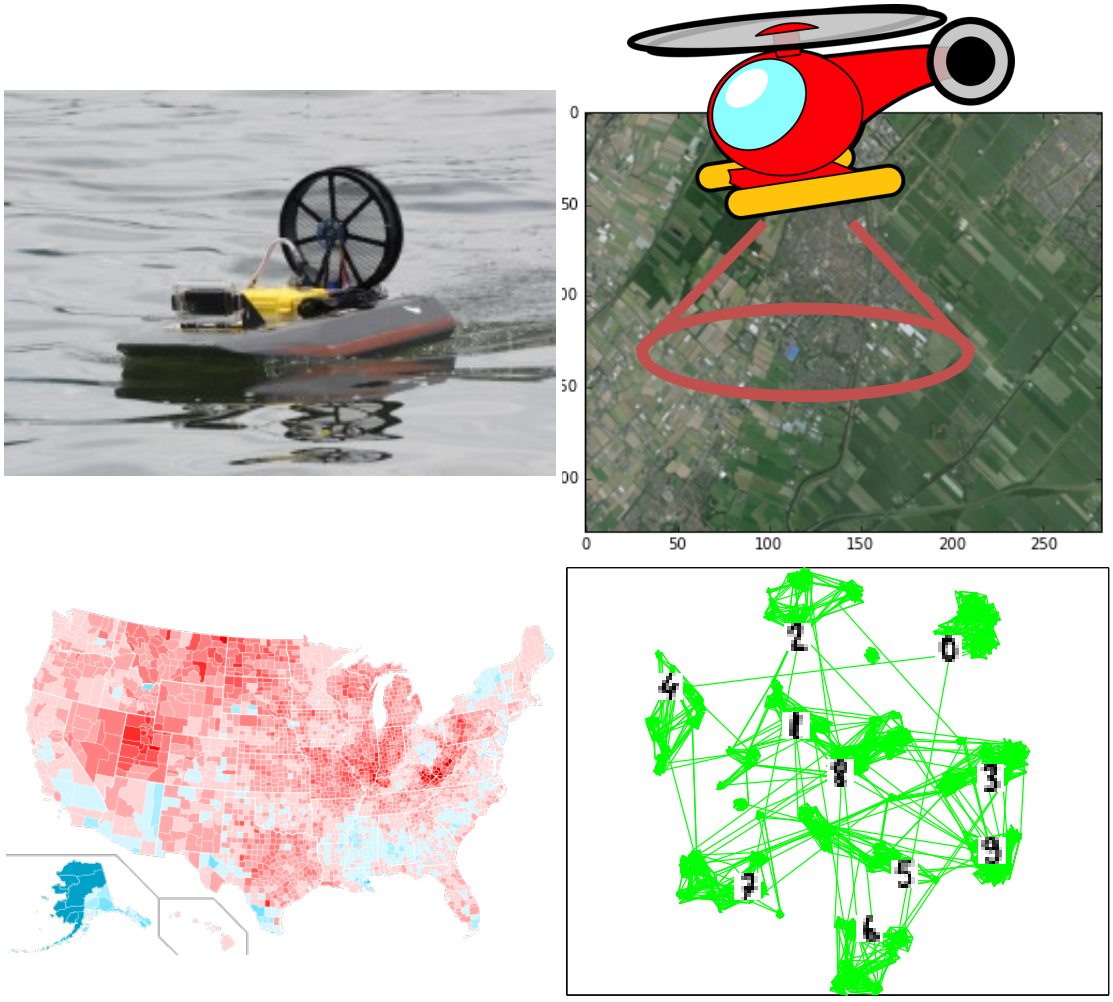


Figure 1.1: Active search applications: environmental monitoring, aerial search, public opinion search, and finding all relevant information.

where the ultimate goal is to retrieve all values whose keywords match the search word [Croft et al., 2010, Manning et al., 2008]. For our *active* search in this context, a more relevant task would be to interactively refine search results in cases where the initial keyword is ambiguous. We will visit a similar problem in details in Chapter 2.

Another related but different interpretation of active search is recommender systems [Adomavicius and Tuzhilin, 2005, McMahan et al., 2013]. These systems are widely used in online interactive marketplaces, where the goal is to provide online customers items that they will likely to click, i.e. positive items in the prediction of clicks based on the customers' previous browsing history. The idea is to model every customer's preference based on all other customers who have exhibited similar preferences in their previous browsing history. Even though recommender systems are built for customer interactions, the algorithms themselves do not usually use interactive learning or active explorations. As a result, recommender systems are not suitable for use in active search applications in unknown environments.

A slightly more complicated approach is reinforcement learning [Sutton and Barto, 1998]. However, reinforcement learning focuses on finding an optimal strategy after solving many active search problems in controlled environments, whereas we focus on finding good strategies to solve new active search problems under lenient assumptions.

1.1 Common Solutions to Active Search Problems

So, how do we solve active search problems? Although active search is a newer concept, there are many algorithms in related fields that can serve as a good starting point.

Designs of experiments (DOE) [Krause et al., 2008, Montgomery, 2012] are based on the idea that collecting best quality data is often more useful than collecting more data, especially when data collection is costly. In our terminology, experiments mean human/environmental interactions. The goal in experimental designs is usually to reduce the uncertainty in the parameter space of the model that predicts interaction outcomes (i.e., instance labels in our case). When applied to active search, once the underlying model is obtained, the positive instances may be directly observed. In fact, many existing systems are built on the explore-then-commit idea, including robotic search for radiation sources, a/b testing and adoption of the optimal policy, etc. While being the most reliable baseline, the idea of explore-then-commit is usually not the most efficient for finding all positives using as few query interactions as possible.

At the other extreme, Bayesian optimization (BO) [Brochu et al., 2010, Jones et al., 1998, Mockus, 1974] aims to directly find the global optimum of a black-box function. BO relies on Bayes priors, which define the scope of the black-box optimization problem (or the active search problem) via probability distributions that jointly model all possible interaction outcomes at all queryable instances. The Bayesian view also allows for simulation on the evolution of the interaction outcomes without interactions actually taking place. This thought process is called look-ahead modeling. Upon revealing of true interaction outcomes, a posterior model is formed by reasoning with both empirical evidence and the prior model. Then, new data collection decisions are made based on the current posterior model.

A naive solution to BO chooses queries in order to greedily maximizes the expected improvement on the maximum value at the query point, in terms of its one-step look-ahead model [Jones et al., 1998]. Hennig and Schuler [2012], Hernández-Lobato et al. [2016] considered a global measure of utility also in one-step look-ahead modeling. BO may be used for active search to find singular positive instances; to further find all positive instances, one must modify the objective to simultaneously find global optima and stay away from the previously found positive instances [Vanchinathan et al., 2013].

Further, based on the same Bayesian modeling, active search may be directly approached. Garnett et al. [2012], Wang et al. [2013] use an objective that counts the expected number of positives in a multi-step look-ahead model, where at every step the algorithm chooses the Bayes-optimal query according to the look-ahead simulations. A true Bayes-optimal decision is arguably the best decision, but their computation is often prohibitively slow because they involve infinite-step

look-ahead modeling. Garnett et al. [2012], Wang et al. [2013] used two-step approximations and showed good empirical results. Branch-and-bound pruning was used to further increase the decision speed.

Finally, to combine exploratory DOE and greedy BO for long-term rewards without the complexity of multiple-step look-ahead, recent research focuses on a set of statistical models called multi-armed bandits (MAB) [Auer, 2003, Bubeck, 2012, Gittins, 1979]. MAB studies the problem where there is a pool of bandit arms, each of which holds a hidden distribution and can output a random reward value accordingly if it is chosen to be played. The objective is to accumulate maximum sums of rewards after finite rounds, assuming each round costs a unit token for any choice of arm. MAB focuses on guarantees on *cumulative regret*, which is the gap in expected cumulative rewards between the optimal choices of arms and the choices from the algorithm. A meaningful guarantee on cumulative regret should be sub-linear in terms of the number of play rounds. To obtain guarantees on cumulative regrets, a common solution to MAB problems usually involves two considerations: exploitation and exploration. Exploitation prefers to greedily choose the best options based on empirical results, similar to the principles in BO, whereas exploration considers choosing new or under-explored options to reduce model uncertainty like DOE.

We can adapt MAB strategies for use in active search if we treat each arm as a searchable instance and disallow repeated play of the same arms. We show in [Ma et al., 2015a] that similar guarantees are obtainable in our choice of model.

1.2 Limitations on Existing Active Search Solutions

However, current research on active search fails to realize the complexity in real applications. They typically assume that a search action can only apply to an individual arm or a single point, the following observation will only cover that single point, and a search reward will be assigned to the same point. In practice, actions may be allowed on a group of points and the search objective or reward may also be a global pattern defined on a region. Another real-world complexity is the search domain. Instead of a Euclidean space, instances can be embedded as nodes in a graph structure. My research is on intuitive algorithms under these circumstances.

To begin with, we study active search on graphs, where the instances are represented as graph nodes and the pairwise similarity between instances are recorded as graph edges. The edges are observed a-priori, but the node labels are hidden and only revealed upon queries. For example, in an email investigation, the links play an important role for the distribution of questionable content. Active search aims to find all emails that may be positive evidence for a misconduct, decided by human investigators. Simple application of linear-bandits [Dani et al., 2008] and Gaussian process-bandits [Srinivas et al., 2010b] will cause undesirable focus on the graph periphery (i.e., leaf nodes that have long graph traversal distances to most other nodes), where the uncertainty is the largest according to linearization of the graphical models. However, querying on the periphery intuitively fails on the promise of model uncertainty reduction.

Our next application is on active search for patterns defined at a region level. An example is environmental monitoring by an autonomous fan-boat. While the boat travels and takes point measurements with its on-board sensors at locations of its choice, identification of real pollution problems requires consistent measurements in a large region. We label a region as positive if the mean value in the region exceeds a given threshold with high probability. Another example is electoral polling where the objective is to find winning states that include a lot of sample points. We even want to find more complex patterns defined by functionals on regions. However, using point-based active search may not be the most efficient solution.

We also consider searching for signal hotspots in a large area using aerial robots that take aggregate measurements at high altitudes. Examples of the signal hotspots include radiation sources, gas leaks, and human survivors of disasters. The measurements are aggregate, taken at high altitudes with limited spatial resolutions. For simplicity, we consider single-pixel cameras that record the average values in rectangular regions. Intuitively, a good search algorithm should take advantage of the increased coverage of measurements at high altitudes, while also pay attention to the increased noise as the coverage increases. However, the problem of aerial search using aggregate measurements under rectangular constraints has rarely been discussed before.

Finally, Bayesian approaches for active search and optimization traditionally ignores the complexity of the decision process, assuming that the actual experiments cost much more time and resources. However, such assumptions may hinder their wider applications in less-expensive experiments. Recent discussions on Thompson sampling suggest that inaccurate, noisy decisions can also yield reasonable convergence. Despite conceptual simplicity, little real advantage has been shown for Thompson sampling in either computational or statistical complexity. For example, Thompson sampling requires an exact draw from the Bayes posterior distribution, which is often hard for complex distributions. Can we use Thompson sampling to make fast, inaccurate draws from approximate posterior sampling, in order to truly speed up the decision process to choose queries, especially in the above applications?

Table 1.1 summarizes the three components for my PhD thesis.

Table 1.1: Thesis Components

Active search	Point rewards	Region rewards
Point actions	• Active search on graphs [Ma et al., 2013, 2015a] • Conjugate sampling [Ma et al., 2017b]	• Active area search and Pointillism [Ma et al., 2014, 2015b]
Group actions	• Active needle search with region sensing [Ma et al., 2017a]	• A unified model (future work)

1.3 Thesis Contributions

1.3.1 Active Search on Graphs

Assume we are given a graph with known edges but unknown node labels; we study the sequential design of queries on the node labels for several interconnected problems: to *survey* the percentage of positive nodes, to *learn* (i.e., predict) all the unqueried nodes, and eventually to *search for* (i.e., collect) all positive nodes. The objective is to achieve the best task performance with any query budget, starting with no initial labels and only using information given by the graph connectivity.

There are many ways to use the information embedded in the graph structure; we assume a prior distribution on the node values in the family of Gaussian Random Fields (GRFs) [Zhu et al., 2003a]. For active learning and surveying, we aim to minimize the uncertainty of the posterior model, using a novel Σ-optimality criterion [Garnett et al., 2012]. For active search of positive nodes, we aim to minimize cumulative regret, which is the cumulative gap in the node values between an optimal sequence of query choices and our query choices, using a method called GP-SOPT that combines GP-UCB [Srinivas et al., 2010b] and Σ-optimality.

On both active learning and surveying, Σ-optimality empirically outperformed a rich set of baselines including uncertainty sampling [Settles, 2010], expected error reduction [Zhu et al., 2003b], D-optimality [Krause et al., 2008], and V-optimality [Ji and Han, 2012]. One explanation we found was that Σ-optimality tends to query cluster centers, whereas the alternatives tend to query on the periphery (e.g., leaf nodes) of a graph. We also showed a near-optimal theoretical guarantee on the sequential application of D, V, and Σ-optimality. On active search, GP-SOPT also outperformed GP-UCB, while having comparable theoretical regret bounds.

1.3.2 Active Area Search and Pointillism

We introduce the problem of active area search, which seek to discover regions of a domain exhibiting desired behavior with limited observations. Unusually, the patterns we consider are defined by large-scale properties of an underlying function that we can only observe at a limited number of points. Given a description of the desired patterns (e.g., the average value in the regions exceeding a given threshold or patterns defined in the form of a classifier taking functional inputs), we sequentially decide where to query function values to identify as many regions matching the pattern as possible, with high confidence. Our naive solution, called Active Pointillist Pattern Search (APPS), uses Monte-carlo estimation of the expected rewards in one-step lookahead. For one broad class of models, including finding regions with high average values, the expected reward of each unobserved point can be computed analytically, yielding an analytical solution we call Active Area Search (AAS). We demonstrate the proposed algorithms on three difficult search problems: locating polluted regions in a lake via mobile sensors, forecasting winning electoral districts with minimal polling, and identifying vortices in a fluid flow simulation.

1.3.3 Active Needle Search with Region Sensing

We consider using aerial robots to search for threats, gas leaks, or human survivors of disasters. Intuitively, search algorithms may increase efficiency by collecting aggregate measurements summarizing large contiguous regions. However, most existing search methods either ignore the possibility of such region observations (e.g., Bayesian optimization and multi-armed bandits) or make strong assumptions about the sensing mechanism that allow each measurement to arbitrarily encode all signals in the entire environment (e.g., compressive sensing), which ignores the physical limitations of aerial robots with on-board sensors. We model the limitation as *region sensing constraint*, which allows only noisy observations of the plain average values in rectangular regions (including single points).

We propose an algorithm that actively collects data to search for sparse signals using region sensing, based on the greedy maximization of information gain. Assuming that the observation noise is a superstition of standard Gaussian noise at every point in a region, we analyze our algorithm in $1d$ and show that it requires $\tilde{O}(n/\mu^2 + k^2)$ measurements to recover all of k signal locations with small Bayes error, where μ and n are the signal strength and the size of the (discretized) search space, respectively. We also show that active designs can be fundamentally more efficient than passive designs with region sensing, contrasting with the results of Arias-Castro et al. [2013]. We demonstrate the empirical performance of our algorithm on a search problem using satellite image data and in high dimensions.

1.3.4 Conjugate Sampling

We study conjugate sampling, which is an alternative to Thompson sampling to further speed up Bayesian decision making by using fast, inaccurate draws from approximate posterior sampling. Conjugate sampling makes Bayesian optimization decisions in $O(\sqrt{\kappa} t_A)$ time and with $O(n)$ excess memory at the same time, where κ is the condition number of the information matrix of the posterior distribution, n is the dimension of the design space, and t_A is the time complexity of matrix-vector multiplications involving the information matrix. While comparable to Thompson sampling in general cases, our method yields additional computational benefits, in terms of both space and time complexity, when we use sparse or circulant information models such as Gaussian random fields [Ma et al., 2015a] or Gaussian processes with Kronecker-decomposable kernels [Flaxman et al., 2015, Ma et al., 2014, Wilson and Nickisch, 2015].

2

Active Search on Graphs

2.1 Introduction

As the world gets increasingly digitized and electronically recorded, how to quickly identify relevant pieces of information becomes a major issue. Internet search engines are an integral part of modern life, serving as a probe into the diverse, complex and expanding space of human digital traces. Despite being successful in many information retrieval tasks, the keyword-based query mechanism in most search engines may fall short when the targets are characterized by complex patterns or signatures beyond keywords. For example, financial transactions associated with illegal activities bear signatures involving multiple factors such as time, location, occupation of the account owner, etc. In the investigation of organizational misconduct, such as the Enron scandal, the important leads or evidence, oftentimes buried in a sea of diverse electronic and paper trails, usually involve information exchange among key individuals and their relationship. In these situations, keyword-based search may serve as a good starting point, but is certainly far from completing the task.

Such needs of more general search paradigms have recently motivated several efforts [Garnett et al., 2012, Vanchinathan et al., 2013, Wang et al., 2013], most of which are related to the active search framework proposed by Garnett et al. [2012]. Active search is an interactive search mechanism that begins with the user providing one or few target examples, referred to as seeds, such as past financial transactions that have been linked to illegal activities. Based on these seeds, an algorithm figures out what instances the user should examine next and presents them to the user, who then decides whether the presented instances are relevant or not. Upon receiving this feedback, the algorithm updates its search strategy accordingly and selects the next instances to present. The loop continues until the user quits, and the goal is to maximize the total number of

relevant instances found.

As one can see, active search has close connections to some well-studied machine learning paradigms. At a first glance, active learning [Settles, 2010] seems the most related because they both ask for user feedback incrementally and adaptively. However, active learning aims at improving generalization performances with as few label queries as possible, while active search is evaluated by how many relevant instances it can find along the way, and therefore must carefully balance Exploitation and Exploration (E&E). In contrast, active learning only considers exploration, which is half of the problem. The E&E trade-off relates active search to stochastic optimization in the multi-armed bandit setting [Bubeck et al., 2009, Dani et al., 2008, Kleinberg et al., 2008, Robbins, 1985], where the goal is to find the maximum of an unknown function using as few function evaluations as possible. However, active search deviates from this setting in that it selects instances *without replacement* and is competing with the best *subset* of instances rather than the single best.

We investigate active search when the instances are represented by a graph whose edges encode pairwise similarity among the instances, represented as nodes. Many real-world data are of this type, such as web pages, citation networks, and e-mail correspondences. For data that are not naturally represented as graphs, a graph that connects the nearest neighbors of each data point can still be beneficial because it may reveal useful manifold structures [Belkin and Niyogi, 2001, Tenenbaum et al., 2000]. We use active search to find positive nodes on the given graph, using the information that connected nodes tend to have similar labels to improve its efficiency.

2.1.1 Graphs

The main character of graph-based representation of data is that, before collecting actual labels, all prior features of a data point, represented as a node in the graph, are implicitly characterized by the connections it has with all other data points, i.e. the graph edges. The graph representations are, in principle, tangential to the usual tabular representation of data where instances are separated by rows and features are separated by columns. For simplicity, we only discuss the graphical properties of data. For example, when making document recommendations, we will mostly only consider the citation patterns, while ignoring any information on the text of the documents, such as their topic features. To make the distinctions clear, it is possible to include the tabular information when building the graphs, i.e., the edges may include the similarity in topics between two documents, besides their citation links. The difference is that such edge engineering is done as a preprocessing step, out of the scope of this thesis. A formal treatment may use Conditional Random Fields (CRFs) [Lafferty et al., 2001], which transforms all types of features into a graphical model.

Example 2.1 (Graphs)**.** *Some datasets are naturally represented by graphs. Ji and Jin [2017] used coauthorship and citation information to infer communities among statisticians. Figure 2.1 shows a large component in a coauthorship network for statisticians, where an edge is formed if two authors have coauthored two or more papers in high-profile venues. Names are shown for nodes with the highest degrees. Nodes are also colored according to a result from community*

detection using Newman's Spectral Clustering method (NSC) [Newman, 2006]. The communities can be explained by the researchers' academic ties and interests. The example shows that using the connectivity information alone, one may be able to infer useful properties of the nodes.

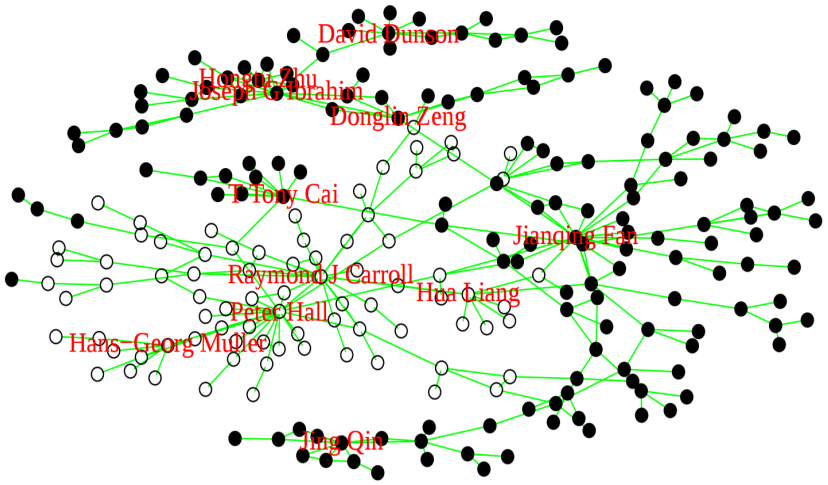


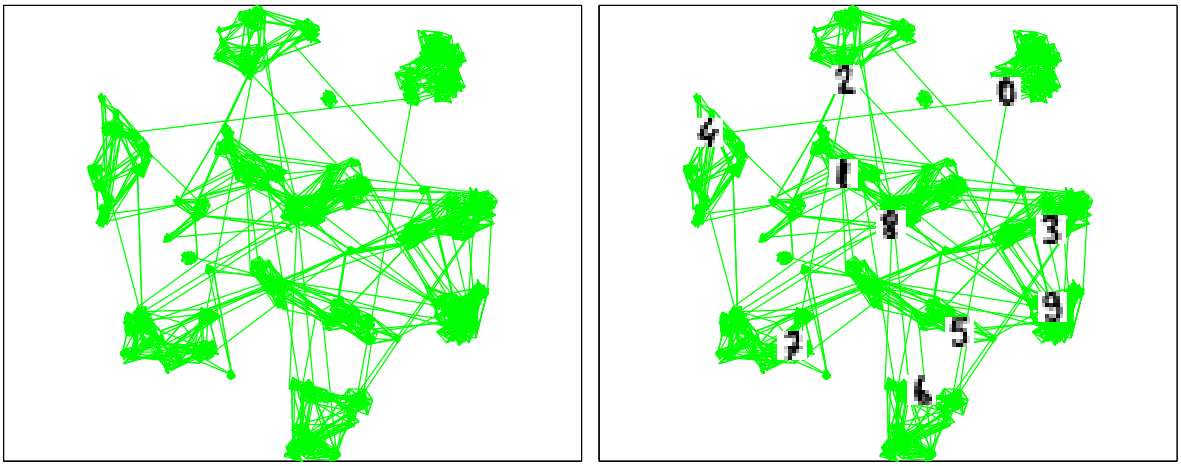
Figure 2.1: (Partial) coauthorship network for statisticians by Ji and Jin [2017].

On the other hand, it is possible to turn a feature-based database into a graph. In Figure 2.2, we show the graph constructed from a UCI dataset where the input features are images of 8*-by-*8*-pixel hand-written digits. The graph is constructed by connecting every data point to its* $k(=4)$ *nearest neighbors, where the distance is taken as the Euclidean distance on the raw pixel values, represented as* 64*-dimensional vectors by aligning pixel values in natural sequential orders. In other words, the weight of the edge between node* i *and* j *is*

$$w_{ij} = \mathbb{1}_{\{j \in \mathcal{N}_k(i)\}} + \mathbb{1}_{\{i \in \mathcal{N}_k(j)\}} \in \{0, 1, 2\}$$

where $\mathcal{N}_k(i)$ *is the index set of the* k *nearest neighbors for data point* i*. Here a weight of* 0 *indicates that the corresponding edge does not exist. To better visualize the resulting graph, we use the scores of the first two principal components of the graph Laplacian (to be defined in Section 2.3) as the coordinates of the nodes that represent images. In fact, each cluster represents a single digit label shown by a small image icon (chosen by our* Σ*-optimality active learning criterion).*

Besides the demonstrated unsupervised learning results, graphs are also good places to exercise Semi-Supervised Learning (SSL), which is the problem where part of the graph nodes have actual class labels, e.g., obtained by active queries. The goal in SSL is to infer the correct labels of the remaining nodes where the true labels are hidden from the algorithm. While the prediction task may as well be solved via purely supervised learning, using graphs may improve accuracy by using the density of the unlabeled data points. A good intuitive solution is label propagation [Zhu et al., 2003a], which predicts the label of a node by propagating the labels (or predicted labels) from its neighbors, until reaching a stable solution.



(a) An example unlabeled graph (b) Exploratory queries by Σ-optimality.

Figure 2.2: Active learning on graphs: given (a) with no labels to start with, we aim to design an exploratory set (b) to query for the labels in order to correctly classify the remaining nodes.

Beyond SSL, the true question for active search is how to choose the set of nodes to directly query for their labels, given a querying budget and an objective (e.g., images of a particular digit). To optimally design for queries, active systems typically require a definition of the family of models to be considered or a Bayes prior for probable node label distributions. Here, we use Gaussian Random Fields (GRFs) [Zhu et al., 2003a], which is a natural extension to SSL, which we will discuss in more details in Section 2.3.

2.1.2 Problems Being Solved

Existing active search approaches [Garnett et al., 2012, Vanchinathan et al., 2013, Wang et al., 2013] either lack theoretical guarantees or ignore certain graph properties, thereby degrading empirical performances. We improve on the existing systems by analyzing better exploration designs for GRFs, the Bayesian prior for label distributions. The problem of active search is decomposed into two subproblems:

Active learning (exploration) on GRFs. We consider the problem of designing a good active learning strategy that, under labeling budget constraints, selects which data points to query for labels that are most helpful for classification on a graph-represented database. We assume that the node label distribution is modeled by a GRF with known hyper-parameters. The performance of a specific active learning strategy is measured by the classification accuracy using SSL that is based on label propagation.

Active search (E&E) with GRFs. We assume that the node labels are binary and we aim to find all positive nodes, in a sequential querying framework, using as few query points as possible. Unit reward is granted to every positive query outcome. The performance of an algorithm is measured by the cumulative reward when the sequential process is stopped at any time step. The

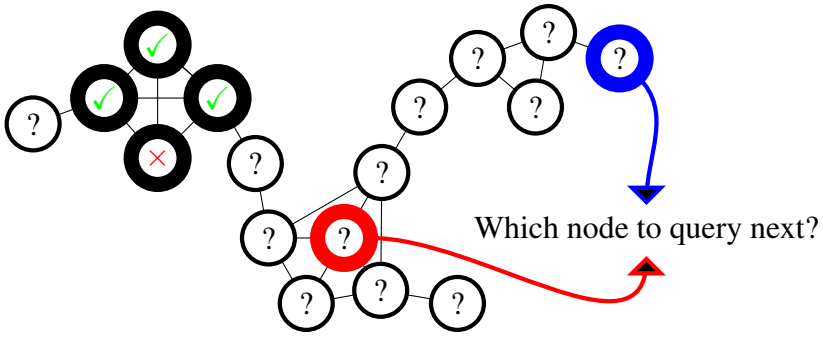


Figure 2.3: Active search problem on a toy graph

performance is measured by cumulative regret, which is the gap between the maximum cumulative outcomes using the optimal designs and the actual cumulative outcomes using our algorithm. We hope to obtain no-regret guarantees, where the average cumulative regret converges to zero, as the number of queries grows to infinity.

2.1.3 Main Contributions

Our main contribution is to show Σ-optimality [Garnett et al., 2012] as a better exploration criterion with GRFs for active learning, active surveying and active search, with theoretical guarantees. We studied in the following aspects to support our claim:

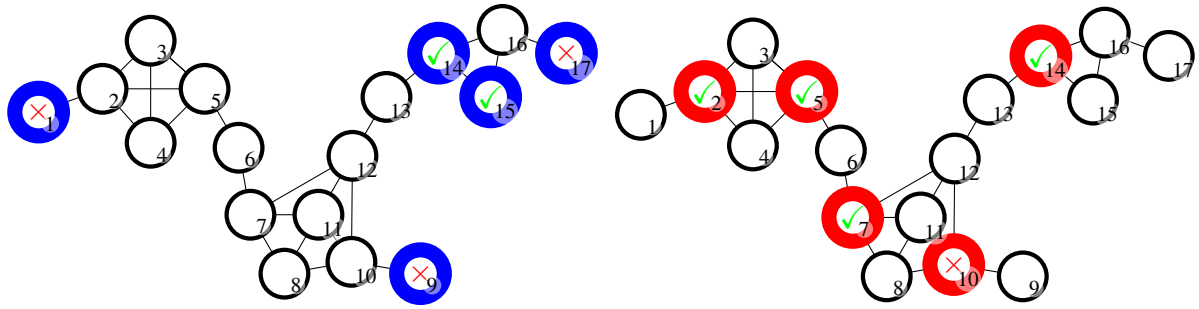
First, a variety of design principles: D-optimality [Krause et al., 2008], V-optimality [Ji and Han, 2012], and Σ-optimality, can be cast in the framework of greedy Bayes-optimal selection rules for active learning on GRFs. However, the design principles are global objectives that measure the entire set of query choices, which may be infeasible to optimize for. We show that greedy, sequential selection of queries is nearly optimal in the optimality with at least $(1 - 1/e)$-ratio. This result was previously unknown for V/Σ-optimality, despite they have better empirical performance. One key insight is that all of the objective functions are monotone and submodular (i.e., later inclusion of a node always provides a diminished return, given all other choices unchanged).

As a corollary, we showed that GRFs are suppressor free. In linear regression, an explanatory variable is called a suppressor if adding it as a new variable enhances correlations between the old variables and the dependent variable [Walker, 2003]. Suppressors are persistent in real-world data. We show GRFs to be suppressor-free. Intuitively, this means that with more labels acquired, the conditional correlation between unlabeled nodes decreases monotonously until their Markov blanket is formed. That GRFs present natural examples for the otherwise obscure suppressor-free condition may be interesting.

For practical active learning on graphs, each objective optimizes for a different surrogate objective, which is unanimously an approximation to the true binary classification objective. Moreover, the GRF prior is a linear relaxation of the true prior distribution that allows only binary

labeling of nodes. Therefore, the best choice of optimality criterion may not be obvious. We conducted thorough experiments on network graphs as well as nearest-neighbor graphs and showed that greedy application of Σ-optimality had the best performance in our experiments. The experiments also included other baselines such as mutual information gain [Krause et al., 2008], uncertainty sampling, and expected error reduction [Settles, 2010, Zhu et al., 2003b].

Active search aims to collect as many positives as quickly. To perform well, a good acquisition rule needs to consider both exploitation (choosing near proven positive nodes) and exploration (finding new areas of positive nodes). The common Gaussian Process-Upper Confidence Bound (GP-UCB) algorithm [Srinivas et al., 2010b] uses marginal variance as the exploration measure. However, when applied to graphs, it tends to select nodes at the periphery of the graph (e.g., leaf nodes) because they have large predictive variance. Yet, the rewards of these nodes reveal little information about the reward distribution over the whole graph. Instead, we propose a new method, GP-SOPT, which uses Σ-optimality as the exploration function. The improvement is visualized in Figure 2.4.



(a) Choices from UCB

(b) Choices by GP-SOPT

Figure 2.4: For the toy graph example, choices from (a) direct application of GP-UCB [Valko et al., 2014, Vanchinathan et al., 2013] versus (b) our vanilla GP-SOPT. We observe that our method (b) tends to select more from cluster centers, which helps reduce variance of the unobserved values/rewards, whereas the previous method (a) tends to select on the graph periphery.

We further showed theoretical guarantees for GP-SOPT in terms of cumulative regrets. Cumulative regret measures the difference in the number of positives found between our algorithm and the optimal active search algorithm given the same number of active search steps. Our guarantee used a newer concept called *effective dimensions* [Valko et al., 2014] that more accurately measures the spectral complexity of the graph. Despite better empirical performance, our theoretical result is at best comparable to the [Valko et al., 2014], if not worse for constant terms. We speculate better results if the complexity assumption is measured in other terms; in the current analysis, it is measured by Shannon entropy (D-optimality) which is an inferior measure of the true complexity.

Finally, we discussed the connection between Σ-optimality and spectral norm minimization, which leads to discussions about choosing other prior models on graphs and possible ongoing acceleration ideas in Chapter 5.

2.2 Related Work

Settles [2010] provided a general introduction to active learning methods in practice. A few principled solutions were discussed, including uncertainty sampling, expected error reduction, variance reduction, etc. The paper was written for active learning, but the ideas are general enough to other similar settings including regressions, with simple modifications. On the other hand, there was a lack of graph-based solutions.

A more concrete example was considered by Krause et al. [2008] for sensor placement in a large area. The measurement values are measured by a Gaussian process (GP) [Rasmussen and Williams, 2006] and information gain (i.e., D-optimality) is used as the design criterion. One of the reasons for choosing information gain is to have near-optimal global guarantees on the final design if the sensors are placed in a sequential greedy manner. Despite the theoretical motivation, Krause et al. [2008] noted the issue that the outcomes of plain information gain criteria tend to select queries at the boundary of the environment and provided a fix by altering the design criterion to use mutual information gain.

When the Bayes prior is limited to GRFs, Ji and Han [2012] proposed greedy variance minimization (which we call V-optimality) as a cheap and high profile surrogate active classification criterion. To decide which node to query next, the active learning algorithm finds the unlabeled node which leads to the smallest average predictive variance on all other unlabeled nodes. Experiments on citation networks were used to justify for the greedy algorithm. However, the motivation of the objective was little discussed, nor were there any theoretical guarantees. In fact, we show that V-optimality has the same types of near-optimality global guarantees as [Krause et al., 2008], when limited to GRFs (as opposed to GPs). Completing the picture, [Krause et al., 2008] showed a counter-example for similar guarantees of V-optimality in general GPs.

The problem of active surveying and our contribution of Σ-optimality were earlier discussed by [Garnett et al., 2012]. Here, however, the problems were discussed in low-dimensional Euclidean spaces where GP priors are more natural choices. Since the solution was based on variance reduction, like the counter-example in Krause et al. [2008], no global guarantees were provided. Instead, the authors used a multiple-step look-ahead method accompanied with subtree-pruning techniques.

On the global optimality for the greedy approaches, a key result from Nemhauser et al. [1978], shows that any *submodular and monotone* set function yields a $(1 - 1/e)$ global optimality guarantee for greedy solutions. Our proof results coincide with Friedland and Gaubert [2011], but we used different principles and were not restricted to spectral functions.

Garnett et al. [2012] also motivated active search and later Wang et al. [2013] extended the settings to graphs, where GRF priors were used. Despite decent empirical performance, the solutions, which also used multi-step look-ahead planning with pruning, do not hold any theoretical guarantees.

Vanchinathan et al. [2013] proposed a GP-based algorithm, GP-SELECT, for sequentially selecting instances with high user scores or ratings (rewards). This algorithm extends the popular

GP-UCB algorithm [Auer, 2003, Cox and John, 1997] for stochastic optimization and inherits nice theoretical guarantees [Srinivas et al., 2010b].

Valko et al. [2014] considered bandit problems where arms correspond to nodes on a graph and the rewards form a smooth function over the graph. Their algorithm can be viewed as a special case of GP-UCB with a kernel defined by the inverse of a graph Laplacian (augmented with an identity matrix). To analyze the performance of their GP-UCB-style algorithm, they propose the notion of *effective dimension* of a graph, which can be viewed as a measure of the spectral decay of the graph kernel, thereby determining, the performance of the algorithm Srinivas et al. [2010b]. Our solution is different but we also use the effective dimension to analyze our proposed methods. Other recent developments on active learning and search include Chen et al. [2014], Gadde and Ortega [2015], Jun and Nowak [2016], Liu et al. [2015], Wang et al. [2016]

2.3 Background

There are many ways to use graph connectivity information. We will explore the idea of using energy-based models that are generally known as random fields, specifically Gaussian random fields (GRFs). To build intuitions, we will explain why GRFs naturally leads to label propagation in Semi-Supervised Learning (SSL) settings. However, our focus in active search requires us to also pay attention to the uncertainty measures that distinguish GRFs from label propagation.

2.3.1 Gaussian Random Fields (GRFs)

We use the Gaussian random fields on graphs as described in [Zhu et al., 2003a]. Let $G = (V, W)$ represent an undirected graph with n nodes, where each node v_i has an (either known or unknown) label value f_i and each edge w_{ij} has a fixed nonnegative weight that reflects the proximity, similarity, etc, between nodes v_i and v_j (recall the handwritten digits example in chapter introduction). The value f_i is unknown at first and can be revealed only when it is queried explicitly. There are two ways to model label observations: one assuming that the labels are directly observable, while the other assuming that the observations have additive Gaussian noise:

$$y_i = f_i, \quad \text{or} \quad y_i = f_i + \varepsilon, \text{ where } \varepsilon \sim \mathcal{N}(0, \sigma^2). \tag{2.1}$$

The first observation model is equivalent to the second when taking $\sigma \to 0$.

We relax f_i to real values, $f_i \in \mathbb{R}, \forall i$. GRF generates them according to a joint distribution on

the node values, which we represent by a vector $\mathbf{f} = (f_1, \ldots, f_n)^\top$, using the energy function

$$E(\mathbf{f}) = \frac{1}{2}\sum_{i\sim j} w_{ij}(f_i - f_j)^2 + \frac{1}{2}\omega_0 \sum_{i=1}^{n}(f_i - \bar{\mu}_0)^2 \tag{2.2}$$

$$= \frac{1}{2}(\mathbf{f} - \bar{\boldsymbol{\mu}}_0)^\top(\mathbf{D} - \mathbf{W} + \omega_0\mathbf{I})(\mathbf{f} - \bar{\boldsymbol{\mu}}_0)$$

$$= \frac{1}{2}(\mathbf{f} - \bar{\boldsymbol{\mu}}_0)^\top\bar{\mathbf{L}}(\mathbf{f} - \bar{\boldsymbol{\mu}}_0), \tag{2.3}$$

where "$i \sim j$" indicates that node v_i is directly connected to v_j on the graph and $\bar{\mu}_0$ is a prior mean value, set at the average class proportion of positives. Eq (2.3) puts (2.2) in vector/matrix forms, where $\mathbf{W} = (w_{ij})_{i,j=1}^n$ is the weight matrix such that the (i,j)-th element is w_{ij}, $\mathbf{D} = \mathrm{diag}(\mathbf{W1}) = \mathrm{diag}(\sum_j w_{1j}, \ldots, \sum_j w_{nj})$, $\bar{\boldsymbol{\mu}}_0 = (\bar{\mu}_0, \ldots \bar{\mu}_0)^T$, and $\mathbf{I}$ is the identity matrix. Matrix $\bar{\mathbf{L}} = \mathbf{D} - \mathbf{W} + \omega_0\mathbf{I}$ is called *augmented graph Laplacian matrix*. Define $\bar{\mathbf{C}} = \bar{\mathbf{L}}^{-1}$.

GRF prior. The higher the energy $E(\mathbf{f})$ for a choice of $\mathbf{f}$, the more improbable $\mathbf{f}$ is to be generated. This intuition can be modeled by a multivariate normal prior distribution using the negative energy as its potential,

$$p(\mathbf{f}) = \frac{\exp(-E(\mathbf{f}))}{(2\pi)^{\frac{n}{2}}(\det(\bar{\mathbf{L}}))^{\frac{1}{2}}} \quad \Leftrightarrow \quad \log p(\mathbf{f}) \simeq -E(\mathbf{f}), \tag{2.4}$$

where "$\simeq$" hides the normalization constant that turns (2.4) into a proper probability distribution.

Posterior distribution. GRF describes a world generation process using (2.1)&(2.4). However, the true values of $\mathbf{f}$ is only one draw from the prior distribution. When observations are made at a set of nodes $v_{s_1}, v_{s_2}, \ldots, v_{s_t}$, we need to update the Bayes belief to a posterior distribution. Let $S_t = \{s_1, \ldots, s_t\} \subset V$ be the index set of node queries and let $\mathbf{y}_{S_t} = (y_{s_1}, \ldots, y_{s_t})^\top$ be the observation outcomes in the corresponding order; GRF will update its posterior model to

$$\log p_t(\mathbf{f}) = \log p(\mathbf{f} \mid S_t, \mathbf{y}_{S_t}) \simeq \log p(\mathbf{f}) + \sum_{\tau=1}^{t} \log p(y_{s_\tau} \mid f_{s_\tau})$$

$$\simeq -\frac{1}{2}\sum_{i\sim j} w_{ij}(f_i - f_j)^2 - \frac{1}{2}\omega_0\sum_{i=1}^{n}(f_i - \bar{\mu}_0)^2 - \frac{1}{2\sigma^2}\sum_{\tau=1}^{t}(y_{s_\tau} - f_{s_\tau})^2. \tag{2.5}$$

Notice, (2.5) is a multivariate normal distribution with a different mean vector and a different covariance matrix.

2.3.2 GRF Posteriors

Posterior Mean Solves Semi-Supervised Learning (SSL)

After obtaining part of the node values, SSL aims to predict the remaining node values. One natural solution is to use label propagation, which iteratively propagates the known values or

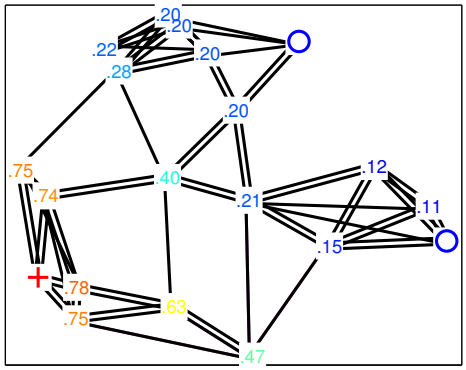


Figure 2.5: SSL example. Red "+" and blue "◯" are the only provided supervisions. The number on every node is the chance that it belongs to class "+", predicted by label propagation.

previously propagated values to neighboring nodes. In this way, the final stable node values will be influenced by the structure of the graph. Mathematically, label propagation uses iterative assignments to find the stable point in the following,

$$\begin{cases} \mu_i = y_{s_\tau}, & \text{if } i = s_\tau \in S, \\ d_i \mu_i = \sum_{j \sim i} w_{ij} \mu_j, & \text{otherwise}, \end{cases}$$

where $i = s_\tau \in S$ indicates that the node v_i is queried at step τ and labeled as y_{s_τ}. Labeled nodes are not changed during the iterative assignments, whereas the remaining nodes keep updating according to the mean value in their neighbors until convergence. It is intuitive that label propagation must converge when $w_{ij} \geq 0, \forall i, j$, and the solution must observe $0 \leq \mu_i \leq 1, \forall i$, if all labels are within $[0, 1]$.

How does label propagation relates to GRFS?

By setting the gradient in the GRF posterior (2.5) to zero, we may find that the posterior mean, i.e., the max-a-posteriori estimate of the GRF posterior, solves an augmented version of label propagation,

$$\begin{cases} \left(\frac{1}{\sigma^2} + \omega_0 + d_i\right)\mu_i = \frac{1}{\sigma^2} y_{s_\tau} + \sum_{j \sim i} w_{ij} \mu_j + \omega_0 \bar{\mu}_0, & \text{if } i = s_\tau \in S, \\ (\omega_0 + d_i)\mu_i = \sum_{j \sim i} w_{ij} \mu_j + \omega_0 \bar{\mu}_0, & \text{otherwise}, \end{cases}$$

where if we take $\sigma \to 0$ and $\omega_0 = 0$, the solution will be the exact label propagation.

Thus, GRF posterior distribution can be seen as a full Bayesian extension to label propagation. Moreover, GRFs additionally provides covariance matrices to measure the full model uncertainty.

Covariance Matrix

When label propagation makes predictions for SSL, it ignores the certainty of the prediction itself. For example, a prediction value of $\mu_i = 0.5$ can mean either an *approximation error*, if the node directly connects to two nodes with different labels, or an *estimation error*, if the result of 0.5 is due to the node being far from all other labeled nodes. There is no easy way to improve on the former case unless we change the model that describes node value distributions, i.e., by changing the graph itself. On the other hand, the latter can be improved if we change the label queries to be close to μ_i. For general active learning, we want to choose queries to be close to all unlabeled nodes.

Since GRF posterior model is a multivariate normal distribution, its covariance matrix is an effective way to measure how far each node is to the labeled nodes. In fact, the marginal posterior variance on the node variable f_i shows the graph commute time from node v_i to any of the labeled nodes using random walks [Doyle and Snell, 1984]. Another intuitive analogy uses spring network systems. If all the graph nodes are masses connected by springs according to the graph edges, after pinning the queried nodes at their label values, the stiffness of the unlabeled nodes, i.e., the certainty of the prediction mean, will be inversely proportional to the marginal variance reflected in (2.5). The farther a node is to the labeled nodes, the less stiff the corresponding mass is and also the larger its posterior marginal variance becomes. The posterior correlation between any pair of variables f_i, f_j can also find analogy in the spring mass system, as how much displacement one node has if the other node is displaced by a unit distance.

Notice, the prior covariance matrix is properly defined if the augmentation coefficient $\omega_0 > 0$. $\omega_0\mathbf{I}$ is considered an *augmentation* matrix because it effectively builds a weak connection between every node and the prior mean $\bar{\mu}_0$, such that the prior model has full rank.

Explicit Solutions in Matrix Form

For convenience, we can rewrite (2.5) in matrix form. Recall $\bar{\mathbf{L}} = \mathbf{D} - \mathbf{W} + \omega_0\mathbf{I}$; let $\mathbf{e}_{s_\tau} = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$ be an indicator vector whose nonzero is at index s_τ, the posterior distribution becomes

$$\begin{aligned}\log p_t(\mathbf{f}) &\simeq -\frac{1}{2}(\mathbf{f} - \bar{\boldsymbol{\mu}}_0)^\top \bar{\mathbf{L}}(\mathbf{f} - \bar{\boldsymbol{\mu}}_0) - \sum_{\tau=1}^{t} \frac{1}{2\sigma^2}(y_{s_\tau} - f_{s_\tau})^2 \\ &\simeq -\frac{1}{2}\mathbf{f}^\top \left(\bar{\mathbf{L}} + \frac{1}{\sigma^2}\sum_{\tau=1}^{t} \mathbf{e}_{s_\tau}\mathbf{e}_{s_\tau}^\top\right)\mathbf{f} + \left(\bar{\mathbf{L}}\bar{\boldsymbol{\mu}}_0 + \frac{1}{\sigma^2}\sum_{\tau=1}^{t} y_{s_\tau}\mathbf{e}_{s_\tau}\right)\mathbf{f} \qquad (2.6)\end{aligned}$$

Let $\boldsymbol{\mu}^{(t)}$ and $\mathbf{C}^{(t)}$ be the posterior mean vector and covariance matrix, the explicit solution to

GRF posterior is

$$\boldsymbol{\mu}^{(t)} = \mathbf{C}^{(t)} \left(\bar{\mathbf{L}} \bar{\boldsymbol{\mu}}_0 + \frac{1}{\sigma^2} \sum_{\tau=1}^{t} y_{s_\tau} \mathbf{e}_{s_\tau} \right)$$
$$\mathbf{C}^{(t)} = \left(\bar{\mathbf{L}} + \frac{1}{\sigma^2} \sum_{\tau=1}^{t} \mathbf{e}_{s_\tau} \mathbf{e}_{s_\tau}^\top \right)^{-1} \tag{2.7}$$

When $\sigma \to 0$, the posterior distribution has zero covariance on queried variables, but is still properly defined on the remaining variables. Without loss of generality, assume S_t corresponds to the first t nodes in all nodes V; the corresponding posterior covariance matrix becomes

$$\boldsymbol{\mu}^{(t)} = \begin{pmatrix} \mathbf{y}_{S_t} \\ -\left(\bar{\mathbf{L}}_{U_t U_t}\right)^{-1} \bar{\mathbf{L}}_{U_t S_t} \mathbf{y}_{S_t} \end{pmatrix}, \quad \mathbf{C}^{(t)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \left(\bar{\mathbf{L}}_{U_t U_t}\right)^{-1} \end{pmatrix}, \tag{2.8}$$

where $U_t = V \setminus S_t$ is the index set of the unlabeled nodes. Notice, $\boldsymbol{\mu}_{U_t}^{(t)}$ remains nonnegative because $\bar{\mathbf{L}}_{U_t S_t}$ is the off-diagonal block whose elements are non-positive. In fact, in the appendix we show that $\mu_i^{(t)} \in [0, 1], \forall i \in U_t$, if the labels allow $y_{s_\tau} \in [0, 1], \forall s_\tau \in S_t$.

Let $\mathbf{C} = \left(\bar{\mathbf{L}}_{U_t U_t}\right)^{-1}$ and $\tilde{\mathbf{C}} = (\bar{\mathbf{L}}_{U_{t+1} U_{t+1}})^{-1}$ and without loss of generality, suppose s_t is positioned as the last node. By Shur's Lemma (or GP-regression update rule [Rasmussen and Williams, 2006]), the following can be verified,

$$\begin{pmatrix} \tilde{\mathbf{C}} & \mathbf{0} \\ \mathbf{0} & 0 \end{pmatrix} = \mathbf{C} - \frac{1}{C_{s_t, s_t}} \cdot \mathbf{C}_{:, s_t} \mathbf{C}_{s_t, :}. \tag{2.9}$$

In general, with $\sigma > 0$, similar incremental update rules can be derived by following GP literature:

$$\boldsymbol{\mu}^{(t+1)} = \boldsymbol{\mu}^{(t)} + \mathbf{C}_{:, s_t}^{(t)} \left(C_{s_t, s_t}^{(t)} + \sigma^2 \right)^{-1} \left(y_{s_t} - \mu_{s_t}^{(t)} \right) \tag{2.10}$$
$$\mathbf{C}^{(t+1)} = \mathbf{C}^{(t)} - \mathbf{C}_{:, s_t}^{(t)} \left(C_{s_t, s_t}^{(t)} + \sigma^2 \right)^{-1} \mathbf{C}_{s_t, :}^{(t)}. \tag{2.11}$$

The above rule also applies to increments with multiple observations $(y_{s_t}, y_{s_{t+1}}, \ldots, y_{s_{t+\tau}})$, if one replaces the element subscriptions with sub-matrix subscriptions.

Finally, for notation convenience, we may also write the posterior mean and covariance as functions, i.e., $\mu_t(v_i) = \mu_i^{(t)}$ and $C_t(v_i, v_j) = C_{i,j}^{(t)}$. Similarly, the variables or labels may also take either vector or function forms, i.e., $f(v_i) = f_i$ and $y(v_i) = y_i$.

2.4 Methods for Active Learning and Surveying

We begin introducing new methods with novel exploratory query designs, which solves half of the problem in active search. Using GRFs, we relax the node labels to real values and build the

joint distribution of node values as a multivariate normal distribution. Effectively, the problem is reduced to an optimal design problem, which aims to minimize model uncertainty (using some measure of the posterior covariance matrix) after collecting a set of queries in a multi-step lookahead manner. We will examine several surrogate design criteria and motivate our own version of Σ-optimality.

2.4.1 Minimization on Surrogate Objectives

All of the following surrogate loss functions are defined as a set function $R(S_t)$, whose input is S_t, the set of node indices for the first t queries, and whose output is an objective to be minimized. All surrogate objectives take the form:

$$\min_{S_t} R(S_t) \quad \text{s.t.} \quad |S_t| \leq t, S_t \subset V \tag{2.12}$$

When can be inferred from context, we use $U_t = V \setminus S_t$ to denote the indices of the unlabeled nodes. Let $p_t(\mathbf{f})$ indicate the posterior distribution after selecting the set S_t with size t.

D-Optimality for Differential Entropy Minimization

To reduce the overall model uncertainty, a natural idea is to decrease the differential entropy of the full GRF posterior, which causes the full posterior distribution to concentrate around its posterior mean, i.e., the SSL predictions via label propagation. Minimizing differential entropy is also known as D-optimality in regression designs, because it minimizes the determinant of the posterior covariance matrix. According to (2.6),

$$R_D(S_t) = H(p_t(\mathbf{f})) \simeq \frac{1}{2} \log\det(\mathbf{C}^{(t)}),$$

where the normalization constants are ignored. We use subscript D to indicate D-optimality, which is a popular choice for exploratory measures in [Gotovos et al., 2013, Krause et al., 2008, Srinivas et al., 2010a, Valko et al., 2014].

A potential issue is that, while D-optimality aims to reduce the entropy of the paremeters $\mathbf{f}$, its greedy application is equivalent to selecting nodes with the largest marginal variance:

$$\arg\min_s H(\mathbf{f} \mid \mathbf{y}_{S_t \cup \{s\}}) = \arg\max H(y_s \mid \mathbf{y}_{S_t}) = \arg\max_s \ \text{Var}(y_s \mid \mathbf{y}_{S_t}).$$

Greedy application turns D-optimality to a no-step lookahead algorithm; in the early phase of queries, the optimal solutions may often be found on the boundary of the environment, where the marginal variance is the largest. Figure 2.6 shows a "successful" application of D-optimality-based algorithm, where the initial query points are mostly on the boundary of the environment.

When applied to graphs, the issue is more severe, because graphs usually have a larger boundary due to its high intrinsic dimensionality and very different eigenvalue distribution. Figure 2.7

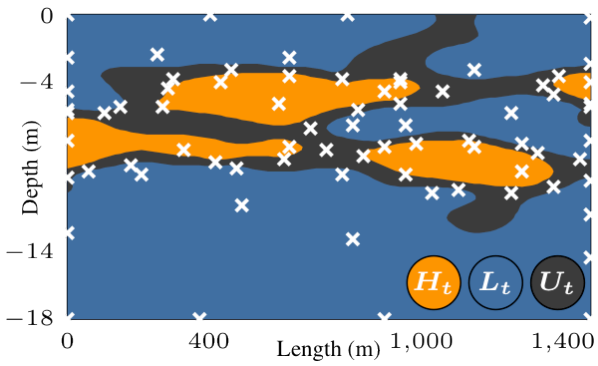


Figure 2.6: Pathology in D-optimality: many query points are on the boundary of the environment before they appear at the center where true exploration should happen. Example from [Gotovos et al., 2013].

visualizes the choices of greedy D-optimality on DBLP coauthorship graph[1]. The nodes represent scholars and the weighted edges are based on the number of papers bearing both scholars' names. Visualization is due to OpenOrd layout [Martin et al., 2011], where the dense areas indicate graph clusters. The node colors show the true labels based on the research area of the authors, which is not used by the designs and shown to visually validate our GRF assumption. Here, D-optimality focuses on the periphery of the graph, choosing many leaf nodes, which is not ideal for exploration.

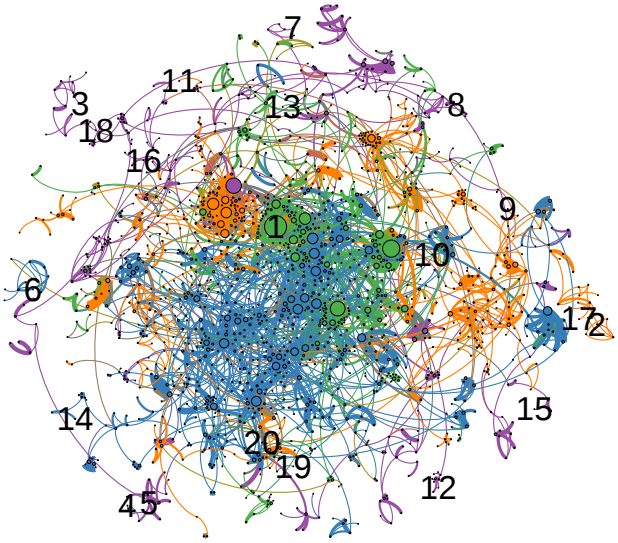


Figure 2.7: D-optimality chooses boundaries (e.g., leaf nodes) in a graph.

[1] http://www.informatik.uni-trier.de/~ley/db/

V-Optimality for ℓ_2 Risk Minimization

Another objective is to directly minimize the ℓ_2 risk on the independent node predictions. We use Bayes risk,

$$R_V(S_t) = \mathbb{E}\left[\mathbb{E}\left[\sum_{i=1}^{n}(y_i - f_i)^2 \,\middle|\, \mathbf{y}_{S_t}\right]\right] = \operatorname{tr}(\mathbf{C}^{(t)}), \tag{2.13}$$

where, for simplicity, the summation is over all nodes including both labeled and unlabeled. Notice, when $\sigma \to 0$, the objective is equivalent to summation only on the unlabeled node set, because by (2.8), we have $\operatorname{tr}(\mathbf{C}^{(t)}) = \operatorname{tr}((\bar{\mathbf{L}}_{U_t,U_t})^{-1})$.

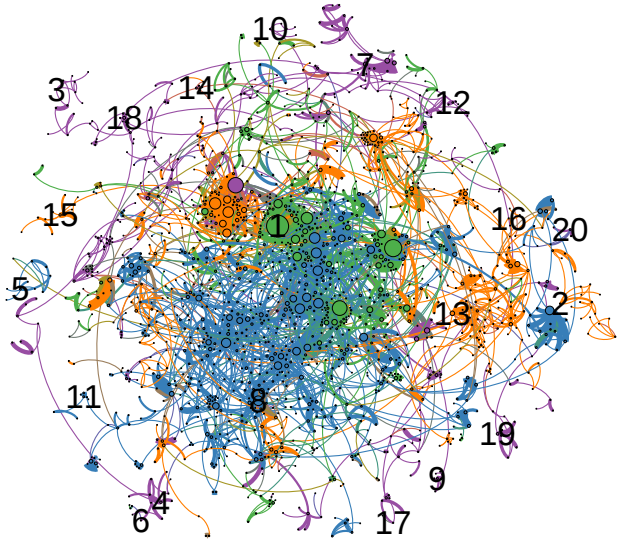


Figure 2.8: V-optimality improves the exploration, but the choices are still not central enough.

Ji and Han [2012] used a similar objective which they call *variance minimization*. The optimality may also be called A-optimality, because $\mathbf{f}$ is both the set of model parameters and the prediction values according to the GRF model.

The greedy application of V-optimality is shown in (2.15), which evaluates global influence of queries and thus is a true lookahead measure. However, the visualization in Figure 2.8 does not seem central enough. Can we do better?

Σ-Optimality for Survey Risk Minimization

Besides active learning, a different task we also consider is active surveying. Surveying aims to determine the proportion of nodes belonging to each class. It usually uses fewer queries than active learning.

For active surveying, the Bayes risk is:

$$R_\Sigma(S_t) = \mathbb{E}\left[\mathbb{E}\left[\left(\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} f_i\right)^2 \middle| \mathbf{y}_{S_t}\right]\right] = \mathbf{1}^\top \mathbf{C}^{(t)} \mathbf{1}, \tag{2.14}$$

where, for simplicity, the summation is also over all nodes. When $\sigma \to 0$, the objective is equivalent to summation only on the unlabeled nodes, because by (2.8), we have $\mathbf{1}^\top \mathbf{C}^{(t)} \mathbf{1} = \mathbf{1}^\top (\bar{\mathbf{L}}_{U_t,U_t})^{-1} \mathbf{1}$.

Further, we will also consider the application of the Σ-optimality in active learning because (2.14) is a valid metric on the predictive variance. Surprisingly, although both (2.13) and (2.14) are approximations of the real objective (the 0/1 risk), greedy reduction of the Σ-optimality outperformed greedy reduction of the V-optimality in active classification, as well as several other methods including expected error reduction. In Figure 2.9, we may also visualize that Σ-optimality indeed explores at the cluster centers, producing the most amount of information compared to the alternative D/V-optimality.

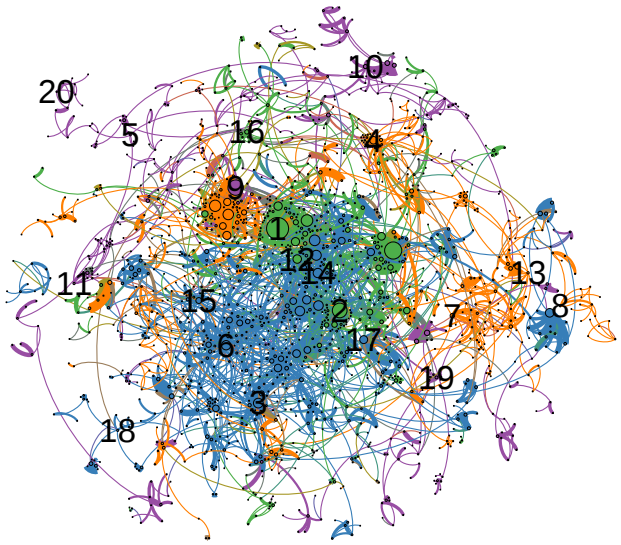


Figure 2.9: Σ-optimality finally explores at the cluster centers, visually producing the most effective designs.

2.4.2 Greedy Application of D, V, and Σ-Optimality

Calculating the global optimum (2.12) with any of the objectives may be intractable. As will be shown later in the theoretical results, all objectives are submodular set functions and the greedy sequential update algorithm (Algorithm 2.1) yields a solution that has guaranteed approximation ratio to the optimum (Theorem 2.2).

Algorithm 2.1 Greedy subset selection.

Input: Graph Laplacian $\bar{\mathbf{L}}$, objective function $R(\cdot)$, budget T.
Output: A subset $S_T = \{s_1, \ldots, s_T\} \subset V$ by greedy selection.
for $t = 1, 2, \ldots, T$ **do**
  Find $s_t = \arg\min_v G_{t-1}(v)$
  Update posterior distribution by (2.10) and (2.11)
end for

The following applies Algorithm 2.1 to our specific objective functions. At iteration $t+1$, with an already obtained set S_t, define

$$G_t(v) = R(\{s_1, \ldots, s_t\}) - R(\{s_1, \ldots, s_t, v\})$$

Notice $R(\cdot)$ is a function on the posterior covariance matrix. Let $\mathbf{C} = \mathbf{C}^{(t)} = (C_t(v_i, v_j))_{i,j=1}^n$ and further denote $C_t(v_i, v_j) = \rho_t(v_i, v_j)\sigma_t(v_i)\sigma_t(v_j)$. The incremental update rule (2.11) yields

$$\mathbf{C} - \mathbf{C}^{(t+1)} = \mathbf{C}_{:,v} \left(\sigma_t^2(v) + \sigma^2\right)^{-1} \mathbf{C}_{v,:}.$$

For D-optimality, we then have,

$$G_{D,t}(v) = I(\mathbf{f}; y(v)) = H(y(v)) - H(y(v) \mid \mathbf{f}) = \frac{1}{2} \log\det(\sigma^2 + \sigma_t^2(v)) - \log\det(\sigma),$$

where $I(\mathbf{f}; y_{s_t})$ is the mutual information and all measures are with respect to distribution $p_{t-1}(\mathbf{f})$.

We can also put (2.9) inside $R_\Sigma(\cdot)$ and $R_V(\cdot)$ to get the following equivalent criteria:

$$D\textbf{-optimality}: \ \arg\max_{v \in U_t} \ G_{D,t}(v) = \log\det(1 + \sigma^{-2}\sigma_t^2(v)) \mapsto \sigma_t^2(v)$$

$$V\textbf{-optimality}: \ \arg\max_{v \in U_t} \ G_{V,t}(v) = \frac{\sum_{j=1}^n (C_t(v, v_j))^2}{\sigma_t^2(v) + \sigma^2} \mapsto \sum_{v' \in U_t} \rho_t(v, v')^2 \sigma_t(v')^2 \quad (2.15)$$

$$\Sigma\textbf{-optimality}: \ \arg\max_{v \in U_t} \ G_{\Sigma,t}(v) = \frac{(\sum_{j=1}^n C_t(v, v_j))^2}{\sigma_t^2(v) + \sigma^2} \mapsto \sum_{v' \in U_t} \rho_t(v, v')\sigma_t(v'). \quad (2.16)$$

where the right side of the mappings take $\sigma \to 0$ and are equivalent in terms of having the same solution for "argmax".

Remark: Let $G_t(v) = g_t^2(v)$, we may generalize the V- and Σ-optimality to a broader class of λ_p-optimalities:

$$(\lambda_p\text{-optimality}): \ \arg\max_{s_t \in U_t} g_{\lambda_p,t}^p(v) = \sum_{v' \in U_t} (\rho_t(v, v')\sigma_t(v'))^p$$

where V-optimality corresponds to $p = 2$ and Σ-optimality $p = 1$ (up to the same optimizer).

2.4.3 Comparing the Greedy Applications of the Σ and V-Optimality

Both the V/Σ-optimality are approximations to the 0/1 risk minimization objective. Unfortunately, we cannot theoretically reason why Σ-optimality outperformed V-optimality in our experiments. Nonetheless, some observations may be helpful.

Eq. (2.15) and (2.16) suggest that both the greedy Σ/V-optimality selects nodes that (1) have high variance and (2) are highly correlated to high-variance nodes, conditioned on the labeled nodes.

The difference between V and Σ-optimality lies in the measure to evaluate global influence. While V-optimality naturally chooses ℓ_2-measure based on the optimal designs for regression problems, Σ-optimality realizes ℓ_1-measure that may be more robust to large values. Since GRFs are continuous relaxations to the true binary label distribution, approximation errors can influence design choices. Specially, at the boundary nodes, the (posterior) marginal variance can be unbounded large. By taking ℓ_1-measure for influence, Σ-optimality can obtain additional robustness against the modeling error. Additional visualizations comparing the choice of V and Σ-optimality may be found in Appendix A.3.

2.5 Methods for Active Search

Algorithm 2.2 General GP-style Active Search

Input: Graph laplacian $\bar{\mathbf{L}}$, desired number T of nodes to be selected, α_t, and σ.
Output: Query selections $S_T = \{s_1, \ldots, s_T\}$.
for $t = 1, \ldots, T$, **do**
  $s_t := \arg\max_{v \in U_{t-1}} \mu_{t-1}(v) + \alpha_t g_{t-1}(v)$.
  Query the label y_{s_t} of s_t.
  Update μ_t and C_t by (2.10) and (2.11).
end for

We propose active search algorithms follow the general GP-style template in Algorithm 2.2. At iteration $t+1$, Algorithm 2.2 selects the next node to query based on a deterministic selection rule of the form:

$$\underset{s_t \in U_{t-1}}{\arg\max}\ \mu_{t-1}(s_t) + \alpha_t \cdot g_{t-1}(s_t), \tag{2.17}$$

where $\mu_{t-1}(s_t)$ is the usual exploitation term and $g_{t-1}(s_t)$ encourages exploration, with the two being balanced by a possibly iteration-dependent parameter $a_t > 0$.

Examples from existing literature like the popular GP-UCB algorithm and its extension to active search, GP-SELECT [Vanchinathan et al., 2013], amount to setting $g_t(v) = \sigma_t(v)$, the predictive variance of node v. Although this is a very reasonable choice in many situations, it may lead to undesirable exploration behaviors on graphs. Under our model assumption, low-degree nodes, which usually lie at the periphery of a graph, tend to have high predictive variances. Direct

applications of GP-UCB may result in the selection of many such outliers, which fail to reveal much information about the reward values of most other nodes at the core of the graph.

Intuitively, a good exploration criterion should favor nodes that have high influences on other parts of the graph. That is, the knowledge of the function values at these nodes should reveal a lot about the function values at other nodes. Under our model assumption, this principle naturally connects with the predictive covariances of a node with others. Research in active learning on graphs has already made use of predictive covariances to construct better selection rules. Ji and Han [2012] proposed to select nodes based on their sums of squares of predictive covariances with other nodes, which is derived from the minimization of squared prediction error, known as V-optimality in experiment design. Our previous section reviewed that V-optimality can still be undesirably sensitive to outliers and proposed the Σ-optimality criterion:

$$g_t^2(v) = \frac{\left(\sum_{v' \in V} C_t(v, v')\right)^2}{\sigma_t^2(v) + \sigma^2}, \tag{2.18}$$

We propose three exploitation-exploration style algorithms with exploration criteria motivated by Σ-optimality, which are *vanilla* Σ-optimality and its two variants with an additional parameter k for theoretical justifications. All algorithms select the next node to query by the general rule (2.17), but with different exploration terms:

GP-SOPT (Vanilla Σ-Optimality):

$$g_t(v) = \frac{1}{\sqrt{1 + \frac{\sigma^2}{\sigma_t^2(v)}}} \sum_{v' \in V} \rho_t(v, v')\sigma_t(v').$$

GP-SOPT.TT (Thresholded Total Covariance):

$$g_t(v) = \min\left(k\sigma_t(v), \sum_{v' \in V} \rho_t(v, v')\sigma_t(v')\right).$$

GP-SOPT.TOPK (Top-k Covariance):

$$g_t(v) = \max_{B \subset V, |B| = k} \sum_{v' \in B} \rho_t(v, v')\sigma_t(v').$$

As one can see from Figure 2.4, the nodes selected by vanilla GP-SOPT indeed reside in more central parts of the toy graph than the nodes selected by its competitor, GP-UCB. In a large graph with many peripheral nodes, we believe that the improved exploration criteria of GP-SOPT and its variants contribute to a better recall rate of search targets in real graphs.

The reason we propose the latter two variants is to both address proof difficulties and increase practical robustness.

2.6 Theoretical Properties

2.6.1 Greedy Variance Reduction

For the general GP model, greedy optimization the ℓ_2 risk has no guarantee that the solution can be comparable to the brute-force global optimum (taking exponential time to compute), because the objective function, the trace of the predictive covariance matrix, fails to satisfy submodularity in all cases [Krause et al., 2008]. However, in the special case of GPs with kernel matrix equal to the inverse of an augmented graph Laplacian, GRFs do provide such theoretical guarantees, both for V and Σ-optimality. The latter is a novel result.

We reuse $G(\cdot)$ as a set function showing the decrease in various criteria, $G(S) = R(\emptyset) - R(S)$ for either $R_V(S)$ or $R_\Sigma(S)$. The following results concern greedy maximization of $G(S)$:

Theorem 2.2 (Near-optimal guarantee for greedy applications of V/Σ-optimality)**.** *In risk reduction,*

$$G(\hat{S}) \geq (1 - 1/e) \cdot G(S^*), \tag{2.19}$$

where $G(S) = R(\emptyset) - R(S), \forall S \subset V$, for either $R(S) = R_V(S)$ or $R_\Sigma(S)$, e is Euler's number, $\hat{S}$ is the greedy optimizer, and S^* is the true global optimizer under the constraint $|S^*| \leq |\hat{S}|$.[2]

According to Nemhauser et al. [1978], it suffices to show the following properties of $G(S)$:

Lemma 2.3 (Normalization, Monotonicity, and Submodularity)**.** $\forall S_1 \subset S_2 \subset V, v \in V$,

$$G(\emptyset) = 0, \tag{2.20}$$

$$G(S_2) \geq G(S_1), \tag{2.21}$$

$$G\big(S_1 \cup \{v\}\big) - G(S_1) \geq G\big(S_2 \cup \{v\}\big) - G(S_2). \tag{2.22}$$

Another sufficient condition for Theorem 2.2, which is itself an interesting observation, is the *suppressor-free* condition. Walker [2003] describes a *suppressor* as a variable, knowing which will suddenly suppress a strong correlation between the predictors. An example is $y_i + y_j = y_k$. Knowing any one of these will suppress correlations between the others. Walker further states that suppressors are common in regression problems. Das and Kempe [2008] extend the suppressor-free condition to sets and showed that this condition is sufficient to prove (2.13). Formally, the condition is:

$$\big|\mathrm{corr}(y_i, y_j \mid S_1 \cup S_2)\big| \leq \big|\mathrm{corr}(y_i, y_j \mid S_1)\big|, \quad \forall v_i, v_j \in V, \forall S_1, S_2 \subset V. \tag{2.23}$$

In fact, it may be easier to understand (2.23) as a decreasing correlation property. It is well known for Markov random fields that the labels of two nodes on a graph become independent if conditioned on their Markov blanket. Here we establish that GRF boasts more than that: the

[2] The results (2.20)–(2.19) can be extended to nonuniform node costs. Denote c_v as the node cost of $v \in V$. In this case, a corresponding greedy algorithm maximizes the marginal risk reduction divided by the marginal cost and the constraint in (2.19) becomes $\sum_{v \in S^*} c_v \leq \sum_{v \in \hat{S}} c_v$

correlation between any two nodes decreases as more nodes get labeled, even before a Markov blanket is formed. To summarize, we have:

Theorem 2.4 (Suppressor-Free Condition)**.** (2.23) *holds for pairs of nodes in the* GRF *model.*

2.6.2 Regret Analysis

We present an UCB-style analysis for GP-SOPT.TT and GP-SOPT.TOPK, and an analysis based on Contal et al. [2014] for GP-SOPT. We combine several results on GP optimization [Contal et al., 2014, Srinivas et al., 2012, Vanchinathan et al., 2013] and the spectral bandit analysis [Valko et al., 2014]. To be compatible with GP notations, we use the function form f such that $f(v_i) = f_i$. As in these results, our regret bounds depend on the mutual information between f and observed values $\mathbf{y}_S$ at a set S of nodes:

$$\mathcal{I}(\mathbf{y}_S; f) := H(\mathbf{y}_S) - H(\mathbf{y}_S \mid f),$$

where $H(\cdot)$ denotes the entropy. If f is drawn from a GP with observation noise distributed independently as $\mathcal{N}(0, \sigma)$, the mutual information has the following analytical form:

$$\mathcal{I}(\mathbf{y}_S; f) \;=\; \mathcal{I}(\mathbf{y}_S; f_S) \;=\; \frac{1}{2}\log|I + \sigma^{-2}\bar{\mathbf{C}}_{S,S}|.$$

Let

$$\gamma_T := \max_{S \in V, |S|=T} \frac{1}{2}\log|I + \sigma^{-2}\bar{\mathbf{C}}_{S,S}|,$$

i.e., the maximum information about f gained by observing T function evaluations. The regrets of our algorithms depend on the growth rate of γ_T, which can be linear in T for arbitrary graphs. However, real-world graphs often possess rich structures, such as clusters or communities, and practical measures of relevance are often highly correlated with these structures, resulting in slowly-growing γ_T. To formalize this intuition, we follow Valko et al. [2014] to consider the *effective dimension*:

$$d_T^* := \max\left\{ i \mid \lambda_i \leq \frac{\sigma^{-2}T}{(i-1)\log(1 + \frac{T}{\sigma^2\omega_0})} \right\},$$

where λ_i is the i-th smallest Eigenvalue of $\tilde{L}$ and $\lambda_1 = \omega_0$. The effective dimension is small when the first few λ_i's are small and the rest increase rapidly, as is often the case for graphs with community or cluster structures. On the contrary, if all the Eigenvalues are small then d_T^* may be linear in T. The following lemma bounds γ_T in terms of d_T^*:

Lemma 2.5. *Let T be the total number of rounds. Then*

$$\gamma_T \leq 2d_T^* \log\left(1 + \frac{T}{\sigma^2\omega_0}\right).$$

Proof. By Lemma 7.6 of Srinivas et al. [2012] and the fact that λ_i^{-1} is the i-th largest Eigenvalue of the kernel $K = \tilde{L}^{-1}$, we have

$$\gamma_T \leq \max_{\substack{\{m_i\}_{i=1}^T, m_i \geq 0, \\ \sum_{i=1}^T m_i = T}} \sum_{i=1}^{T} \log\left(1 + \frac{m_i}{\sigma^2 \lambda_i}\right).$$

Then by applying the same argument that proves Lemma 6 of Valko et al. [2014], we obtain the desired result. □

Active Search Regret

We bound the cumulative regret of an active search algorithm, which is defined by

$$R_T := \sum_{t=1}^{T} f(v_t^*) - f(v_t),$$

where $\{v_t\}_{t=1}^T$ is the sequence of unique nodes selected by the algorithm and $\{v_t^*\}_{t=1}^T$ is the set of optimal choices. For the two proposed UCB-style algorithms, GP-SOPT.TT and GP-SOPT.TOPK, we give the following bound on their cumulative regrets.

Theorem 2.6. *Pick $\delta \in (0,1)$. Assume the true function f lies in the RKHS characterized by the kernel $\bar{\mathbf{C}} = (\mathbf{D} - \mathbf{W} + \omega_0 \mathbf{I})^{-1}$ and its RKHS norm is upper-bounded by B, i.e. $\mathbf{f}^\top \bar{\mathbf{C}}^{-1} \mathbf{f} \leq B^2$. Assume the observation noise ϵ_t has zero-mean conditioned on the past and is bounded by σ almost surely. Let GP-SOPT.TT and GP-SOPT.TOPK use the GP prior with zero mean and covariance $\bar{\mathbf{C}}$, the Gaussian observation noise model $\mathcal{N}(0, \sigma^2)$, and $\alpha_t := \sqrt{2B^2 + 300\gamma_t \log^3(t/\delta)}$. Their cumulative regrets satisfy*

$$Pr(\{R_T \leq k\sqrt{c_1 T \alpha_T \gamma_T} \ \ \forall T \geq 1\}) \geq 1 - \delta,$$

where the randomness is over the observation noise and $c_1 := \frac{8/\omega_0}{\log(1+\sigma^{-2})}$. This implies

$$R_T = \tilde{O}(k\sqrt{T}(B^2\sqrt{d_T^*} + d_T^*))$$

with high probability.

This result is easily derived from the regret analysis of the GP-SELECT algorithm proposed by Vanchinathan et al. [2013] because the exploration terms used by GP-SOPT.TT and GP-SOPT.TOPK both satisfy $\sigma_t(v) \leq s_t(v) \leq k\sigma_t(v)$, thereby maintaining the UCB property. Although our regret bounds are k times worse than the GP-SELECT bound, the actual regrets tend to behave more favorably as we observe in our experiments that after a few tens of rounds, $s_t(v)$ becomes smaller than $k\sigma_t(v)$ for almost all unqueried nodes, and the two proposed algorithms usually outperform GP-SELECT.

2.7 Experiments

2.7.1 Active Learning and Surveying

The active learning heuristics to be compared are:

1. The new Σ**-optimality** with greedy one-step lookahead applications.
2. **V-optimality** with greedy one-step lookahead [Ji and Han, 2012].
3. **Expected error reduction (EER)** [Settles, 2010] with one-step lookahead. Nodes are selected which maximize the average probability margin between the most likely one-vs-all class and the second most likely one-vs-all class ($\hat{y}^{(1)} - \hat{y}^{(2)}$) in expectation.
4. **Uncertainty sampling (Unc)** with uncertainty measured by the prediction margin.
5. **Mutual information gain (MIG)** described in Krause et al. [2008]
6. **Random selection** with 12 repetitions.

We use GRF/BP model with $\delta = 0$ and $\beta = 1$ as our learning model. In such a setting, the connectivity between different nodes on a graph is strongest and the effect of the outliers is at its minimum. We feel that these parameters generally yields to better baseline results.

Comparisons are made on the following real-world network graphs or manifold graph embeddings.

1. **DBLP coauthorship network (DBLP)**.[3] This is a coauthorship graph from the DBLP database. The nodes represent scholars and the weighted edges are the number of papers bearing both scholars' names. The largest connected component has 1711 nodes and 2898 edges. The node labels were hand assigned in Ji and Han [2012] to one of the four expertise areas of the scholars: machine learning, data mining, information retrieval, and databases. Each class has around 400 nodes.
2. **Cora citation network (Cora)**.[4] This is a citation graph of 2708 publications, each of which is classified into one of seven classes by topic. The network has 5429 links. We took its largest connected component, with 2485 nodes and 5069 undirected and unweighted edges.
3. **CiteSeer citation network (CiteSeer)**.[4] This is another citation graph of 3312 publications, each of which is classified into one of six classes by topic. The network has 4732 links. We took its largest connected component, with 2109 nodes and 3665 undirected and unweighted edges.
4. **Scikit-learn handwritten digits (digits)**.[5] This is an image classification database published in the scikit-learn software. The database contains 1797 images of hand written digits (0-9) with 8×8 pixel resolution. Every digit class contains roughly 180 images. We created a 7-nearest neighbor **(7-nn)** graph using Euclidean distances of raw features and symmetrized the resulting graph.
5. **Isolated Letter Speech Recognition (ISOLETe / ISOLET4)**.[6] This is a UCI benchmark

[3] http://www.informatik.uni-trier.de/~ley/db/

[4] http://www.cs.umd.edu/projects/linqs/projects/lbc/index.html

[5] http://scikit-learn.org/stable/auto_examples/manifold/plot_lle_digits.html

[6] http://archive.ics.uci.edu/ml/datasets/ISOLET

database of human pronunciations of the 26 English letters. For every letter pronunciation, 617 domain-specific features are created. We used the first 4 mini-batches which contain 120 human subjects **(ISOLET4)**. Further, we also looked at a harder problem that distinguishes letters containing "e" sound (B, C, D, E, G, P, T, V, Z) **(ISOLETe)**. For both problems, we constructed a 4-nearest neighbor **(4-nn)** graph using Euclidean distances of raw features and symmetrized the resulting graph.

6. **Face pose recognition (pose)**.[7] This is a database that regresses semantic information from images. 687 pictures of the same sculpture face were taken with different face poses and lighting conditions. The goal is to reconstruct the face poses (2-dimensional: left-right and up-down). To solve the problem, we constructed a 7-nearest neighbor **(7-nn)** graph using Euclidean distances of the first 240 principal components and symmetrized the resulting graph.

To summarize, our pool of databases aims to cover most of Table 2.1

Table 2.1: Datasets and Experiments Overview

Model Type \ Task	Classification & Survey	Regression
Network graphs	**DBLP, Cora, CiteSeer**	N/A
Manifold graph embeddings of the Euclidean space	**digits, ISOLET4, ISOLETe**	**pose**

2.7.2 Network Graphs

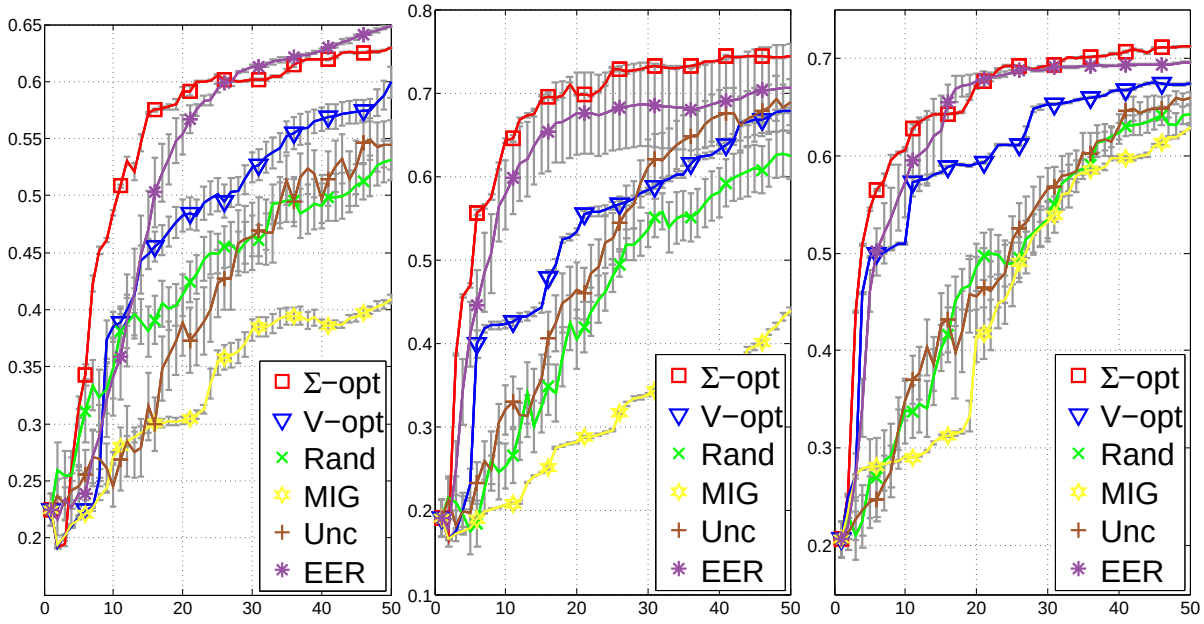
Classification. For active classification, Figure 2.10 shows the prediction accuracy of the unlabeled nodes using only the labels from the nodes that each active learning queries, except for the first common seed node which was assigned at random. Every curve shows the mean and its standard error after 12 runs.

On all three datasets, Σ-optimality outperforms other methods by a large margin especially during the first five to ten queries. The runner-up, EER, catches up to Σ-optimality in some cases, but (1) it is an order slower to evaluate, (2) it requires query results immediately before the next query, whereas both V-optimality and Σ-optimality do not, and (3) it does not have theoretical guarantees.

The win of Σ-optimality over V-optimality has been intuitively explained as Σ-optimality having better exploration ability and robustness against outliers. That all three active learning algorithms win over random selection validates the effectiveness of the GRF model which assumes node labels cluster according to graph clusters.

We also noticed that MIG and Unc methods do not perform significantly better than random. This is because both heuristics tend to query mostly outliers on the graph.

[7] `http://isomap.stanford.edu/datasets.html`



(a) DBLP coauthorship, 4 classes. (b) Cora citation, 7 classes. (c) CiteSeer citation, 6 classes.

Figure 2.10: Classification accuracy vs the number of queries. Model is GRF/BP with $\delta = 0$.

Surveying. We also performed real-world experiments on the root-mean-square-error (RMSE) of the class proportion estimations, which is the survey risk that the Σ-optimality minimizes. The Σ-optimality beats the V-optimality (Figure 2.11).

With the survey experiments, the objective is $\|\hat{\mathbb{E}}\hat{\mathbf{y}} - \pi\|_2/\sqrt{C}$ on unlabeled set $\mathbf{u}$, where $\hat{\mathbf{y}}$ is the vector of prediction means in different one-vs-alls, C is the number of classes and π is the C-dimensional true class distribution of unlabeled nodes. Every curve shows the mean and its standard error after 12 random initializations.

2.7.3 Manifold Graph Embeddings of the Euclidean Space

Detailed data preprocessing. To embed the Euclidean features from the databases **digits**, **ISOLETe**, **ISOLET4**, and **pose** in graphs, we used k-nearest neighbor graphs using the Euclidean distance. In **digits**, we created a 7-nearest neighbor graph based on the Euclidean distance of raw features, i.e. the concatenation of 64 image pixel gray values. The graph was further symmetrized by removing the direction information (and also doubling the edge weight if an edge was originally bi-directional). The resulting graph contain 1797 nodes and 8727 edges. Visual inspection shows that the resulting graph fits the labels well.

In both **ISOLETe** and **ISOLET4**, we found the 4-nearest neighbor graph based also on Euclidean distances of raw features, which is the 617 dimensional domain-specifc features. The graphs were further symmetrized in the same manner. The resulting graph for **ISOLETe** contains 2160 nodes and 6337 edges and for **ISOLET4** 6238 nodes and 18662 edges. Visual inspection shows that the resulting graphs are moderately difficult: while some classes are separated from other

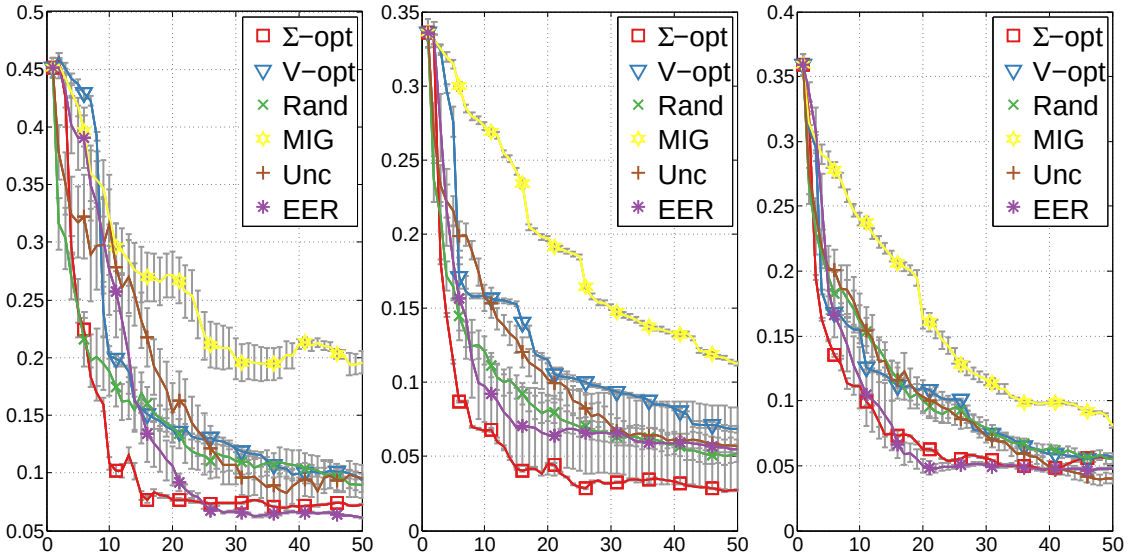


Figure 2.11: Survey RMSE, $\|\hat{\mathbb{E}}\hat{\mathbf{y}} - \pi\|_2/\sqrt{C}$, on unlabeled set $\mathbf{u}$. Model is GRF/BP with $\delta = 0$.

classes by sparse cuts, about half of the nodes are close to nodes of other classes in graph distances.

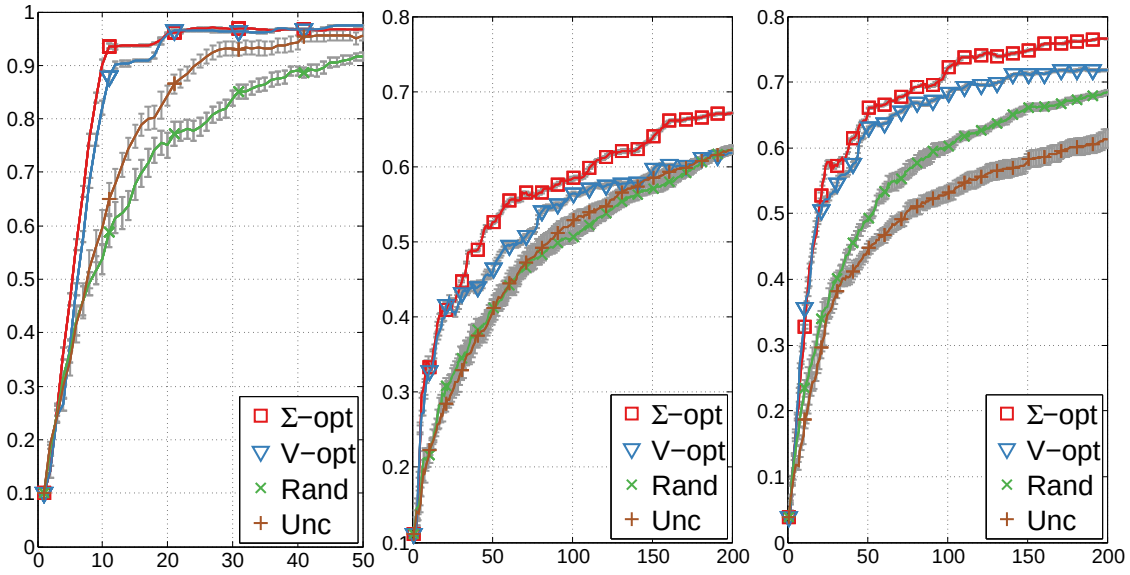
Classification results.

Figure 2.12 shows the prediction accuracy of the unlabeled nodes using only the labels from the nodes that each active learning queries, except for the first common seed node which was assigned at random. Every curve shows the mean and its standard error after 12 runs. MIG and EER were excluded in comparison because they are slow to run.

On all three manifold graph embeddings of the Euclidean space, Σ-optimality again outperforms other methods by a large margin, while all baseline methods yield to acceptable classification accuracies. We reason that this result follows the spectral and cut similarity between manifold graph embeddings and the network graphs in previous experiments. Specifically, we observed that in the 2D layouts of these manifold graphs, graph clusters have purer labels and there are also smaller and less important clusters that distract the heuristics.

Regression. Finally, we performed a graph regression experiment on the **pose** database. To create a manifold graph embedding, we used the 7-nearest neighbor graph based on the 240 principal components of face images that come with the database we downloaded. Then we symmetrized the resulting graph. There are 698 nodes and 2562 edges on this graph. The validity of this graph is checked as we recover a 2-dimensional (2D) Euclidean space layout of our graph similar to the Isomap method [Tenenbaum et al., 2000]. The relative positions of the recovered 2D coordinates agree with the relative yaws and pitches of the original face poses.

Figure 2.13 show the RMSE of the 2D pose predictors of all unlabeled nodes based on the 2D pose labels queried by various active learning heuristics. The curves are averaged after 12 runs from different randomly sampled starting nodes. The error bars show the standard error of the mean.



(a) **digits**, 7-nn, 10 classes. (b) **ISOLETe**, 4-nn, 9 classes. (c) **ISOLET4**, 4-nn, 26 classes.

Figure 2.12: Classification accuracy vs the number of queries. Model is GRF/BP with $\delta = 0$.

V-optimality outperforms Σ-optimality and both outperformed random selection. The result is similar to what we have seen in the simulation. An explanation is that for active regression problems, V-optimality directly minimizes the corresponding risk and thus is the best-performing heuristic.

2.7.4 Active Search

We conduct experiments on three graph data sets that were studied by Wang et al. [2013]. We briefly summarize them below.

5000 Populated Places. The nodes of this graph are 5000 concepts in the dbpedia[8] ontology marked as populated places. Each place is supported by a Wikipedia page, and an undirected edge is created between two places if either one of their two Wikipedia pages links to the other. There can be multiple edges between two places. The dbpedia ontology divides populated places into five categories: administrative regions, countries, cities, towns and villages. The 725 administrative regions are selected as targets while all the others are considered irrelevant.

Citation Network. This dataset consists of 14,117 papers in top Computer Science venues available on citeseer. The graph is created by adding an undirected edge between two papers if either one cites the other. The 1844 NIPS papers are chosen as targets.

Wikipedia Pages on Programming Languages. A total of 5,271 Wikipedia pages related to programming languages are the nodes of this graph, and an undirected edge exists between two pages if they are linked together. Wang et al. [2013] performed topic modeling and chose the

[8] www.dbpedia.org

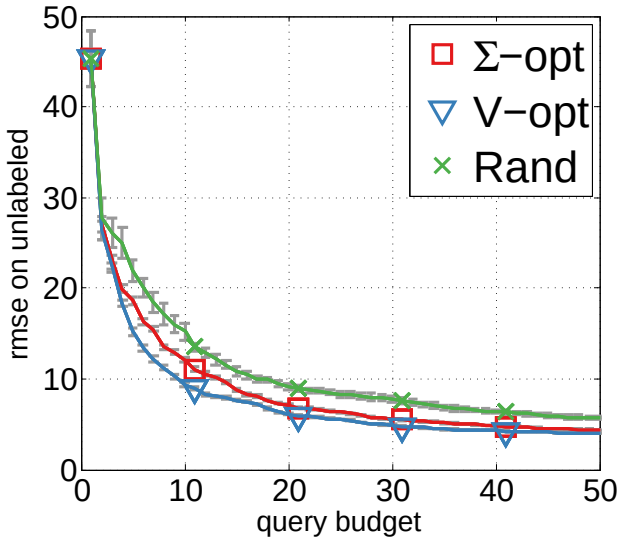
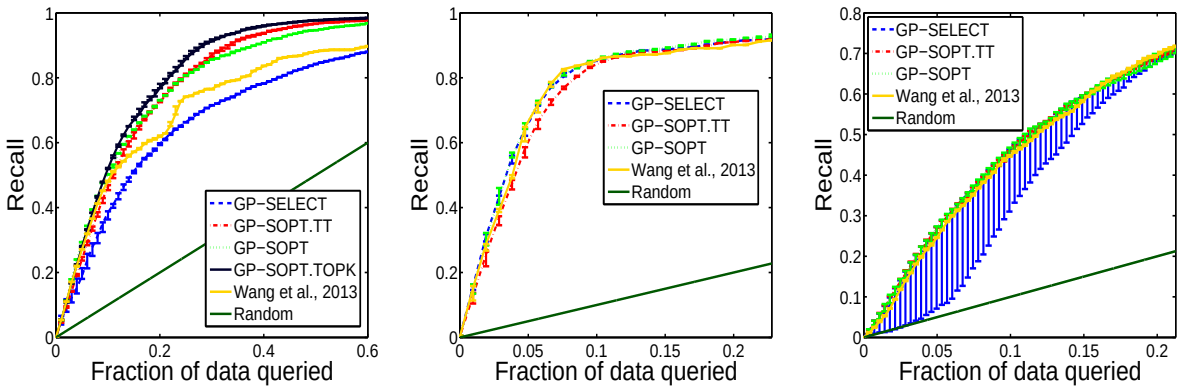


Figure 2.13: Regression RMSE vs the number of queries on the **pose** 7-nn graph. Lower is better.



(a) 5000 Populated Places (b) Wikipedia (c) Citation Network

Figure 2.14: Recall v.s. Percentage of labels queried

202 pages related to objective oriented programming as the targets, treating all the others as irrelevant.

As demonstrated by Wang et al. [2013], the three graphs and their target label distributions exhibit qualitative differences and thus serve as good benchmarks. The citation network has many small components and target nodes appear in many of them, while the Wikipedia graph has large hubs and most target nodes reside in one of them. The graph of populated places lies in between these two extremes, with components of various sizes containing target nodes.

On all of the three data sets we compare two of the proposed methods: GP-SOPT.TT and GP-SOPT against GP-SELECT (GP-UCB without replacement) and the active search algorithm (AS-on-Graph) by Wang et al. [2013]. We only evaluate GP-SOPT.TOPK on the 5000 populated places data due to its heavy computation. For each dataset we perform 5 experiments,

each with a randomly chosen target node as the seed. For the proposed methods and GP-SELECT, the main tuning parameters are the exploration-exploitation trade-off parameter α_t and the observation noise variance σ^2. For GP-SOPT.TT and GP-SOPT.TOPK there is additionally the thresholding parameter k. We consider the following values for them. Populated Places: $\alpha_t \in \{4, 2, 1, 0.1, 0.01, 0.001\}$, $\sigma^2 \in \{1, 0.5, 0.25, 0.1\}$ and $k \in \{200, 400, 800\}$. Wikipedia: $\alpha_t \in \{0.1, 0.01, 0.001\}$, $\sigma^2 \in \{1, 0.5, 0.25, 0.1\}$ and $k \in \{200, 400, 800\}$. Citation Network: $\alpha_t \in \{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}, \sigma^2 \in \{1, 0.5, 0.25, 0.1\}$ and $k \in \{400, 800, 1600\}$. Although in theory α_t should be iteration-dependent, we find that a fixed value often performs well in practice. On all data sets we set the kernel regularization parameter $\omega_0 = 0.01$. The AS-on-Graph algorithm has several parameters, and we only tune the exploration-exploitation trade-off parameter α. It is set to 0.1 on Populated Places and Citation Network, and 0.0001 on Wikipedia, which are the best performing values. Other parameters are set based on Wang et al. [2013].

Results are in Figure 2.14, where we plot the recall, i.e., the percentage of targets found by the algorithms, versus the percentage of the whole data set queried. More specifically, for each algorithm we obtain its mean recall curve over the top 15% (except for AS-on-Graph) parameter combinations in each experiment, as judged by the area under the recall curve. We then plot the median, maximum and minimum over the five experiments in Figure 2.14.

The three proposed methods clearly outperform AS-on-Graph and GP-SELECT on Populated Places, while all methods perform equally well on Wikipedia. We think this has to do with the underlying graph structure and target distribution. As mentioned before, target nodes in the Populated Places graph are spread over sub-graphs of various sizes, and therefore exploration strategies do make a difference. We observe that the proposed methods tend to select high-degree nodes in the first few iterations, thereby gaining much information, while GP-SELECT initially selects low-degree nodes. In contrast, most target nodes in the Wikipedia graph reside in one large component, and therefore less exploration is needed. In fact, the best values for α_t are very small, suggesting that an exploitation-only strategy is good enough for this data. On Citation Network, most methods perform well except that GP-SELECT performs quite poorly in one experiment. This may again indicate GP-SELECT is less robust in the presence of many low-degree nodes.

2.8 Discussions

In this chapter, we discuss active search on a graph with known structure. Each node bears a reward, which is unknown at first but can be noisily observed upon query. An active search algorithm aims to accumulate as large a sum of rewards from the queried nodes as possible under limited budgets. We assume that the node rewards vary smoothly along the graph.

Popular Bayesian UCB-style algorithms [Srinivas et al., 2012, Valko et al., 2014, Vanchinathan et al., 2013] use the marginal standard deviation as their exploration criterion, leading to the undesirable tendency of selecting peripheral nodes on a graph. Instead, we consider Σ-optimality on graphs, which can more efficiently reduce the variance of the reward function estimate by

sampling cluster centers. We show the advantage of our method in experiments with real graphs and provide a theoretical guarantee on the cumulative regret.

One interesting future direction is deriving tighter regret bounds for the proposed methods that match their empirical performances. We imagine it may be possible to bound the regret directly by the difference in Σ-optimality (Bayes survey risks, R_Σ), which may have better properties than differential information gain, γ_T on graphs. On the other hand, γ_T is based on D-optimality, which may be have severe issues with graphs (Figure 2.7).

GRFs are only one possible way to extend label propagation in SSL. They connect to unnormalized graph Laplacians. On the other hand, normalized graph Laplacians give different properties that may be empirically interesting to test. Further, an ideal model of the graph, including both the edge features and regularizations, should be learned or transferred from experiments in similar domains. Learning the graph structure is a different but rich topic [Lafferty et al., 2001, Smola and Kondor, 2003].

Additionally, we make the following observation on the spectral aspect of Σ-optimality. Analyzing the spectrum of a graph Laplacian may yield even more convincing arguments on the generalization of active learning. Besides, extracting the smallest eigenvalues and their corresponding eigenvectors is easier to scale than computing the full inverse of an augmented graph Laplacian.

2.8.1 Spectral Observations

Many exploration heuristics can be written as a function of the spectral difference between the current model and one-step look-ahead posterior model. Let $\mathbf{C}_t$ be the covariance matrix with decreasingly sorted eigen-values $\boldsymbol{\lambda}_t^2 = (\lambda_{t,(1)}^2, \ldots, \lambda_{t,(n)}^2)^\top$, and $\mathbf{C}_{t+1}$ and λ_{t+1}^2 to be their posterior counterparts after observing a node, v. A score based on spectral difference is then,

$$s_t(v) = h^{-1}\left(\sum_{k=1}^{n} h\Big(\lambda_{t,(j)}\Big) - \sum_{k=1}^{n} h\Big(\lambda_{t+1,(j)}\Big)\right)$$
$$\text{s.t. } h'(s) > 0, \forall s > 0,$$

where the difference inside $h^{-1}(\cdot)$ is nonnegative, because we can prove using induction and definition of eigen-vectors, for example with $j = 1$ and $\mathbf{q}_{t+1,(1)}$ being the eigen-vector corresponding to $\lambda_{t+1,(1)}$ in the posterior model, $\lambda_{t,(j)}^2 - \lambda_{t+1,(1)}^2 \geq {\langle \mathbf{q}_{t+1,(1)}, \mathbf{c}_{t+1}(v)\rangle^2}/{(\sigma_n^2 - \sigma_{t+1}^2(v))} \geq 0$.

Case 1. $h(s) = -log(s), s_t(v) = \sqrt{1 + {\sigma_t^2(v)}/{\sigma_n^2}}$. This heuristic adds biases to maximize the differential information gain of the joint distribution of node values, turns out to pay too much attention to the graph periphery, which actually prevents information gathering in the true problem against intuition. Precisely, differential entropy is sensitive to tails of the distribution, which happens to be the place of the biggest model mismatch of our GRF models.

Case 2. $h(s) = s^2, s_t(v) = \sqrt{\mathrm{tr}(\mathbf{C}_t) - \mathrm{tr}(\mathbf{C}_{t+1})} = \sqrt{{\|\mathbf{c}_t(v)\|_2^2}/{(\sigma_t^2(v) + \sigma_n^2)}}$. This criterion resembles **V-optimality**, which though alleviates the situation by adding independence assumptions on the

nodes and measuring the sum of the marginal variances, cannot completely address the selection bias at graph peripheries, because the self-variance term usually dominates the sum of squares of $\|\mathbf{c}_t(v)\|_2^2$.

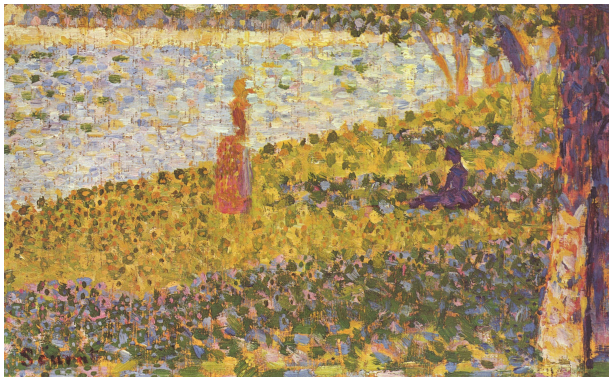
Case 3. $h(s) = s^p, p \to \infty, \lambda_{\max}(\mathbf{C}_t) - \lambda_{\max}(\mathbf{C}_{t+1})$. This heuristic aims to globally control the posterior marginal variances of every node, by upper-bounding them by $\lambda_{\max}^2$. Indeed, for any node k and any covariance matrix $\mathbf{C}$, $C_{kk} = \mathbf{e}_k^\top \mathbf{C} \mathbf{e}_k \leq \max {\mathbf{v}^\top \mathbf{C} \mathbf{v}}/{\mathbf{v}^\top \mathbf{v}} = \lambda_{\max}^2(\mathbf{C})$.

Our intuition is that **Sigma-optimality** connects to this criterion via approximations. First, assuming that the principal eigen-vector of $\mathbf{C}_t$ is $\mathbf{q}_t$, then $\lambda_{\max}^2(\mathbf{C}_{t+1} \mid v) \approx \lambda_{\max}^2(\mathbf{C}_t) - \frac{\langle \mathbf{q}_t, \mathbf{c}_t(v) \rangle^2}{(\sigma_t^2(v) + \sigma_n^2)}$ and, compounding the square-root operator, $s_t(v) \approx \frac{1}{2\lambda_{\max}(\mathbf{C}_t)} \frac{\mathbf{c}_t(v)^\top \mathbf{q}_t}{\sqrt{\sigma_t(v)^2 + \sigma_n^2}}$.

Realize that $\mathbf{C}_0^{-1} = \mathbf{D} - \mathbf{A} + \omega_0 \mathbf{I}$ has its smallest eigen-vector (with respect to ω_0) very close to $\frac{1}{n}\mathbf{1}$, that same vector carries to be $\mathbf{q}_0$ for the largest eigen-value of $\mathbf{C}_0$. At this point, $s_t(v)$ is our Sigma-optimality up to a selection-independent constant.

In fact, this approximation can be valid for larger t's. Further break the graph down to different (relatively isolated) connected components, where each individual component is relatively unexplored, and therefore contains a principal eigen-vector, relative to the component, which will approximate $\mathbf{q}_{t,(c)} \approx \mathbf{1}_{\mathcal{C}}$, where c is the rank of this eigen-vector and $\mathcal{C}$ the subset of nodes of this connected component. The more under-explored the component is, the more likely that $\mathbf{q}_{t,(c)}$ becomes the principal eigen-vector, $\mathbf{q}_t$ and also $\mathbf{q}_{t,(c)}$ gets close to $\mathbf{1}_{\mathcal{C}}$.

In the meantime, every column on the current covariance matrix $\mathbf{c}_t(v)$ will also reflect independence between these (relatively isolated) components. Thus, the inner product can be roughly approximated as, $\mathbf{q}_t^\top \mathbf{c}_t(v) \approx \mathbf{1}_{\mathcal{C}}^\top \mathbf{c}_t(v) + \mathbf{1}_{\bar{\mathcal{C}}}^\top \mathbf{0} = \mathbf{1}^\top \mathbf{c}_t(v)$, where $\bar{\mathcal{C}}$ is the complement of $\mathcal{C}$. Again, Sigma-optimality approximates the difference of the spectral norm between prior and one-step look-ahead covariance matrices.



George Seurat, *Femmes au bord de l'eau*, 1885-86.

3

Active Area Search and Pointillism

3.1 Introduction

Consider a function containing interesting patterns that are defined only over a region of space. For example, if you view the direction of wind as a function of geographical location, it defines fronts, vortices, and other weather patterns, but those patterns are defined only in the aggregate. If we can only measure the direction and strength of the wind at point locations, we then need to infer the presence of patterns over broader spatial regions.

Many other real applications also share this feature. For example, an autonomous environmental monitoring vehicle with limited onboard sensors needs to strategically plan routes around an area to detect harmful plume patterns on a global scale [Valada et al., 2012]. In astronomy, projects like the Sloan Digital Sky Survey [Eisenstein et al., 2011] search the sky for large-scale objects such as galaxy clusters. Biologists investigating rare species of animals must find the ranges where they are located and their migration patterns [Brown et al., 2014]. We aim to use active learning to search for such global patterns using as few local measurements as possible.

Traditionally, active learning assumes that a label is associated with each observable data point, which may be revealed upon querying. Traditional active search then aims to maximize the number of positively-labeled points that can be collected, given a finite query budget. Here, however, the labels are instead defined by the presence of specific patterns over broader spatial regions. While we allow (noisy) observations of the values of the smooth underlying function at any feasible point locations, the function in fact turns into an *auxiliary* function because it does not directly define rewards. Instead, our goal is to identify the most number of positive regions where positive patterns can be inferred, given any finite budget of point observations.

Since we aim to search for positive patterns over broader spatial regions, the point query strategy will be very different from plain active search for positive points. This bears some resemblance to the artistic technique known as *pointillism*, where the painter creates small and distinct dots each of a single color, but when viewed as a whole they reveal a scene. Pointillist paintings typically use a denser covering of the canvas, but in our setting, "observing a dot" is expensive. Therefore, we make fewer observations in order to uncover interesting regions as quickly as possible.

To simplify discussions, we assume the pool of regions that are feasible to contain positive patterns are predefined. In the common scenario, it includes a set of sliding windows of equal sizes that cover the entire navigable space with reasonable overlaps. Some applications use other natural definition of regions. The patterns, on the other hand, can be either simple or complex, depending on the application:

Active Area Search (AAS). We search for simple patterns that are defined on the average value of the smooth auxiliary function in a region. Positive labels are assigned to regions where the average value is larger than a predefined threshold, with high probability.[1] AAS is useful in the example of environmental monitoring with mobile sensors. The variability of the sensors and environmental conditions on a river mean that no single sensor reading will ever be sufficient to identify a significant pollution issue. Instead, real pollution issues are identified by a set of regions within a certain region that have a large average value. Although a boat gives us the capability to take a measurement anywhere, it does not provide the sensing bandwidth to monitor every location all the time. Besides, sensing cost dominates travel cost in many cases.[2] Therefore we need an algorithm to sequentially choose sensing locations with a goal of identifying polluted regions.

Active Pointillist Pattern Search (APPS). We search for complex patterns that are defined by a classifier that takes functional inputs. Since the classifier operates under uncertainty when we have incomplete observation of the function in the region, positive labels are assigned when the classifier has a sufficiently high probability output. In applications, APPS allows us to find vortices by selecting point locations to observe the corresponding wind flow vectors. APPS can be viewed as a generalization of AAS, by allowing arbitrary classifiers rather simple thresholds of the function average.

Functional Probit Models (FPMs). AAS is a special case of APPS, where the classifier is formed by a probit link function of a linear functional of the underlying function that produces observations. We call the family of models Functional Probit Models (FPMs), which is a slight generalization of AAS.

Mathematically, we assume that the low-level responses of point queries comes from a random function with a Gaussian process (GP) prior [Rasmussen and Williams, 2006], whose hyperparameters are externally designed. This assumption allows to infer region patterns with incomplete

[1] Theoretically, the true average value is never obtainable because it requires complete observation of every point value in the region using infinite sensing budget.

[2] A typical dissolved oxygen sensor requires about one minute for the reading to settle down after moving [Valada et al., 2012], which is enough time for the small boat to travel end-to-end in the areas we've considered so far. Similarly, any application requiring *in situ* lab analysis of samples would have this property.

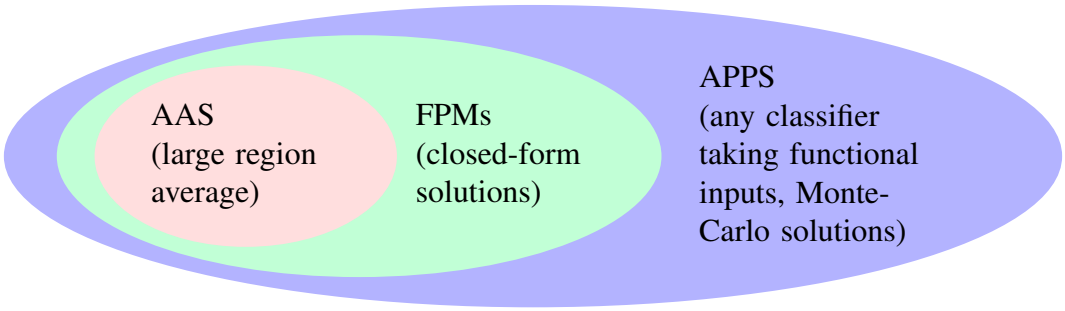


Figure 3.1: Region patterns with increasing complexity.

observation. We accomplish active area search by sequentially selecting point locations to observe so as to approximately maximize expected reward for identifying positive patterns. We also have closed-form solutions and insights when the patterns are simple, such as AAS or FPMs.

3.1.1 Related Work

Our concept of active pattern search falls under the broad category of *active learning* [Settles, 2010], where we seek to sequentially build a training set to achieve some goal as fast as possible. Our focus solely on finding positive ("interesting") regions, rather than attempting to learn to discriminate accurately between positives and negatives, is similar to the problem previously described as *active search* [Garnett et al., 2012]. In previous work on active search, however, it has been assumed that the labels of interest can be revealed directly. In active pattern search, on the other hand, the labels are never revealed but must be inferred via a provided classifier. This indirection increases the difficulty of the search task considerably.

In *Bayesian optimization* [Brochu et al., 2010, Osborne et al., 2009], we seek to find the global optimum of an expensive black-box function. Bayesian optimization provides a model-based approach where a Gaussian process (GP) prior is placed on the objective function, from which a simpler acquisition function is derived and optimized to drive the selection procedure. In [Tesch et al., 2013], the authors extend this idea to optimizing a latent function from binary observations. Our proposed active pattern search also uses a Gaussian process prior to model the unknown underlying function and derives an acquisition function from it, but differs in that we seek to identify entire *regions* of interest, rather than finding a single optimal value.

Another intimately related problem setup is that of *multi-arm bandits* [Auer et al., 2002], with more focus on analysis of the cumulative reward over all function evaluations. Originally, the goal was to maximize the expectation of a random function on a discrete set; a variant considers the optimization in continuous domains [Kroemer et al., 2010, Niranjan et al., 2010]. However, like Bayesian optimization, multi-arm bandit problems usually do not consider discriminating a regional pattern.

Level set estimation [Gotovos et al., 2013, Low et al., 2012], rather than finding optima of a function, seeks to select observations so as to best discriminate the portions of a function above

and below a given threshold. This goal, though related to ours, aims to directly map a portion of the function on the input space rather than seeking out instances of patterns. LSE algorithms can be used to attempt to find some simple types of patterns (say, areas with high mean), but even then its learning goal underperforms in the mismatched search objective, and it does not attempt more complex models.

3.2 Problem Formulation

There are three key components of the APPS framework: a function f which maps input covariates to data observations, a predetermined set of regions wherein instances of function patterns are expected, and a classifier that evaluates the salience of the pattern of function values in each region. We define $f\colon \mathbb{R}^m \to \mathbb{R}$ to be the function of interest,[3] which can be observed at any location $x \in \mathbb{R}^m$ to reveal a noisy observation y. We assume the observation model

$$y = f(x) + \varepsilon, \quad \text{where } \varepsilon \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_n^2).$$

We suppose that a set of regions where matching patterns might be found is predefined, and will denote these $\bar{\mathcal{A}} = \{A_j \subset \mathbb{R}^m \colon j = 1, \ldots, k\}$. Finally, for each region A, we assume a classifier h_A which evaluates f on A and returns the probability that it matches the target pattern, which we call *salience*:

$$h_A(f) = h(f; \theta_A) \in [0, 1],$$

where θ_A is the set of parameters including both the location of A and other necessary variables that define the region pattern classifier. The mathematical interpretation of h_A is similar to a functional of f. Classifier forms are typically the same for all regions with different parameters.

In the example of AAS, positive labels are assigned to regions where the average value is above a predefined threshold τ. In this case, $\theta_A = (A, \tau)$ and the region labels are defined by,

$$h_A(f) = \mathbb{1}_{\left\{\frac{1}{|A|}\int_{x\in A} f(x)\,\mathrm{d}x > \tau\right\}}. \tag{3.1}$$

Figure 3.2 demonstrates AAS in a $1d$ environment where the regions are line segments and the labels are defined by the average values.

A slight generalization of AAS is a FPM. Here, the classifier is formed by a probit link function of a weighted integral of the underlying function that produces observations. A probit link function uses the cumulative distribution function of the standard normal, $\Phi(a) = \int_{-\infty}^{a} \frac{1}{\sqrt{2\pi}} \exp\{-\frac{u^2}{2}\}\,\mathrm{d}u$. Let the set of classifier parameters be $\theta_A = (w_A(\cdot), \tau, c)$, where $w_A(\cdot)$ is a weight function that is nonzero only when $x \in A$, τ is a scalar, $c > 0$ is a scale variable; the functional probit model is defined as

$$h_A(f) = \Phi\left(\frac{1}{c}\left[\int w_A(x) f(x)\,\mathrm{d}x - \tau\right]\right), \tag{3.2}$$

which is equivalent to AAS classifier if we take $c \to 0$.

[3]For clarity, in this and the next sections we will focus on scalar-valued functions f. The extension to vector-valued functions is straightforward; we consider such a case in Section 3.6.3.

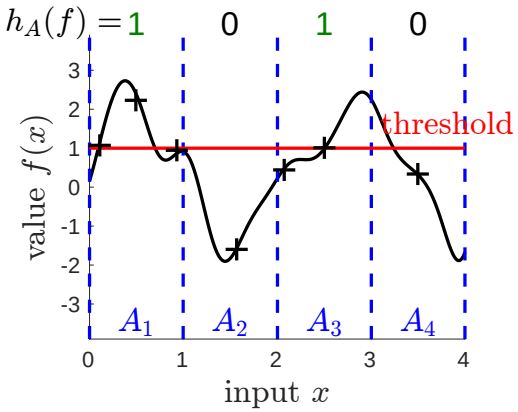


Figure 3.2: Problem definition given full knowledge of the underlying function $f(x)$. For AAS, positive labels are given to regions where the average value is above a predefined threshold.

3.2.1 Region Rewards with Incomplete Function Observations

Unfortunately, in general, we will have little knowledge about f other than the limited observations made at our selected set of points. Classifiers which take functional inputs (such as our assumed h_A) generally do not account for uncertainty in their inputs, which should be inversely related to the number of observed data points. We thus must consider the probability that $h_A(f)$ is high enough, marginalized across the range of functions f that might match our observations. As is common in nonparametric Bayesian modeling, we model f with a Gaussian process (GP) prior; we assume hyperparameters, including prior mean and covariance functions, are set by domain experts. Given a dataset $D = \{(x_i, y_i) : i = 1, \ldots, t\}$, we define

$$f \sim \mathcal{GP}(\mu, \kappa); \qquad f \mid D \sim \mathcal{GP}(\mu_{f|D}, \kappa_{f|D}), \tag{3.3}$$

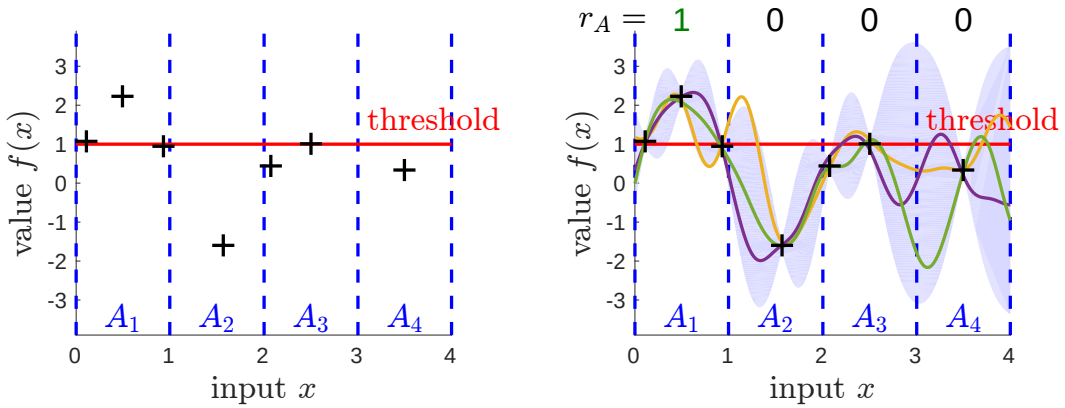
to be a given GP prior and its posterior conditioned on D, respectively. (Formal discussions are in Section 3.2.2.) Since f is a random variable, we can obtain the marginal probability that A is salient,

$$P(A \mid D) = \mathbb{E}_f\big[h_A(f) \mid D\big]. \tag{3.4}$$

We then define a matching region as one whose marginal probability passes a given threshold $1 - \alpha$. Unit reward is assigned to each matching region A:

$$r(A \mid D) = \mathbb{1}_{\{P(A|D) > 1-\alpha\}}. \tag{3.5}$$

Similar to active search [Garnett et al., 2012], active area search aims to maximize the cumulative reward at the end of a fixed number of queries. Additionally, we assume that unit reward can be collected at the same region only once. As soon as a region is flagged as potentially matching (i.e., its marginal probability exceeds $1 - \alpha$), it will be immediately flagged for further review and no longer considered during the run. Additionally, we assume that the data resulting from this investigation will not be made immediately available during the course of the algorithm;



(a) Incomplete observation of the function (b) Reward by inference with GP sampling

Figure 3.3: Given incomplete observations, true region pattern is never known to us. However, we may draw smooth functions from GP — shown as the three solid lines inside the shaded envelope in (b), which allow us to assign rewards $r_A = 1$ if the probability is sufficiently high.

rather the classifiers h_A will be trained offline. For example, if the algorithm is being used to run autonomous sensors and scientists collect separate data to follow up on a matching region, these assumptions allow the autonomous sensors to continue in parallel with the human intervention, and avoid the substantial complexity of incorporating a completely different modality of data into the modeling process. Making different assumptions would lead to interesting extensions to our algorithms that we do not consider here. As a result, the immediate reward of every point measurement is

$$r_t(D_t) = \sum_{A \in \mathcal{A}_t} r(A \mid D_t), \quad \text{where} \quad \mathcal{A}_t = \{A : r(A \mid D_\tau) = 0, \forall \tau < t\}, \tag{3.6}$$

and we aim to maximize the cumulative reward

$$R(D_T) = \sum_{t=1}^{T} r_t(D_t) = \sum_{j=1}^{k} \mathbb{1}_{\{\exists \tau \le t \text{ s.t. } P(A_j|D_\tau) > 1-\alpha\}}.$$

Remark 3.1. *Active search aims to find all positive subjects instead of the global optimum. If we allow repeated rewards, as soon as one positive region is found, a greedy solution could simply refuse to collect more data in the positive regions so as to abuse the current rewards, because our reward is binary. Although the greedy solution may also choose to collect in other regions in order to maximize the expected sum of rewards, the pathology in the established positive regions will unavoidably influence the designs in their neighboring regions in a negative way. We will show more in our analysis in Section 3.4.*

Another issue of reward abuse may happen when we make repeated tests about the label of a region in different query time steps. This may lead to inferior precision for the discovery of true positive regions or an increased false discovery rate. A classical fix is to notice that the distribution of maximum value in a set of variables and to apply $O(log(t))$ multiplicative corrections to

the standard deviation at step t as a safety margin. Alternatively, one may choose to use smaller and different α for each time step. We unfortunately did not consider such rigorousness, but only showed that the precision in our experiments remain empirically high.

3.2.2 Closed-Form GP Models and Rewards in AAS or FPMs

It is useful to express the GP posterior (3.3) for completeness. Further, when the classifier is as simple as AAS (3.1) or FPMs (3.2), we may express the actual reward (3.5) in closed-form in terms of the collected data. The way to achieve closed-form solutions is to realize that GP is closed under linear transformation of variables, including AAS and FPMs.

First, a Gaussian process (GP) is a statistical process to draw smooth random functions, where the outputs corresponding to every set of inputs (including sets with only one element) have a joint Gaussian distribution with parameters given by the input. A GP $f(x)$ is characterized by two (prior) function parameters, a mean function $\mu(x)$ and a kernel function $\kappa(x, x')$. The kernel function is also known as covariance function, because it defines the second moment of a GP. On the other hand, a GP is fully defined by its first two moments through the prior mean and kernel functions. Let $x_1, \dots, x_n \in \mathbb{R}^m, \forall n \geq 1$ be any combination of any number of input points. Define $\mathbf{X} = (x_1, \dots, x_n)^\top$ and we further overload

$$\mu(\mathbf{X}) = \begin{pmatrix} \mu(x_1) \\ \dots \\ \mu(x_n) \end{pmatrix}, \quad \text{and} \quad \kappa(\mathbf{X}, \mathbf{X}) = \begin{pmatrix} \kappa(x_1, x_1) & \dots & \kappa(x_1, x_n) \\ \dots & \dots & \dots \\ \kappa(x_n, x_1) & \dots & \kappa(x_n, x_n) \end{pmatrix},$$

the corresponding outputs from a GP always have joint distribution,

$$\big(f(x_1), \dots, f(x_n)\big)^\top \sim \mathcal{N}\big(\mu(\mathbf{X}), \kappa(\mathbf{X}, \mathbf{X})\big), \\ \text{where } \mathbf{X} = (x_1, \dots, x_n)^\top, \forall x_1, \dots, x_n \in \mathbb{R}^m, \forall n \geq 1.$$

An example of GP would have zero-mean and square-exponential kernel:

$$\mu(x) = 0, \quad \kappa(x, x') = \sigma_f^2 \exp\left\{-\frac{\|x - x'\|^2}{2\ell^2}\right\},$$

where $\sigma_f, \ell > 0$ are called hyper-parameters. Other forms of kernel functions are allowed, as long as the resulting covariance matrix is always symmetric and positive-definite for any combination of input points $x_1, \dots, x_n$.

Next, we aim to derive the closed-form solutions for the reward with incomplete observations. Notice that (3.1)&(3.2) define the reward by (weighted) integral of the function f and that GP is closed under any linear functionals, we may extend the input space to allow such linear functionals:

$$\langle f, \delta_x \rangle = f(x), \quad \langle f, \tfrac{1}{|A|} \mathbb{1}_{\{A\}} \rangle = \frac{1}{|A|} \int_{x \in A} f(x) \, \mathrm{d}x, \quad \langle f, w_A \rangle = \int w_A(x) f(x) \, \mathrm{d}x,$$

where δ_x is a Dirac delta function that represents the original point evaluation in the functional space. We will only use the more general form of the linear functionals and overload $f(A) = \langle f, w_A \rangle$.

Let $\psi \in \Psi$ be a unified representation in the extended space. Now, the GP prior mean and kernel functions extend to

$$\bar{\mu}(\psi) = \begin{cases} \bar{\mu}(x) & \text{if } \psi = \delta_x, \\ \bar{\mu}(w_A) = \int \bar{\mu}(x) w_A(x) \, \mathrm{d}x & \text{if } \psi = A, \end{cases}$$

$$\bar{\kappa}(\psi, \psi') = \begin{cases} \bar{\kappa}(x, x') & \text{if } \psi = \delta_x, \psi' = \delta_{x'}, \\ \bar{\kappa}(A, x') = \int \bar{\kappa}(x, x') w_A(x) \, \mathrm{d}x & \text{if } \psi = A, \psi' = \delta_{x'}, \\ \bar{\kappa}(A, A') = \iint \bar{\kappa}(x, x') w_A(x) w_{A'}(x') \, \mathrm{d}x \, \mathrm{d}x' & \text{if } \psi = A, \psi' = A'. \end{cases}$$

After collecting a set of measurements at $\mathbf{X} = (x_1, \ldots, x_n)^\top$ and observing their outcomes as $\mathbf{y} = (y_1, \ldots, y_n)$, the posterior distribution is a conjugate GP with the following new mean and kernel functions:

$$\begin{aligned} \mu(\psi \mid D) &= \bar{\mu}(\psi) + \bar{\kappa}(\psi, \mathbf{X}) \bar{\mathbf{V}}(\mathbf{X}, \mathbf{X})^{-1} \mathbf{y}, \\ \kappa(\psi, \psi' \mid D) &= \bar{\kappa}(\psi, \psi') - \bar{\kappa}(\psi, \mathbf{X}) \bar{\mathbf{V}}(\mathbf{X}, \mathbf{X})^{-1} \bar{\kappa}(\mathbf{X}, \psi'), \end{aligned} \tag{3.7}$$

where $\bar{\mathbf{V}}(\mathbf{X}, \mathbf{X}) = \bar{\kappa}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}$ is the prior covariance matrix for the noisy observations. Define marginal variance $\sigma^2(\psi \mid D) = \kappa(\psi, \psi \mid D)$. When $\psi = \psi' = A$, the posterior distribution can be efficiently computed by reusing (partial) integrals of the kernel function at the corresponding region:

$$\begin{aligned} \mu(A \mid D) &= \int \bar{\mu}(x) w_A(x) \, \mathrm{d}x + \left[\int \bar{\kappa}(x, \mathbf{X}) w_A(x) \, \mathrm{d}x \right] \bar{\mathbf{V}}(\mathbf{X}, \mathbf{X})^{-1} \mathbf{y}, \\ \sigma^2(A \mid D) &= \iint \bar{\kappa}(x, x') w_A(x) w_A(x') \, \mathrm{d}x \, \mathrm{d}x' \\ &\quad - \left[\int \bar{\kappa}(x, \mathbf{X}) w_A(x) \, \mathrm{d}x \right] \bar{\mathbf{V}}(\mathbf{X}, \mathbf{X})^{-1} \left[\int \bar{\kappa}(\mathbf{X}, x') w_A(x') \, \mathrm{d}x' \right]. \end{aligned}$$

Finally, for AAS, the probability of positive outcome is the cumulative density function:

$$P(A \mid D) = \Pr\left(\frac{1}{|A|} \int_{x \in A} f(x) \, \mathrm{d}x > \tau \;\middle|\; D \right) = \Phi\left(\frac{\mu(A \mid D) - \tau}{\sigma(A \mid D)} \right).$$

For FPMs, the probability of positive outcome also has closed-form solutions because of the conjugacy between probit models and Gaussian distributions. Let $u = \frac{1}{c}[\int w_A(x) f(x) \, \mathrm{d}x - \tau]$,

the solution is

$$
\begin{aligned}
P(A \mid D) &= \mathbb{E}\left[\Phi\left(\frac{1}{c}\left[\int w_A(x) f(x)\,\mathrm{d}x - \tau\right]\right) \,\middle|\, D\right] \\
&= \int \Phi(u)\mathcal{N}\left(u \,\middle|\, \frac{\mu(A \mid D) - \tau}{c}, \frac{\sigma^2(A \mid D)}{c^2}\right)\mathrm{d}u \\
&= \Phi\left(\frac{\mu(A \mid D) - \tau}{\sqrt{\sigma^2(A \mid D) + c^2}}\right). \qquad (3.8)
\end{aligned}
$$

When the linear functional is $w_A(x) = \frac{1}{|A|}\mathbb{1}_{\{A\}}(x)$, the FPM reward is effectively the reward of a noisy observation of the inferred function average, with noise variance c^2. As $c \to 0$, FPMs become equivalent to AAS. In the later discussions, we will use the more general form of linear models and define

$$
\begin{cases}
\nu^2(x \mid D) &= \sigma^2(x \mid D) + \sigma_n^2 \\
\nu^2(A \mid D) &= \sigma^2(A \mid D) + c^2
\end{cases}
\qquad \Rightarrow \qquad
P(A \mid D) = \Phi\left(\frac{\mu(A \mid D) - \tau}{\nu(A \mid D)}\right).
$$

The actual reward is binary depending on the probability output of the inference. Recall (3.5)&(3.6):

$$
r(A \mid D) = \mathbb{1}_{\{P(A|D) > 1-\alpha\}}, \quad \text{and} \quad r_t(D_t) = \sum_{A \in \mathcal{A}_t} r(A \mid D_t).
$$

3.3 Method: Greedy Maximization of Expected Rewards

An ideal Bayesian solution would attempt to maximize the expected reward at the end of a fixed number of queries, similar to [Garnett et al., 2012] . Directly optimizing that goal involves an exponential lookahaed process. However, this can be approximated by a greedy search like the one we perform. Closed-form solutions may also be derived for AAS and FPM models.

We now write the greedy criterion our algorithm seeks to optimize. In a sequential querying manner where the first t query steps collect a dataset $D_t = \{(x_\tau, y_\tau) : \tau = 1, \ldots, t\}$, define the remaining search subjects as $\mathcal{A}_t = \{A : P(A \mid D_\tau) \leq 1 - \alpha, \forall \tau < t\}$. We aim to greedily maximize the sum of rewards over all the regions in $\mathcal{A}_t$ in expectation,

$$
x_{t+1} = \arg\max_{x_*} \mathbb{E}^{\tilde{y}_*} \sum_{A \in \mathcal{A}_t} [r(A \mid D_t \cup \{(x_*, \tilde{y}_*)\}) \mid x_*, D_t], \qquad (3.9)
$$

where $D_t \cup \{(x_*, \tilde{y}_*)\}$ is the (random) dataset augmented with x_* and its lookahead observation $\tilde{y}_*$, which is simulated under the GP posterior.

A more careful examination of the GP model can yield a straight-forward sampling method. This method, in the following, turns out to be quite useful in APPS problems with rather complex classifiers. Section 3.3.1 introduces closed-form solution for simple classifiers.

At each step, given the collected observations D_t and any potential input location x_*, we can assume the distribution of possible observations $\tilde{y}_*$ as

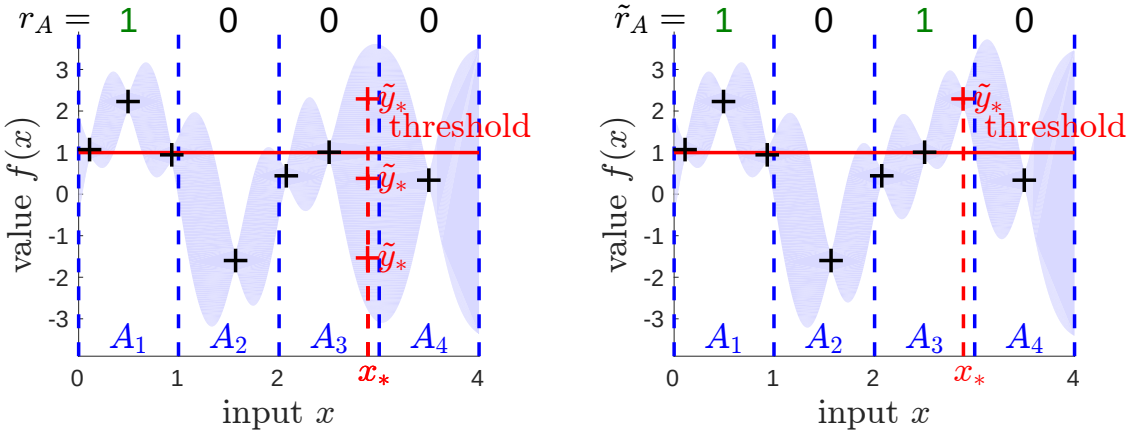
$$\tilde{y}_* \mid x_*, D_t \;\sim\; \mathcal{N}\big(\mu_{f|D_t}(x_*),\; \kappa_{f|D_t}(x_*, x_*) + \sigma^2\big). \tag{3.10}$$

Conditioned on an observation value $\tilde{y}_*$, we can update our GP model to include the new observation $(x_*, \tilde{y}_*)$, which further affects the marginal distribution of region classifier outputs and thus the probability this region is matching. With $\tilde{D}_* = D_t \cup \{(x_*, \tilde{y}_*)\}$ as the updated dataset, we define $r(A \mid \tilde{D}_*)$ to be the updated reward of region A. The utility of this proposed location x_* for region A is thus measured by the *expected* reward function, marginalizing out the unknown observation value $\tilde{y}_*$:

$$\begin{aligned} u_A(x_* \mid D_t) &= \mathbb{E}^{\tilde{y}_*}\left[r(A \mid \tilde{D}_*) \mid x_*, D_t\right] \\ &= \mathbb{E}^{\tilde{y}_* | x_*, D_t}\, \mathbb{1}_{\{P(A|D_t \cup \{(x_*, \tilde{y}_*)\}) > 1-\alpha\}} \end{aligned} \tag{3.11}$$

Finally, in active pointillist pattern search, we select the next observation location x_* by considering its expected reward over the remaining regions:

$$x_{t+1} = \arg\max_{x_*} u(x_* \mid D_t) = \arg\max_{x} \sum_{A \in \mathcal{A}_t} u_A(x_* \mid D_t). \tag{3.12}$$



(a) Sample the possible observations $\tilde{y}_*$.

(b) Rewards on the lookahead dataset.

Figure 3.4: Sampling-based solution to greedily maximize expected reward. For any point x_*: Step 1. sample possible observations $\tilde{y}_*$. Step 2. for each sampled observation, estimate the reward assuming that the lookahead dataset $D_t \cup \{(x_*, \tilde{y}_*)\}$ is the true collected dataset.

For the most general definition of the region classifier h_A, the basic algorithm is to compute (3.11) and thus (3.12) via sampling at two stages:

1. Sample the outer variable $\tilde{y}_*$ in (3.11) according to (3.10).
2. For every draw of $\tilde{y}_*$, sample enough of $(f \mid \tilde{D}_*)$ to compute the marginal reward $P(A \mid \tilde{D}_*)$ in (3.4), in order to obtain one draw for the expectation in (3.11).

To speed up the process, we can evaluate (3.12) for a subset of possible $x_* \in \tilde{\mathcal{X}}$ values as long as a good action is likely to be contained in the set.

3.3.1 Closed-Form Solutions to Utility Functions with AAS and FPMs

To derive the closed-form solution with AAS and FPMs, we start with the closed-form solution to the reward function (3.8) on the lookahead dataset, $\tilde{D}_* = D_t \cup \{(x_*, \tilde{y}_*)\}$, where $\tilde{y}_*$ is randomly sampled from GP posterior, as

$$r(A \mid \tilde{D}_*) = \mathbb{1}_{\left\{P(A|\tilde{D}_*) > 1-\alpha\right\}}, \quad \text{where} \quad P(A \mid \tilde{D}_*) = \Phi\left(\frac{\mu(A \mid \tilde{D}_*) - \tau}{\nu(A \mid \tilde{D}_*)}\right).$$

Fix A and D_t and let

$$\begin{cases} \mu_A = \mu(A \mid D_t), & \sigma_A = \sigma(A \mid D_t), & \nu_A = \nu(A \mid D_t) = \sqrt{\sigma_A^2 + c^2}, \\ \tilde{\mu}_A = \mu(A \mid \tilde{D}_*), & \tilde{\sigma}_A = \sigma(A \mid \tilde{D}_*), & \tilde{\nu}_A = \nu(A \mid \tilde{D}_*) = \sqrt{\tilde{\sigma}_A^2 + c^2}, \end{cases}$$

the expected utility (3.11) of a new observation x_* on region A is

$$u_A(x_* \mid D_t) = \mathbb{E}^{\tilde{y}_*} r(A \mid \tilde{D}_*) = \Pr\left[\Phi\left(\frac{\tilde{\mu}_A - \tau}{\tilde{\nu}_A}\right) > 1 - \alpha\right], \tag{3.13}$$

where we may realize from (3.7) that $\tilde{\mu}_A$ is a random variable that depends on the realization of both x_* and $\tilde{y}_*$, whereas $\tilde{\sigma}_A$ is fixed and only depends on the choice of x_*. In fact, fixing x_*, Eq 3.7 shows that $\tilde{\mu}_A$ has a linear relation with $(\tilde{y}_* \mid x_*)$, which leads to a marginal Gaussian distribution if we integrate out $\tilde{y}_*$. The marginal distribution have the form

$$\tilde{\mu}_A \sim \mathcal{N}(\mu_A, \tilde{s}^2),$$

where the marginal mean equals to the current-step mean and the variance is denoted by $\tilde{s}^2 = \tilde{s}^2(x_*, A \mid D_t)$, which depends on x_* and A. Before we discuss the closed-form solution for $\tilde{s}^2$, define inverse cumulative distribution function of the standard normal as $Q(1-\alpha) = \inf\{x : \Phi(x) \geq 1 - \alpha\}$, we may rewrite the utility (3.13) as

$$\begin{aligned} u_A(x_* \mid D_t) &= \Pr\left[\frac{\tilde{\mu}_A - \tau}{\tilde{\nu}_A} > Q(1-\alpha)\right] \\ &= \Pr\left[\frac{\tilde{\mu}_A - \mu_A}{|\tilde{s}|} > \frac{-\mu_A + \tau + \tilde{\nu}_A Q(1-\alpha)}{|\tilde{s}|}\right] \\ &= \Phi\left(\frac{\mu_A - \tau - \tilde{\nu}_A Q(1-\alpha)}{|\tilde{s}|}\right). \end{aligned} \tag{3.14}$$

To solve for $\tilde{\nu}_A$ and $\tilde{s}$ in (3.14), notice that the lookahead variance $\tilde{\nu}_A^2$ (or $\tilde{\sigma}_A^2$) given x_* can be computed by (3.7) in the same way that ν_A^2 (or σ_A^2) is computed given the previous collection of

data points $x_1, \ldots, x_t$. To express $\tilde{s}^2$, notice the rule of total variance with fixed x_* and D_t is

$$\begin{aligned} \operatorname{Var}(f(A)) &= \mathbb{E}\operatorname{Var}[f(A) \mid \tilde{y}_*] + \operatorname{Var}\mathbb{E}[f(A) \mid \tilde{y}_*] \\ \Leftrightarrow \quad \sigma_A^2 &= \tilde{\sigma}_A^2 + \tilde{s}^2 \quad \Leftrightarrow \quad \nu_A^2 = \tilde{\nu}_A^2 + \tilde{s}^2, \end{aligned}$$

where the equivalence is due to $\tilde{\sigma}_A^2$ (or $\tilde{\nu}_A^2$) being constant for any realization of $\tilde{y}_*$.

As a result, there is only one free parameter between $\tilde{\nu}_A$ and $\tilde{s}$ in (3.14), where all the other variables, μ_A, τ, ν_A are independent of the choice of x_*. Further, both $\tilde{s}^2$ and $\tilde{\nu}_A^2$ (or $\tilde{\sigma}_A^2$) can be solved using the same closed-form GP posterior solution (3.7). For convenience in later analysis, we define:

$$\begin{aligned} \rho_A^* &= \rho(x_*, A \mid D_t) = \frac{\kappa(x_*, A \mid D_t)}{\nu(x_* \mid D_t)\nu(A \mid D_t)} \\ &= \operatorname{Corr}\left(\tilde{y}_*,\ \int w_A(x) f(x)\,\mathrm{d}x + \varepsilon_c \,\middle|\, x_*, D_t\right), \end{aligned}$$

where $\varepsilon_c \sim \mathcal{N}(0, c^2)$ results from the margin of probit transformation in FPMs, which is also an effective additive noise for region integrals (i.e., $c = 0$ for exact AAS). Straight-forward computation via (3.7) shows that

$$\tilde{\nu}_A^2 = (1 - {\rho_A^*}^2)\nu_A^2, \quad \text{and} \quad \tilde{s} = \rho_A^* \nu_A. \tag{3.15}$$

Then, we may rewrite (3.14) with only one free variable ρ_A^* that depends on the choice of x_*, as

$$\begin{aligned} u_A(x_* \mid D_t) &= \Phi\left(\frac{\mu_A - \tau - \nu_A\sqrt{1 - {\rho_A^*}^2}Q(1-\alpha)}{|\rho_A^*\nu_A|}\right) \\ &= \Phi\left(Q(1-\alpha)\frac{R_A - \sqrt{1 - {\rho_A^*}^2}}{|\rho_A^*|}\right), \end{aligned} \tag{3.16}$$

where the other variables that are independent of x_* are summarized by

$$R_A = \frac{Q(P(A \mid D_t))}{Q(1-\alpha)} = \frac{\frac{\mu_A - \tau}{\nu_A}}{Q(1-\alpha)},$$

which is an *exploitation* measure that indicates how close a region is to positive rewards in its current state. For any $\alpha < 0.5$ such that $Q(1-\alpha) > 0$, reward is assigned if and only if $R_A \geq 1$, i.e., $R_A < 1, \forall A \in \mathcal{A}_t$.

3.4 Analysis of the Closed-Form Greedy Solutions

The analytical solution (3.16) to the greedy maximization of expected rewards (3.11) with AAS and FPMs enables us to further study the theory behind the exploration/exploitation tradeoff of APPS in nontrivial cases, assuming:

1. the region pattern classifier is defined by AAS (3.1) or FPMs (3.2);
2. the regions are spatially separated such that every point query only affects the inference outcome of the region that contains the point;
3. only regions $A \in \mathcal{A}_t$ where positive reward has not been assigned are considered.

Particularly, Assumption 2 allows us to ignore the effect a data point has on regions other than its own and consider every region independently. We will answer two questions in this case:

1. which region will be explore next, and
2. what location will be queried for that region.

We start with the closed-form solution (3.16), which depends on R_A and $|\rho_A^*|$.

On the one hand, R_A depends only on collected data D_t and A, i.e., R_A is a measure of the current state. Notice that for any $1 - \alpha > 0.5$, we have $Q(1-\alpha) > 0$, which suggests that R_A is positively related to the current mean estimate of the region integral. In fact, R_A is an exploitation measure which indicates how close a region is to positive reward in its current state, using the ratio between the quantile statistic of the region classifier output and the minimum quantile for reward assignment. Given that $A \in \mathcal{A}_t$ has not been assigned positive reward, we may assume $R_A < 1$.

On the other hand, $\rho_A^* = \rho(x_*, A \mid D_t)$ further depends on the choice of x_* and is a measure of the quality of x_*. By (3.15), $\rho_A^* = \sqrt{1 - \frac{\tilde{\nu}_A^2}{\nu_A^2}}$, the measure of point choice only depends on the one-step lookahead variance reduction of the estimate of the region integral $\int w_A(x) f(x)\,\mathrm{d}x + \epsilon_c$.

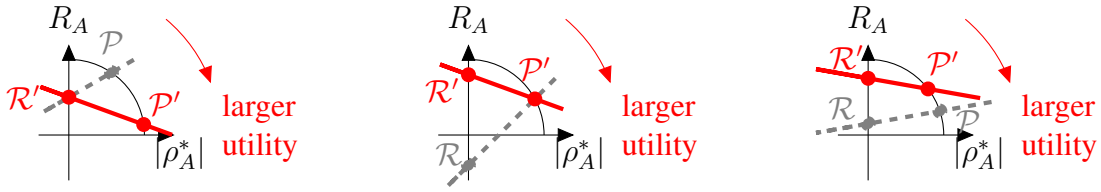
Considering every region independently, the design problem then reduces to optimizing ρ_A^* by choosing x_* so as to maximize (3.16). At this step, it is possible to take partial derivatives to find the maximimum ρ_A^* for (3.16). However, the analysis can be made easier if one realizes that, assuming $R_A < 1$, maximizing (3.16) is equivalent to minimizing the slope of the line joining the following two points $\mathcal{P}, \mathcal{R}$ in $\mathbb{R}^2$:

$$\mathcal{P} = \left(|\rho_A^*|, \sqrt{1 - {\rho_A^*}^2}\right), \qquad \mathcal{R} = \left(0, R_A\right).$$

In Figure 3.5(a), one can observe that the slope of the line can always be made smaller by either increasing $|\rho_A^*| = |\rho(x_*, A \mid D_t)|$, which results in moving the $\mathcal{P}$ point to the right along the arc of the unit circle, or moving $\mathcal{R}$ up.

With the help of Figure 3.5, we can conclude for regions that do not currently have a reward that

1. Fix the region A, $u_A(x_*, D)$ is maximized by simply choosing the location that maximizes $|\rho_A^*| = |\rho(x_*, A \mid D_t)|$. See Figure 3.5a.
2. Similarly, if two regions have equal marginal probability of matching the desired pattern R_A, then a region with a larger $|\rho_A^*|$ will be selected. See Figure 3.5a.
3. Comparing different regions, if two regions can be equally explored (i.e. they have the same $|\rho_A^*|$ value, e.g., resulting from both region having the same number of collected measurements at the same relative locations), then the region with the larger marginal probability of a matching outcome R_A will be selected. Figure 3.5b illustrates the comparison.



(a) Same region or same mean (b) Same number of observations (c) General condition

Figure 3.5: Illustration of selection criterion on independent regions. The solid red line with prime labels is preferred in each plot; it has a smaller slope.

4. In general, APPS will simultaneously consider both point 2 & 3 (i.e., exploitation and exploration), illustrated by Figure 3.5c.

Notice, through Figure 3.5, it can also be inferred that any region that has already been assigned reward will have $R_A \geq 1$, the optimal solution would take $\rho_A^* = 0$ and let the slope to be negative infinity. I.e., the optimal solution at regions with positive patterns (with at least $1 - \alpha$ probability where $\alpha < 0.5$) is to refuse collecting new observations. This observation further suggests that active search should not allow repeated rewarding of the same region, which is consistent with our discussion in Remark 3.1.

3.4.1 Equivalent Solution for Separated Regions

Since Figure 3.5 suggests that for every region $A \in \mathcal{A}_t$ where $R_A < 1$, the optimal solution is to choose observation x_* with the largest $|\rho_A^*|$ in order to reduce the variance in the estimate of the region integral, we may have the following alternative method that also greedily maximizes the expected reward (3.9), assuming that the regions are well-separated and the observation inside one region only affects the inference at the same region.

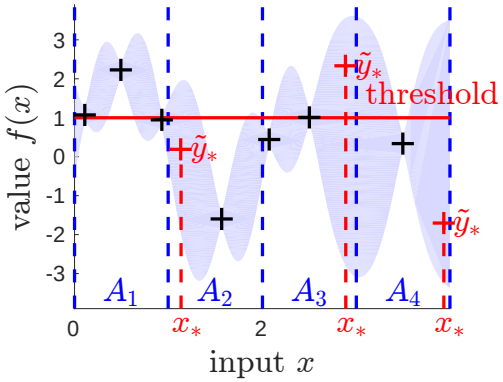


Figure 3.6: When regions are well-separated, maximizer for greedy expected reward must choose from the points that minimize the variance of the lookahead region integrals.

The alternative solution has two steps (illustrated in Figure 3.6):

1. For every region, optimize query location to minimize the variance of the region integral;
2. Choose the final design by evaluating (3.16) at the selected locations from Step 1.

3.4.2 Connection to Bayesian Quadrature, Σ-Optimality, and GP-SOPT

The problem of choosing locations to minimize the variance of region integral is studied in *Bayesian quadrature*, also known as *Bayesian Monte Carlo* Rasmussen and Ghahramani [2003].

Plain region integral is also connected to the problem active surveying (Section 2.4.1), which studies how to obtain the average value of a population. In this case, the population is all points in a region. As a result, $\min_D \sigma^2(A, A' \mid D) = \iint \kappa(x, x' \mid D) w_A(x) w_A(x') \,\mathrm{d}x \,\mathrm{d}x'$ is the Σ-optimality in active surveying problems (2.14).

When $c \to 0$, the solution to lookahead reduction of the variance of region integral uses

$$\begin{aligned}\tilde{s} = \rho_A^* \nu_A = \frac{\kappa(x_*, A \mid D_t)}{\nu(x_* \mid D_t)} &= \int \frac{\kappa(x_*, x \mid D_t)}{\sigma(x_* \mid D_t)} w_A(x) \,\mathrm{d}x \\ &= \int \rho_A^*(x_*, x \mid D_t) \sigma(x \mid D_t) w_A(x) \,\mathrm{d}x,\end{aligned}$$

which is related to the greedy application of Σ-optimality, though the original Σ-optimality focuses on application in Gaussian random fields where $\rho(x_*, x \mid D_t) \geq 0$ is guaranteed. With a GP, such sign guarantees may not hold.

Finally, even though greedy maximization of expected reward also boasts exploitation/exploration tradeoff, it has a different from than the tradeoff in multi-armed bandits. A typical solution for multi-armed bandits is GP-UCB Srinivas et al. [2010a], or its application with thresholding outcomes Locatelli et al. [2016] and with asynchronous application with variance estimates Zhong et al. [2017]. The basic greedy criterion is equivalent to

$$\max_{x_*} \mu(A \mid D_t) + \beta_t \nu(A \mid D_t). \tag{3.17}$$

Notice (3.17) cannot be used to select points because the criterion only depends on region statistics. To choose point observations in independent regions, one fix is to measure exploration via the change in Σ-optimality similar to the GP-SOPT algorithm (2.18), as

$$x_{t+1} = \arg\max_{x_*} \mu(A \mid D_t) + \beta_t \tilde{s}(x_*, A \mid D_t) = \arg\max_{x_*} \nu_A(Q(1-\alpha)R_A + \beta_t \rho_A^*). \tag{3.18}$$

Comparing (3.18) with the greedy solution to our utility function for region A (3.16), one may realize that both criteria are positively related to R_A and ρ_A^*, yet they take different forms. Direct application of GP-SOPT ignores the binary observation outcome, which is more important in our region search problems.

3.5 Simulations

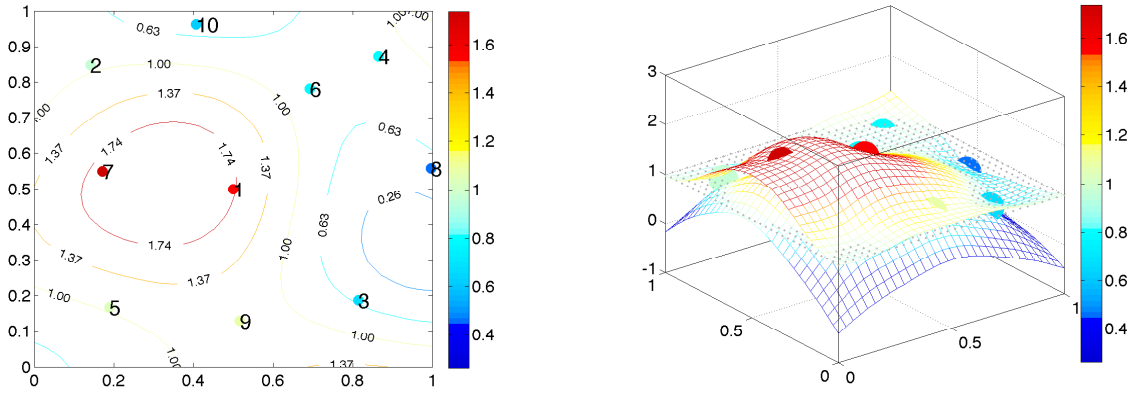
We used a list of simulated experiments to demonstrate properties and performance of AAS. More interestingly, we provide intuition about the behavior of AAS in multi-region cases, which we really care about.

In all simulations, the input space was the 2-dimensional Euclidean space and our function was generated from a GP whose prior mean was constant zero and whose prior covariance was the following isotropic square exponential kernel:

$$\kappa(x, x') = \sigma_f^2 \exp\left\{-\tfrac{1}{2\ell^2}(x - x')^\top (x - x')\right\} \tag{3.19}$$

where σ_f^2 and ℓ were set at different values in different cases to make the simulated problems interesting. Further, actual observations were simulated with additive noise $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$.

3.5.1 One Region Synthetic Data



(a) Sampling locations (b) Posterior estimation

Figure 3.7: One region search. Samples are selected in hope that with posterior distributions, the integral over the entire unit square is greater than 1 with probability at least 0.8.

The first demonstration/experiment was performed on a 2-dimensional unit square which contains only one region. The parameters used to generate the observations in (3.19) are $\ell = 0.33, \sigma_f^2 = {}^1/_{(2\pi\ell^2)} = 1.21^2, \sigma^2 = 0.1^2$. We purposefully made the problem difficult, so that AAS can run for a longer time period, by keeping the *a priori* variance of the integral over the region small, only roughly $\bar{\kappa}(A, A) = 0.737^2$. As a result, the region is not guaranteed to have high average values with high probabilities. We kept sampling function values on a 33×33 dense grid until the average value in the unit square region is greater than the threshold $\tau = 1$. AAS is expected to sequentially sample observations until it believes that the regional average is greater than τ with probability at least $1 - \alpha = 0.8$.

Figure 3.7 (a) visualizes the sampling locations determined by AAS in a sequential order. After these updates, the posterior marginal bandwidth of every point is shown in (b) and the gray mesh

at level 1.0 serves as a reference showing that the integral of the function, under posterior distribution, has high possibility to be greater than the threshold. The behavior of AAS is consistent with our analysis in Section 3.4. Before the algorithm terminates when it verifies that the region is significantly interesting, AAS explores locations which yield the maximal possible decrease of the variance of the integral once sampled, similar to experimental designs in BQ. The intuition is that function values at these points are usually unexplored and may become the best bet to attain a reward.

3.5.2 Multi-Region Synthetic Data

In this experiment, we simulated random GPs on a 2-D space which is externally split into 10×10 unit square regions. The goal was to find as many interesting regions as possible. Similar to before, a region may be flagged and rewarded if the posterior average function value on this region is greater than $\tau = 1$ with probability at least $1 - \alpha = 0.8$.

To allow interactions between regions, we chose a larger length scale for the prior GP.[4] The parameters selected are $\ell = 1, \sigma_f^2 = 1, \sigma_n^2 = 0.1^2$. The prior variance of the integral over any region is $\bar{\kappa}(A, A) = 0.924^2, \forall A \in \bar{\mathcal{A}}$ (roughly 14% regions are interesting). An illustration is in Figure 3.8(a), where the color of a region indicates the average function value in that region. Level sets of the function value are also plotted in (a).

The rest of Figure 3.8 compare the following algorithms

- **Active area search** (AAS): Our proposed method.
- **Level set estimation** (LSE): Gotovos et al. [2013] proposed this theoretically justified algorithm for level set estimations, which is to determine the region in the input space where the function value is close to h. We hope that by finding level sets for $h = \tau$ and recognizing even higher/lower regions, interesting regions may be discovered. Several other parameters were set as $\beta_t^{1/2} = 3, \varepsilon = 0.1$. (The original paper also set β_t fixed and broke theoretical guarantees in experiments.)
- **Uncertainty sampling** (UNC): Seo et al. [2000] used UNC to map the function value over the entire input space. UNC explores locations that have high marginal variance in the posterior distribution. The samples are sparse but blind to outcomes.
- **Random sampling** (RAND) serves as a baseline. It picks locations at random.

From these plots, we can see that AAS samples locations that are both sparse yet concentrate in regions which are more likely to have high average. It favors points on the boundary of multiple regions. It also explores new locations reasonably. The superiority of AAS in interesting region discovery is obvious.

[4] In reality, training can be done offline with pilot data. We usually match the order of region diameter and GP length scale when designing regions for preliminary real-world experiments.

LSE gives the second-best performance. While searching for level sets, LSE can identify positive regions inside. However, LSE is not aimed for this problem and thus it is hard to pin down which threshold and tolerance to ask for in LSE. Further, LSE may be too wasteful to precisely map the level set, and the observations that LSE makes may not lead to discovery of interesting regions. Finally, LSE may sometimes be pessimistic because of its theoretical guarantees and is sensitive to boundaries.

Finally UNC and RAND are the worst because they are generic and unspecific to the objective.

3.5.3 Repeated Experiments

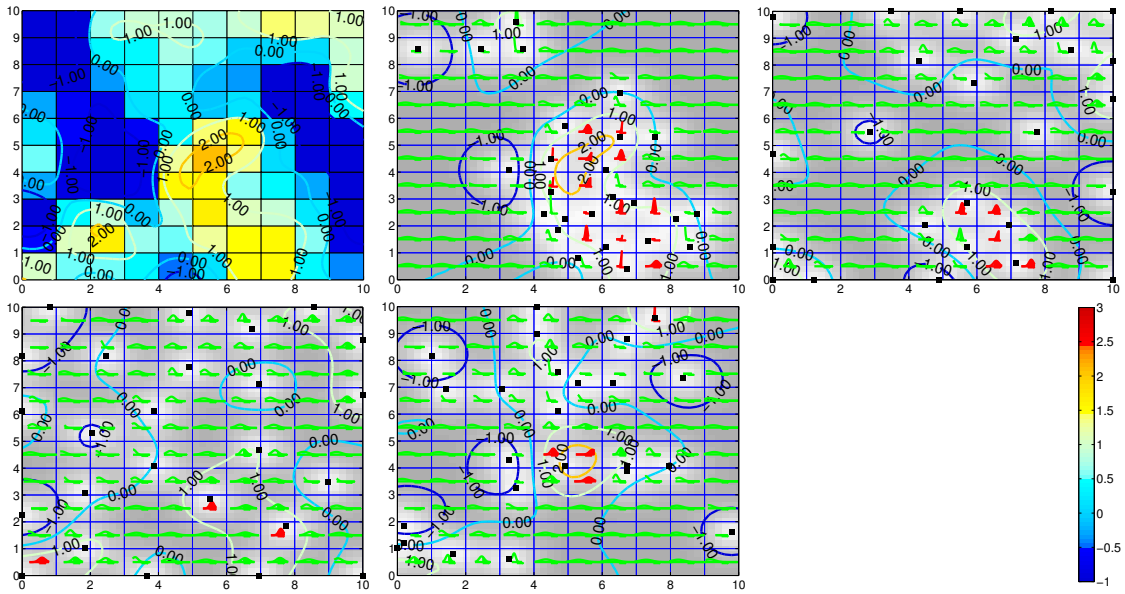


Figure 3.8: Multi-region. Shared color bar. (a) shows both function values and region averages. (b-e) show the first 25 locations sampled by different strategies (black dots). Gray scale indicate marginal variance. Red/green curves in region centers show the posterior tail distribution of the region averages. Red regions are reported.

We repeated our last experiment for 10 times with different functions generated through the same parameters. We report recall in Figure 3.9. Precision is a function of $1 - \alpha$ which is the same in all experiments so it is not reported. The curves indicate the average percent of positive regions reported given different query budgets. Standard error of the average is also reported.

Figure 3.9 shows that AAS outperformed other methods by a large margin. With 20 observations, AAS was able to discover half of the interesting regions. Notice in Figure 3.8, with 25 points, most parts of the function space remain gray even for UNC. The success of AAS mainly attributes to its relevancy to the objective.

LSE performed second best, about 60% as efficient as AAS. It can be observed from Figure 3.8 that LSE also biases towards areas near interesting regions. In contrast, neither UNC or RAND

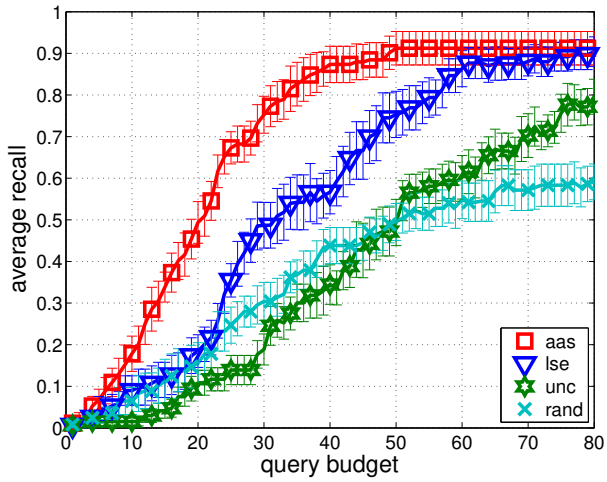


Figure 3.9: Repeated experiments on 10×10 regions

utilize sampling budgets efficiently. RAND is slightly better in the beginning because of its randomness yet UNC improves towards the end because it avoids the "coupon collector's problem."

3.6 Empirical Evaluation

We now turn to an empirical evaluation of our framework, in three different settings and with three different classifiers. Code and data for these experiments is available online.[5]

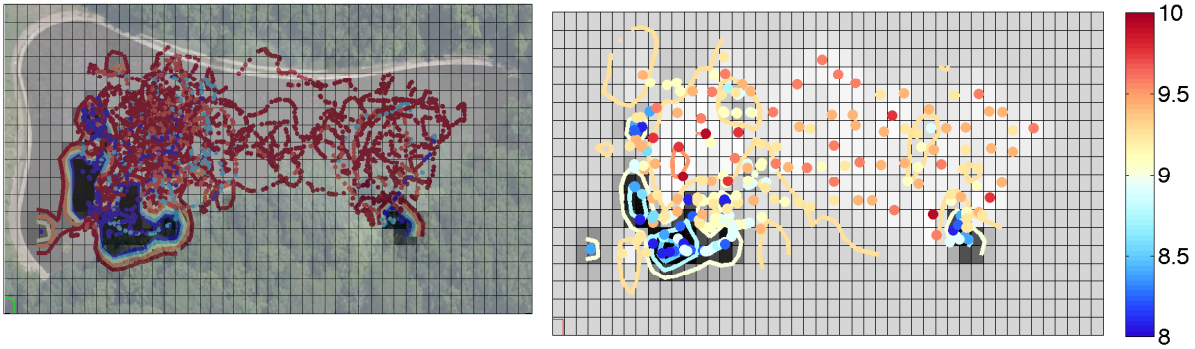
Precision plots are available in the appendix for completeness. Precision is determined primarily by the classifier and $1 - \alpha$, and thus does not vary much across methods.

3.6.1 Environmental Monitoring (Linear Classifier)

In order to analyze the performance of APPS with the MTC, we ran it on a real environmental monitoring dataset and compared to baseline algorithms. Valada et al. [2012] used small (60 cm) autonomous fan-powered boats to collect dissolved oxygen (DO) readings in a pond, with the goal of finding regions that are low in dissolved oxygen, an indicator of poor water quality. The data used in our experiment comes from a pond approximately 150 meters wide and 50 meters long. The mobile robots have a cell-phone module that records the time and location of every measurement. Because of physical limitations, the measurement reading does not stabilize for about one minute. Therefore, in data collection, the boat was moved back and forth in a single location, in the hope that the noise would cancel by averaging these measurements.

In order to verify our methods, we borrowed data from Valada et al. [2012], comprising 16 960 location/DO value pairs, and fit a GP model by maximizing the likelihood of the prior parameters

[5] https://github.com/AutonlabCMU/ActivePatternSearch/



(a) data in one run and true matching regions (black)

(b) APPS collected data and posterior region probability

Figure 3.10: Illustration of dataset and APPS selections for one run. A point marks the location of a measurement whose value is also reflected in its color. Every grid box is a region whose possibility of matching is reflected on gray-scale.

on 500 random samples seven times, taking the median of the learned hyperparameter values. We used a squared-exponential kernel with a learned length scale. We defined regions by covering the map with many windows of size comparable to the GP length scale, and used MTC parameters $b = -9$, $c = -100$. Data points and classifier probability outputs for the ground truth are shown in Figure 3.10a, which also shows the learned length scale (roughly 3 meters).

We then repeated the following experiment: we randomly sampled 6 000 points at a time from data points not used for GP parameter training, and randomly selected 10 of these 6 000 points to form an initial training set D. We then used several competing methods to sequentially make further queries until 300 total observations were obtained. The considered algorithms were: APPS with analytical solutions, APPS with one draw of z_* at each candidate location, AAS in Ma et al. [2014] with analytical solutions, AAS with sampling, the level set estimation (LSE) algorithm of Gotovos et al. [2013] with parameters $\beta^t = 6.25$ and $\varepsilon = 0.1$, uncertainty sampling (UNC), and random selection (RAND). Each algorithm chose queries based on its own criterion; the quality of queried points was evaluated by the MTC classifier with the above parameters and was then compared with true region labels that were computed by MTC using all 6 000 data points. A 70% marginal probability was chosen to be required for a region to be classified as matching $(1 - \alpha = 0.7)$.

Figure 3.11 reports the mean and standard error of the recall of matching regions over 15 repetitions of this experiment. APPS and AAS with both analytical solutions and sampling performed equally well here. The similarity between APPS and AAS is also expected because in linear problems, the choice of c is a fine-tuning problem, which does not show its impact on this real dataset. Notice that AAS is not able to handle any other classifier-based setting; this is the core contribution of APPS. To understand why analytical solutions were similar to sampling, notice that the data collection locations have to be constrained to those actually recorded, which makes it easier to obtain a near-optimal decision.

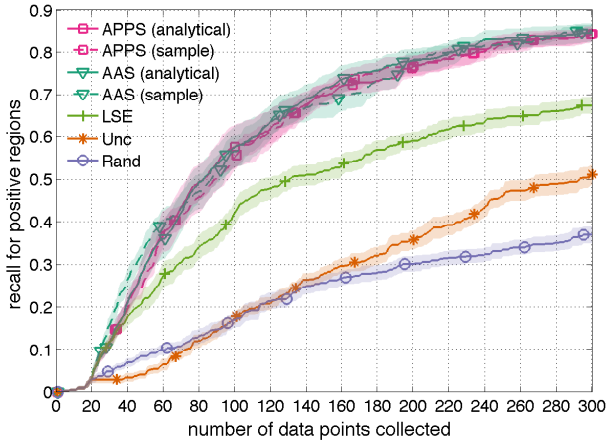


Figure 3.11: Recall curves for pond monitoring experiment. Color bands show standard errors after 15 runs.

The second group in performance ranking is the LSE method. We attempted to boost its performance by selecting its parameters to directly optimize the area under its recall curve, which was, in a sense, cheating. On further analysis of its query decisions, we saw LSE making, for the most part, qualitatively similar selection decisions to APPS. LSE will stop collecting data in a region if there is enough confidence, but does not specifically try to push regions over the threshold, and so its performance on this objective is inferior.

Last in the comparison are RAND and UNC. It is interesting to observe that RAND was initially better than, but later crossed by UNC. In the beginning, since UNC is purely explorative, its reward uniformly remained low across multiple runs, whereas in some runs RAND queries can be lucky enough to concentrate around matching regions. At a later phase, RAND faces the coupon collector's problem and may select redundant boring observations, when UNC keeps making progress at a constant rate.

Figure 3.10b illustrates the selection locations for our APPS method. This plot shows that our APPS method can obtain reasonable data to both explore the available space and gain enough information around the matching regions.

Remark 3.2. *In the example of environmental monitoring, we assumed that sensing is expensive relative to the cost of motion. This is reasonable in this case because of hysteresis in the sensor. It must remain stationary for awhile to collect an accurate measurement. In the case of our actual data, it was not collected that way. the boat moved continuously. This brings up two issues:*

1. *Can we correct for the hysteresis in the data set we used.*
2. *In cases where the assumption does not hold, how might we correctly choose experiments when the travel cost is significant. In the case of either assumption (cheap travel, expensive sensing or expensive travel, expensive sensing) the optimal solution could be written down as a POMDP (e.g. as is described in Garnett et al. [2012]), but that would be intractable to solve in general. In the case of cheap travel we were able to present a good greedy algorithm*

that is tractable. In the case of expensive travel, it remains an interesting open question whether a good greedy algorithm exists.

3.6.2 Predicting Election Results (Linear Classifier)

Consider the problem of a state-level political party official who wishes to determine which races will be won, lost, or might go either way. As surveying likely voters is relatively expensive, we would like to do so with as few surveys as possible.

In a simple model of this problem, the problem of finding races which will be won is a natural fit to a classifier of the form $h_g(f) = \Phi\big(w^\top f(\Xi_g) + b_g\big)$. Our function f maps from the voting precincts in the state to the vote share of a given party in that district, with a covariance kernel defined by demographic similarity and geographic proximity. To account for multiple races taking place in each district (e.g., state and national legislators), we duplicate each precinct with a flag for the type of election. If g is the set of all precincts participating in a particular race and w_g is some constant c times the voting population of each precinct, then $w^\top f(\Xi_g)$ gives c times the total vote portion for the given party in that election. In a simple model which ignores turnout effects, the probability of winning a race is essentially 1 if the underlying proportion is greater than 0.5 and 0 otherwise; this can be accomplished by setting c to some fairly large constant, say 100, and $b = -\frac{1}{2}c$. (An equally simple model that nonetheless more thoroughly accounts for unmodeled effects would just use a smaller value of c.)

We ran experiments based on this model on 2010 Pennsylvania election returns [Ansolabehere and Rodden]. For each voting precinct in the dataset, we used the 2010 Decennial Census [United States Census Bureau, 2010] to obtain a total population count and percentages of the population for gender, race, age, and housing type categories; we also added an (x, y) location based on a Lambert conformal conic projection of point in the precinct, and used these features in a squared-exponential kernel. The data for each precinct was then replicated three times and associated with Democratic vote shares for its U.S. House of Representatives, Pennsylvania House of Representatives, and Pennsylvania State Senate races; the demographic/geographic kernel was multiplied by a positive-definite covariance matrix amongst the races. We learned the hyperparameters for this kernel by maximizing the likelihood of the model on full 2008 election data.

Given the kernel, we set up experiments to predict 2010 races based on surveying an individual voting precinct at a time. For simplicity, we assume that a given voting precinct can be thoroughly surveyed (and ignore turnout effects, voters changing their minds over time, and so on); thus observations were made with the true vote share. We seeded the experiment with a random 10 (out of 16 226) districts observed; APPS selected from a random subset of 100 proposals at each step. We again used $1 - \alpha = 0.7$.

Figure 3.12 shows the mean and standard errors of 15 runs. APPS outperforms both random and uncertainty sampling here, though in this case the margin over random sampling is much narrower. This is probably because the portion of regions which are positive in this problem is much higher, so more points are informative.

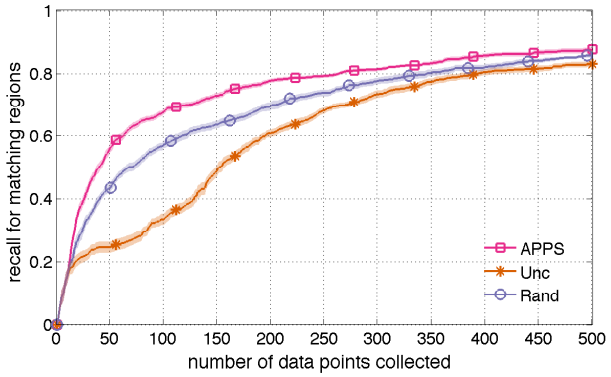


Figure 3.12: Recalls for election prediction. Color bands show standard errors after 15 runs.

Uncertainty sampling is in fact worse than random here, which is not too surprising because the purely explorative nature of UNC is even worse on the high dimensional input space of this problem.

LSE and AAS are not applicable to this problem, as they have no notion of weighting points (by population).

3.6.3 Finding Vortices (Black-Box Classifier)

We now turn to more complex pattern classifiers by studying the task of identifying vortices in a vector field based on limited observations of flow vectors. Linear classifiers are insufficient for this problem,[6] so we will demonstrate the flexibility of our approach with a black-box classifier.

To illustrate this setting, we consider the results of a large-scale simulation of a turbulent fluid in three dimensions over time in the Johns Hopkins Turbulence Databases[7] [Perlman et al., 2007]. Following Sutherland et al. [2012], we aim to recognize vortices in two-dimensional slices of the data at a single timestep, based on the same small training set of 11 vortices and 20 non-vortices, partially shown in Figure 3.13(a).

Recall that h_g assigns probability estimates to the entire function class $\mathcal{F}$ confined to region g. Unlike the previous examples, it is insufficient to consider only a weighted integral of f. Instead, though, we can consider the average flow across sectors (angular slices from the center) of our region as building blocks in detecting vortices. We count how many sectors have clockwise/counter-clockwise flows to give a classification result, in three steps:

1. First, we divide a region into K sectors. In each sector, we take the integral of the inner product between the actual flow vectors and a template. The template is an "ideal" vortex,

[6] The set of vortices is not convex: consider the midpoint between a clockwise vortex and its identical counter-clockwise case.

[7] `http://turbulence.pha.jhu.edu`

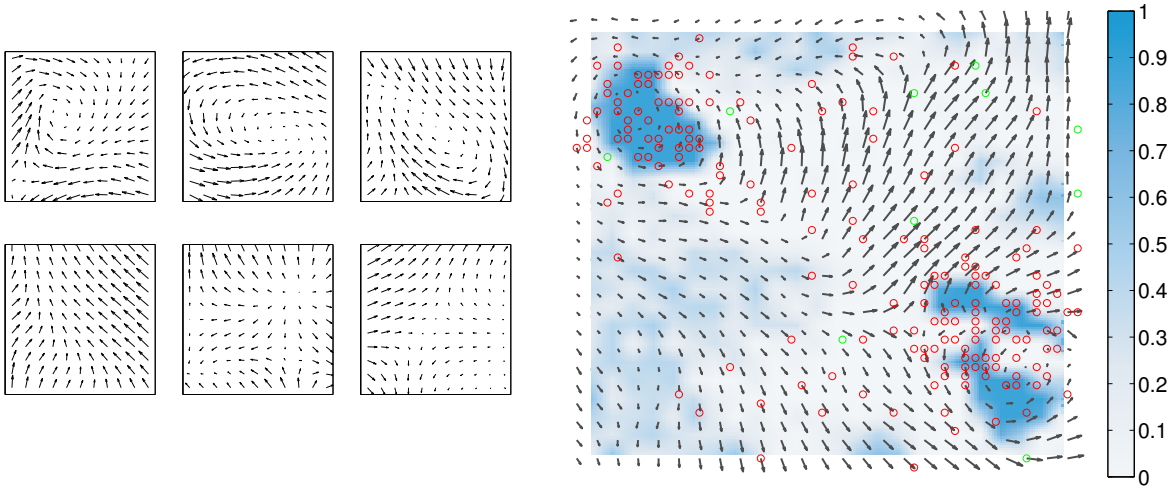
but with larger weights in the center than the periphery. This produces a K-dimensional summary statistic $L_g(f)$ for each region.

2. Next, we improve robustness against different flow speeds in the data by scaling $L_g(f)$ to have maximum entry 1, and flip its sign if its mean is negative. Call the result $\tilde{L}_g(f)$.
3. Finally, we feed the normalized $\tilde{L}_g(f)$ vector through a 2-layer neural network of the form

$$h_g(f) = \sigma\left(w_{\text{out}} \sum_{i=1}^{K} \sigma\left(w_{\text{in}} \tilde{L}_g(f)_i + b_{\text{in}}\right) + b_{\text{out}}\right),$$

where σ is the logistic sigmoid function.

$L_g(f) \mid D$ obeys a K-dimensional multivariate normal distribution, from which we can sample many possible $L_g(f)$, which we then normalize and pass through the neural network as described above. This gives samples of probabilities h_g, whose mean is a Monte Carlo estimate of (3.4).



(a) (b)

Figure 3.13: (a): Positive (top) and negative (bottom) training examples for the vortex classifier. (b): The velocity field used; each arrow is the average of a 2×2 square of actual data points. Background color shows the probability obtained by each region classifier on the 200 circled points; red circles mark points selected by one run of APPS initialized at the green circles.

We used $K = 4$ sectors, and the weights in the template were fixed such that the length scale matches the distance from the center to an edge. The network was optimized for classification accuracy on the training set. We then identified a 50×50-pixel slice of the data that contains two vortices, some other "interesting" regions, and some "boring" regions, mostly overlapping with Figure 11 of Sutherland et al. [2012]; the region, along with the output of the classifier when given all of the input points, is shown in Figure 3.13(b). We then ran APPS, initialized with 10 uniformly random points, for 200 steps. We defined the regions to be squares of size 11×11 and spaced them every 2 points along the grid, for 400 total regions. We again thresholded at $1 - \alpha = 0.7$. We evaluate (3.4) via a Monte Carlo approximation: first we took 4 samples of z_*,

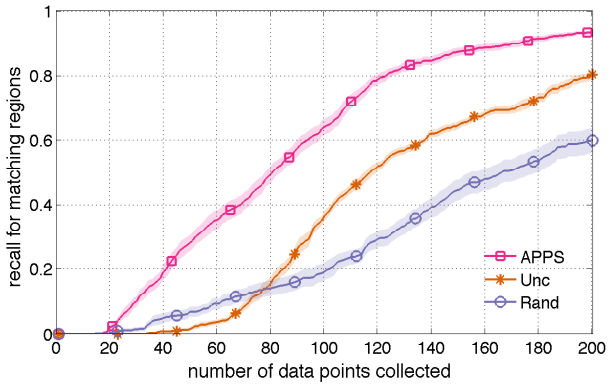


Figure 3.14: Mean recalls over the search process on the vortex experiment. Color bands show standard errors after 15 runs.

and then 15 samples from the posterior of f over the window for each z_*. Furthermore, at each step we evaluate a random subset of 80 possible candidates x_*.

Figure 3.14 shows recall curves of active pattern search, uncertainty sampling, and random selection, where for the purpose of these curves we call the true label the output of the classifier when all data is known, and the proposed label is true if $T_g > 1 - \alpha$ at that point of the search (evaluated using more Monte Carlo samples than in the search process, to gain assurance in our evaluation but without increasing the time required for the search). We can see that active pattern search substantially outperforms uncertainty sampling and random selection. As in Section 3.6.1, uncertainty sampling was initially bad but later surpassed random selection, for the same reason.

3.7 Conclusions

We have introduced the general active area pattern search problem, where we seek to discover specific local patterns exhibited by an underlying smooth function with a limited observation budget. We proposed a framework built on Bayesian decision theory for the sequential active selection of observations so as to maximize the expected number of matching locations discovered at termination. We derived analytical forms for the required quantities for a broad class of models, and demonstrated the method's efficacy across three very different settings, using two different analytical classifier forms and one based on sampling.

We assumed that sensing is expensive relative to the cost of motion. In the case of environmental monitoring, this is reasonable because of hysteresis in the sensor. It must remain stationary for awhile to collect an accurate measurement. This brings up two future research questions: (1) Can we correct for the hysteresis in the data set we used? (2) In cases where the assumption does not hold, how might we correctly choose experiments when the travel cost is significant. It remains an open question whether a good greedy algorithm exists. One could include travel costs in the utility function and apply greedy maximization of the augmented utility. However, I speculate

that such an algorithm would not perform near-optimal, because it requires multi-step lookaheads and surveying Σ-objectives are not known to be submodular for a general GP. Besides, the utility function is to maximize the sum of expected reward, rather than a single region.

4

Active Needle Search with Region Sensing

4.1 Introduction

Active needle search describes the problem where an agent is given a target to search for in an unknown environment and actively makes data-collection decisions so as to locate the target as quickly as possible. Examples of this setting include using aerial robots to detect gas leaks, radiation sources, and human survivors of disasters. The statistical principles for efficient designs of measurements date back to Gergonne [1815], but the growing trend to apply automated search systems in a variety of environments and with a variety of constraints has drawn much research attention recently, due to the need to address the disparate aspects of new applications.

One possibility in such active search scenarios we aim to explore, inspired by the robotic aerial search setting but with statistical insights that we hope to generalize, is the opportunity to take aggregate measurements that summarize large contiguous regions of space. For example, an aerial robot carrying a radiation sensor will sense a region of space whose area depends on its altitude. How can such a robot dynamically trade off the ability to make noisier observations of larger regions of space against making higher-fidelity measurements of smaller regions?

To simplify the discussion, we will limit such *region sensing* observations to reveal the average value of an underlying function on a rectangular region of space, corrupted by independent observation noise. Noisy binary search is a simple realization of active search using such an observation scheme. This mechanism turns out to be sufficiently informative in the cases that we analyze to offer insights into a variety of search problems.

The ability to make aggregate region measurements in noisy environments has rarely been considered in previous work. *Bayesian optimization,* which has been used for localization of sparse

signals [Carpin et al., 2015, Hernández-Lobato et al., 2014, Jones et al., 1998, Ma et al., 2015a], usually considers only point measurements of an objective function. Notice that point observations can be considered in our framework if the allowed region sensing actions are constrained to be arbitrarily small. On the other extreme, *compressive sensing* [Candès and Wakin, 2008, Donoho, 2006, Wainwright, 2009], considers scenarios where every measurement can reveal information about the entire environment through linear projection with arbitrary coefficients. This is not always a realistic assumption, as for example for an aerial robot, which can only sense its immediate vicinity. Between the two extremes, Abbasi-Yadkori [2012], Carpentier and Munos [2012], Haupt et al. [2009], Jedynak et al. [2012], Rajan et al. [2015], Yue and Guestrin [2011] considered policies for search where observations can be made on any arbitrary subset of the search space, including discontiguous subsets, which is also often incompatible with the constraints in physical search systems.

Another assumption we make, common for example in compressive sensing, is *sparsity.* We assume that there are only a small number of strong signals in the environment; our goal is to recover these signals. Sparsity is necessary for the definition of active search problems; otherwise, for dense or weak signals, there is usually no better search approach than simply exhaustively mapping the entire space.

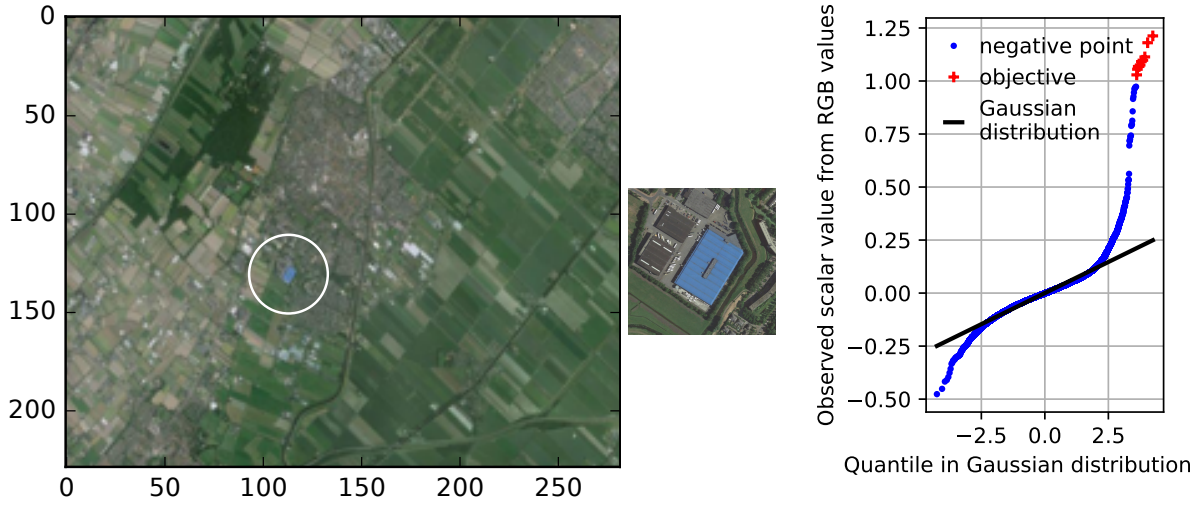
In addition to applicability in real search settings, sparsity has unique mathematical properties when considered alongside region sensing. In unconstrained sensing, Arias-Castro et al. [2013] discovered a paradox that active compressive sensing (that is, the ability to adaptively select observations based on previously collected data) does not improve detection efficiency beyond logarithmic terms over random compressive sensing. This limitation is seen also when considering theoretical detection rates for active compressive sensing methods [Abbasi-Yadkori, 2012, Carpentier and Munos, 2012, Haupt et al., 2009]. However, we show that active learning can in fact offer significant improvements in detection rates when observations are constrained to contiguous regions.

We propose an algorithm we call *Region Sensing Index* (RSI) that actively collects data to search for sparse signals using only noisy region sensing measurements. RSI is based on greedy maximization of information gain. Although information gain is a classic principle, we believe that its use in the recovery of sparse signals is novel and a good fit for robotic applications. We show that RSI uses $\tilde{O}(n/\mu^2 + k^2)$ measurements to recover all of k true signal locations with small Bayes error, where μ and n are the signal strength and the size of the search space, respectively, assuming unit noise per measurement (Theorem 4.4). The number of measurements with RSI is comparable with the rates offered by unconstrained compressive sensing, even though our constraints seem strong (i.e., region sensing loses all spatial resolution inside the region of measurement). Furthermore, we show that all passive designs under our contiguous region sensing constraint in $1d$ search spaces are fundamentally worse, with efficiency no better than sequential scanning of every point location, however strong the signals are. These results provide evidence to promote the use of and research into active methods.

4.1.1 Demo Active Needle Search

To demonstrate the desired properties of an active search algorithm, we simulated an active search scenario using a satellite image (Figure 4.1) where the objectives are all of the blue pixels. This demo directly simulates search and rescue in open areas based on life jacket colors or communication signals and also share similarities with gas leaks or radiation detection, where real data is usually sensitive or expensive.

In this demo, the objectives are found as the roof of a building, circled near the center of the satellite image. We used the scalar values due to an affine transformation from the original RGB values with a predefined matrix that separates the objective blue color and most other colors. The distribution of pixel values is shown in Figure 4.1(c).

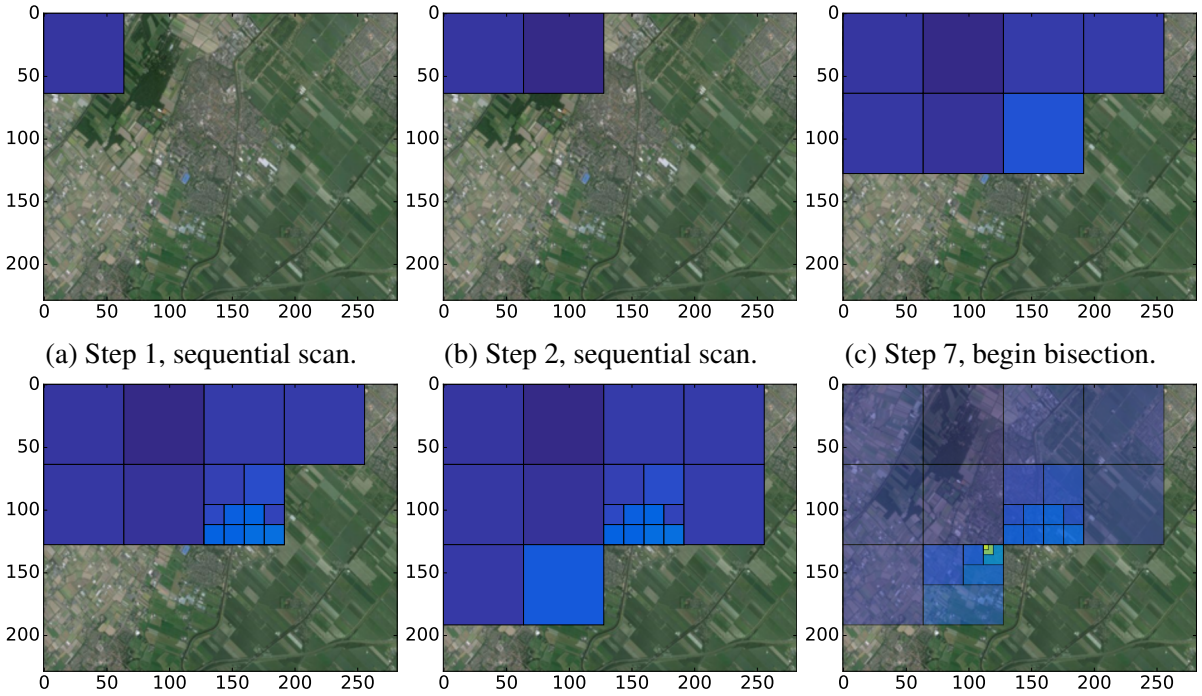


(a) Satellite image and target blue pixels (circled) (b) Enlarged (c) Distribution of pixel values, the goal is to localize the top 10 pixels.

Figure 4.1: Demo active search on a satellite image.

The active search algorithm controls a mobile sensor that is a single-pixel camera that records the average values in any chosen square regions. For simplicity, the side length of a feasible region must be a power of 2 and for every region size, we only consider the set of square regions that cover the entire search space with no gaps or overlaps. As a result, every larger region contains 4 regions of the next smaller size. The construction of the feasible regions resembles a spatial pyramid [Lazebnik et al., 2006].

Figure 4.2 shows the sequential measurement choices of RSI and their outcomes in a blue-to-yellow color scheme. RSI starts with measurements using region sizes that balances fidelity and coverage, so as to maximize measurement efficiency. Then, after the 7th measurement where a large outcome is observed, RSI is expected to investigate at subregion levels which have high probability to contain the a signal source. However, by the 19th measurement, further evidence indicates an overall low likelihood for the signal to originate from the subregions and RSI decides



(a) Step 1, sequential scan. (b) Step 2, sequential scan. (c) Step 7, begin bisection.

(d) Step 19, negative evidence. (e) Step 22, continue scan. (f) Step 36, found signal hotspot.

Figure 4.2: A desirable sequence of measurement designs realized by RSI. Only region averages are observed and their values are reflected in a blue-to-yellow color scheme.

to continue scanning at the optimal region size. Finally, with merely 36 measurements, RSI successfully locates one true signal source. In comparison, the image contains $36\,000$ pixel points.

As one can see, there are several properties for active needle search with region sensing:

1. The signals are usually significantly strong to allow information to be generated from aggregate region measurements.
2. The noise is artificial, used to model the decrease in information one can obtain from a measurement as the region size increases. We can start by approximating the noise as spatially independent when the region is large, though in the demo, we took the estimated standard deviation from the true average values at a feasible size (see Table 4.1).
3. On the other hand, the noise is constant across time-steps. I.e., repeating a measurement does not provide any new information. An efficient algorithm should be robust to noise modeling errors. For example, a Bayesian solution may decide to visit a region with less evidence when the alternatives are equally bad, due to model errors.
4. It is desirable to have upper bounds on the number of experiments. The bounded number should decrease as the Signal-to-Noise Ratio (SNR) increases, until $O(\log n)$, realizable by noiseless bisection search, where n is the size of the search domain.

We will propose and examine *Region Sensing Index* (RSI) for these properties.

Table 4.1: Signal and noise in demo experiment

Region size	1×1	2×2	4×4	8×8	16×16	32×32	64×64
Average in regions with needles (otherwise zero)	1.10	0.95	0.74	0.38	0.14	0.05	0.02
Standard deviation of region averages	0.06	0.06	0.05	0.04	0.03	0.02	0.01
SNR (row1 ÷ row2)	17.73	16.30	14.43	9.29	4.71	2.51	1.33

4.1.2 Related Work

Arias-Castro et al. [2013] proved that the minimax sample complexity[1] for any (i.e., potentially adaptive) algorithm to recover k sparse signal locations is at least $\Omega(\frac{n}{\mu^2})$, analyzing the problem in terms of the mean-squared error in the recovery of the underlying signal values. The authors also showed that a passive *random* design, combined with a nontrivial inference algorithm, e.g., Lasso [Wainwright, 2009] or the Dantzig selector [Candes and Tao, 2007], can have similar recovery rates (up to $O(\log n)$ terms). This result was presented as a paradox, suggesting that the folk statement that active methods have better sample complexity is not always true. Here we show that active search can make a substantial difference in recovery rates when the measurements are subject to the physically plausible constraint of region sensing, especially if the physical space has low dimensions.

Malloy and Nowak [2014] presented the first *active* search algorithm that achieves the minimax sample complexity for general $k \geq 1$. The complexity is the largest value among $O(\frac{n}{\mu^2})$, $O(k)$, and $O(\log n)$. The algorithm is called Compressive Adaptive Sense and Search (CASS) and it can be adapted to region sensing in one-dimensional physical spaces. CASS directly extends bisection search, by allocating different sensing budgets to measurements at different bisection levels so as to minimize the cumulative error rates. Interestingly, CASS is provably rate-optimal even considering other sensing mechanisms that assign different weights to different points, which effectively encode localization information in every measurement. That information turned out to be negligible for the model that is considered by the authors and similar to ours.

However, CASS may fail if the repeated measurements of the same regions do not contain perfectly independent noise. It also has the limitation that it requires knowledge of the sensing budget *a-priori*, yet produces no signal localization results until the very last measurements at the lowest level. Our paper addresses these practical issues with a redesigned active search algorithm using the Bayesian approach, which compares evidence instead of blindly trust the assumptions, and we use Shannon-information criteria, which implies bisection search in noiseless one-sparse cases.

Braun et al. [2015] also used Shannon-information criteria for active search but did not analyze their sample complexity under noisy measurements. Jedynak et al. [2012], Rajan et al. [2015]

[1] Sample complexity is equivalent to the number of measurements.

studied a similar search problem where the "regions" are relaxed to any unions of disjoint subsets.

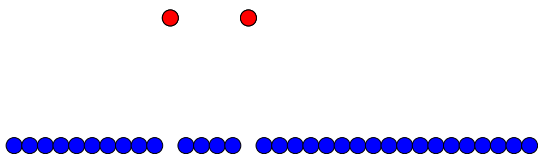
4.2 Problem Formulation

Consider a discrete space that is the Cartesian product of one-dimensional grids, $\mathcal{X} = \prod_{i=1}^{d}[n_i]$; $[n] = \{1, \ldots, n\}$. Let $n = \prod n_i$ be the total number of points in $\mathcal{X}$ (here the product symbol is the arithmetic rather than the Cartesian product). We presume there is a latent real-valued nonnegative vector $\boldsymbol{\beta} \in \mathbb{R}^n$ that represents the vector of true signals at all locations in $\mathcal{X}$. We further assume that $\boldsymbol{\beta}$ is sparse: it has value $\mu > 0$ on $k \ll n$ locations in $\mathcal{X}$ and has value 0 elsewhere. We consider making observations related to $\boldsymbol{\beta}$ through rectangular region sensing measurements, defined by

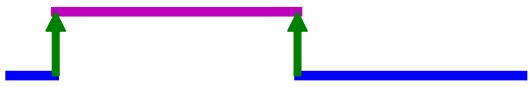
$$y_t = \mathbf{x}_t^\top \boldsymbol{\beta} + \varepsilon_t, \text{ s.t. } x_{tj} = w_t 1_{j \in A_t}, \ \varepsilon_t \sim \mathcal{N}(0, \sigma_t^2).$$

Here $\mathbf{x}_t \in \mathbb{R}^n$ is a sensing vector that has support on $A_t \subseteq \mathcal{X}$, a rectangular subset of $\mathcal{X}$. We assume that the sensing vector has equal weight w_t across its support. The resulting measurement, y_t, is equal to the mean value of $\boldsymbol{\beta}$ on A_t corrupted by independent Gaussian noise with variance σ_t^2. Note that selecting A_t suffices to specify the measurement location.

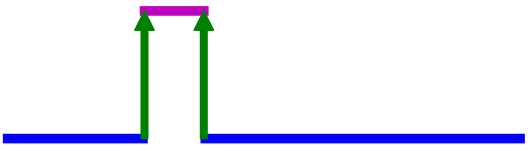
In $1d$ search environments, A_t may be any interval of $[n]$, and the corresponding design takes the form $\mathbf{x}_t = (0, \ldots, 0, w_t, \ldots, w_t, 0, \ldots, 0)^\top$. In higher search dimensions, we consider only regions that are contained in a hierarchical spacial pyramid [Lazebnik et al., 2006], i.e., a sequence of increasingly finer grid boxes with dyadic side lengths to cover the space at multiple resolutions.



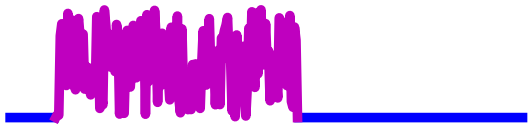
(a) Sparse signal vector, with n points, of which $k(\ll n)$ are nonzero with value $\mu(\gg 0)$.



(b) Measurement vector corresponding to a region, with uniform weight $w_t = \frac{1}{\sqrt{\|\mathbf{x}_t\|_0}}$ inside.



(c) Smaller regions have larger sensing weights.



(d) CS-type measurements infeasible because nonuniform weights encode location information.

Figure 4.3: Visualization of sparse signals and region sensing measurements in a $1d$ environment.

Our goal is to choose a sequence of designs $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T$ so as to discover the support of $\boldsymbol{\beta}$ with high confidence. Given a particular confidence, we will measure sample complexity by assuming $\|\mathbf{x}_t\|_2 = 1$ and $\sigma_t \equiv 1$ for each measurement and count the total number of measurements required to achieve that confidence, T. Letting $\|\mathbf{x}_t\|_2 = 1$ implies $w_t = \frac{1}{\sqrt{\|\mathbf{x}_t\|_0}}$, which can be seen as a *relaxed* notion of the region average, because the signal strength of a region measurement, which is μw_t, still decreases as the region size $\|\mathbf{x}_t\|_0$ increases.

Remark 4.1. *In fact, the most important measure for developing algorithms and comparing rates is the Signal-to-Noise Ratio (SNR) of an aggregate measurement. In this sense, our model has an alternative explanation with physical basis. It is equivalent to directly measuring the plain average in a region, if we assume that every point has an independent standard Gaussian noise that perturbs the observed average in the aggregate (similar to our discussion in the demo experiment).*

To show the equivalence, for any region with size $a_t = \|\mathbf{x}_t\|_0$, the new model assigns measurement weight $\tilde{w}_t = \frac{1}{a_t}$ to every point inside the region and expects to observe a mean of $\frac{\mu}{a_t}$ per true signal hotspot in the region. As for noise, due to spatial independence, the final observed noise follows a Gaussian distribution with standard deviation $\tilde{\sigma}_t = \frac{1}{\sqrt{a_t}}$. The final SNR of this measurement is $\frac{\mu\tilde{w}_t}{\tilde{\sigma}_t} = \frac{\mu}{\sqrt{a_t}}$, which equals to the SNR with the same region in our original model.

The measure of T is made to be comparable with another common choice of sample complexity, the Frobenius norm of the entire design $\|\mathbf{X}\|_F^2$, when the rows of $\mathbf{X}$ are normalized [Arias-Castro et al., 2013]. However, the normalization is often overlooked in classical compressive sensing, which allows algorithms to cheat in region sensing by making an enormous number of measurements of small weight and changing the sensing locations frequently. Another measure of complexity is to measure both $\|\mathbf{X}\|_F^2$ and the number of location changes simultaneously [Malloy and Nowak, 2014]. However, our discretized counting of measurements is conceptually simpler.

Our analysis is Bayesian and we will analyze performance in expectation, with prior $\boldsymbol{\beta} \sim \pi_0(\boldsymbol{\beta})$, a uniform distribution on the model class, $\mathcal{S}_\mu\binom{n}{k}$, which includes all k-sparse models with μ signal strength among n locations (i.e., it has $\binom{n}{k}$ possible outcomes). The Bayes risk will be measured by the expected Delta loss, $\bar{\epsilon}_T = \frac{1}{k}\mathbb{E}|S\Delta\hat{S}_T|$, where $\hat{S}_T$ is the best estimator of the k signal locations after T measurements and Δ is the symmetric difference operator on a pair of sets.

4.3 Proposed Methods

We note that region sensing loses all spatial resolution inside the region of measurement. Here we borrow ideas from noisy binary search, which has a similar property, and use information gain (IG) to drive the observation process. We name our algorithm *Region Sensing Index* (RSI, Algorithm 4.1). Like other active learning algorithms, RSI is a combination of an *inference* subroutine that constantly updates the distribution of $\boldsymbol{\beta}$ using the collected data and a *design*

Algorithm 4.1 Region Sensing Index (**RSI**)

Require: $\pi_0(k, n, \mu), T$ or ϵ, and the unknown $\boldsymbol{\beta}^*$
Ensure: $\hat{S}_t$ // (4.4)
1: **for** $t = 1, 2, \ldots$ **do**
2: pick $\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{X}} I(\boldsymbol{\beta}; y \mid \mathbf{x}, \pi_{t-1})$ // (4.2)&(4.3)
3: observe $y_t = \mathbf{x}_t^\top \boldsymbol{\beta}^* + \varepsilon_t$
4: update $\pi_t(\boldsymbol{\beta}) \propto \pi_{t-1}(\boldsymbol{\beta}) p(y_t \mid \boldsymbol{\beta}, \mathbf{x}_{t-1})$ // (4.1)
5: find $(\bar{\epsilon}_t, \hat{S}_t) = \arg\min_{|\hat{S}|=k} \frac{1}{k}\mathbb{E}\big[|\hat{S}\Delta S| \mid \pi_t\big]$ // (4.4)
6: break if $t \geq T$ or $\bar{\epsilon}_t < \epsilon$, if either is defined
7: **end for**

subroutine that chooses the next region to sense based on the latest information from the inference subroutine.

The inference subroutine. We use exact Bayesian inference with a uniform prior $\pi_0(\boldsymbol{\beta})$ on the model class $\mathcal{S}_\mu\binom{n}{k}$. Denote the outcome of the first t measurements as $\mathcal{D}_t = \{(\mathbf{x}_\tau, y_\tau) : 1 \leq \tau \leq t\}$. Even though $\mathcal{D}_t$ contains a dependent sequence of data collections, where $\mathbf{x}_\tau$ depends on $\mathcal{D}_{\tau-1}, \forall\tau$, Bayesian inference decomposes into a series of efficient updates:

$$\begin{aligned}\pi(\boldsymbol{\beta} \mid \mathcal{D}_t) &\propto \pi(\boldsymbol{\beta}) p(\mathcal{D}_t \mid \boldsymbol{\beta}) \\ &= \pi_0(\boldsymbol{\beta}) \textstyle\prod_{\tau=1}^t \big(p(\mathbf{x}_\tau \mid \mathcal{D}_{\tau-1}) p(y_\tau \mid \boldsymbol{\beta}, \mathbf{x}_\tau)\big) \\ &\propto \pi_0(\boldsymbol{\beta}) \textstyle\prod_{\tau=1}^t p(y_\tau \mid \boldsymbol{\beta}, \mathbf{x}_\tau), \quad (4.1)\end{aligned}$$

where $p(\mathbf{x}_\tau \mid \mathcal{D}_{\tau-1})$ is the design without knowledge of the true $\boldsymbol{\beta}$ and thus dropped. Define $\pi_t(\boldsymbol{\beta}) = \pi(\boldsymbol{\beta} \mid \mathcal{D}_t)$; the updates have the form $\pi_t(\boldsymbol{\beta}) \propto \pi_{t-1}(\boldsymbol{\beta}) p(y_t \mid \boldsymbol{\beta}, \mathbf{x}_t) = \pi_{t-1}(\boldsymbol{\beta}) \phi(y_t - \mathbf{x}_t^\top \boldsymbol{\beta})$, where ϕ is the standard normal pdf.

The design subroutine. The next sensing vector, $\mathbf{x}_{t+1} \in \mathcal{X}$, is chosen to maximize the IG:

$$I(\boldsymbol{\beta}; y \mid \mathbf{x}, \pi_t) = H(y \mid \mathbf{x}, \pi_t) - \mathbb{E}\big[H(y \mid \mathbf{x}, \boldsymbol{\beta}) \mid \pi_t\big], \quad (4.2)$$

which is the difference between the entropy of the marginal distribution, $p(y \mid \mathbf{x}, \pi_t) = \int \phi(y - \mathbf{x}^\top \boldsymbol{\beta}) \pi_t(\boldsymbol{\beta}) \, \mathrm{d}\boldsymbol{\beta}$, and the expected entropy of the conditional distribution, $p(y \mid \boldsymbol{\beta}; \mathbf{x}) = \phi(y - \mathbf{x}^\top \boldsymbol{\beta})$. The latter, i.e., the conditional distribution for any realization of $\boldsymbol{\beta}$, has fixed entropy: $\log\sqrt{2\pi e}$. Meanwhile, the marginal entropy has no closed-form solutions; instead, we use numerical integration.

The numerical integration is rather straightforward, because the marginal *density* function is analytical. From now on, we will assume that $(\mathbf{x}, A, a, w_{\mathbf{x}})$ correspond to the same design (its sensing vector, its locations, its region size, and its sensing weight per coordinate, respectively). Define two new variables, $\lambda = \mu w_{\mathbf{x}} (= \mu/\sqrt{a})$ and $\gamma = \mathbf{x}^\top\boldsymbol{\beta}/\lambda$, and one new parameter $\mathbf{p} = (p_0, \ldots, p_k)^\top$ in (4.3). The goal is to change the variable of the integration for the marginal

Algorithm 4.2 Region Sensing Index-Any-k **(RSI-A)**

Require: n, μ, ϵ, and the unknown $\boldsymbol{\beta}^*$
Ensure: $\hat{S}$
1: initialize $\hat{S} = \emptyset$, $\hat{\boldsymbol{\beta}} = \mathbf{0}$
2: **for** $k = 1, 2, \ldots,$ **do**
3: infer $\pi_0(\boldsymbol{\beta}^{(k)}) \propto \prod_{\tau=1}^{t} p(y_\tau \mid \boldsymbol{\beta}^{(k)} + \hat{\boldsymbol{\beta}}, \mathbf{x}_\tau)$, $\forall \boldsymbol{\beta}^{(k)} \in \{\mu \mathbf{1}_j : j \notin \hat{S}\}$
4: call $\hat{S}^{(k)} = \text{RSI}\left(\pi_0, \epsilon, \boldsymbol{\beta}^* - \hat{\boldsymbol{\beta}}\right)$
5: aggregate $\hat{S} = \cup_{c \le k} \hat{S}^{(c)}$ and $\hat{\boldsymbol{\beta}} = \sum_{\hat{\jmath} \in \hat{S}} \hat{\mu}_{\hat{\jmath}} \mathbf{1}_{\hat{\jmath}}$.[2]
6: **end for**

density function of y to:

$$\begin{aligned} p(y \mid \mathbf{x}, \pi_t) &= \int \pi_t(\boldsymbol{\beta}) \phi(y - \mathbf{x}^\top \boldsymbol{\beta}) \, \mathrm{d}\boldsymbol{\beta} \\ &= \sum\nolimits_{c=0}^{k} p_c \, \phi(y - c\lambda) = p(y \mid \lambda, \mathbf{p}), \\ \text{where} \quad p_c &= \Pr(\gamma = c) = \sum\nolimits_{\boldsymbol{\beta}: \mathbf{x}^\top \boldsymbol{\beta} = c\lambda} \pi_t(\boldsymbol{\beta}). \end{aligned} \tag{4.3}$$

Notice, γ only has a finite number of choices: $\gamma = |A \cap S| \in \{0, \ldots, k\}$, where S is the nonzero support of $\boldsymbol{\beta}$, because both $\mathbf{x}$ and $\boldsymbol{\beta}$ are constant on their respective supports ($x_j = w_{\mathbf{x}}, \forall j \in A$ and $\beta_j = \mu, \forall j \in S$). We then numerically evaluate $H(y \mid \mathbf{x}, \pi_t) = H(y \mid \lambda, \mathbf{p})$ with the obtained (4.3).

The Bayes estimator of signal locations. We pick the k-sparse set $\hat{S}_T$ to minimize the posterior risk:

$$\min_{|\hat{S}|=k} \frac{1}{k} \mathbb{E}\big[|\hat{S} \Delta S| \mid \pi_T\big] = \frac{1}{k} \sum\nolimits_{\hat{\imath} \in \hat{S}} \mathbb{E}\big(1_{\{\beta_{\hat{\imath}}=0\}} \mid \pi_T\big), \tag{4.4}$$

where $\beta_{\hat{\imath}}$ is the $\hat{\imath}$-th element of $\boldsymbol{\beta}$. In other words, RSI picks the top k locations where the posterior marginal expectation is the largest. When $k = 1$, this is equivalent to picking $\hat{\boldsymbol{\beta}}_T = \arg\max \pi_T(\boldsymbol{\beta})$. Otherwise, (4.4) yields the smallest Bayes risk $\bar{\epsilon}(\mathcal{D}_T)$ given any collected data $\mathcal{D}_T$.

4.3.1 Accelerations

In practice, holding $\binom{n}{k}$ models in memory can be infeasible if k is large, we can instead recover the support of $\boldsymbol{\beta}$ element-wise by repeatedly applying RSI assuming $k = 1$. After the posterior distribution $\pi_t(\boldsymbol{\beta}^{(1)})$ converges to a point-mass distribution at the most-likely one-sparse model with sufficient confidence, we report its location and move on by removing the reported point

[2] In real world experiments, we additionally estimate $\hat{\mu}_{\hat{\jmath}}$ using a point measurement on the inferred signal location for better modeling.

from the search and recomputing the posterior distributions using the uniform prior, $\pi_0(\boldsymbol{\beta}^{(2)})$, on the new class, $\mathcal{S}_\mu\binom{n-1}{1}$.

We call this alternative algorithm *Region Sensing Index-Any-k* (RSI-A, Algorithm 4.2) and use it in our simulations so that the computational cost is no longer exponential in k. Notice, our analysis is for the unmodified RSI; the statistical disadvantage of RSI-A is no more than $O(k)$, multiplicatively.

When implementing RSI-A, we also avoid unnecessary numerical integration (4.2), if the region is guaranteed to have inferior IG, indicated by its $\mathbf{p}$ vector (4.3), which is easier to compute. We use the fact that $I(\gamma; y \mid \mathbf{p}, \lambda)$ with fixed $\lambda > 0$ is concave in the probability simplex $\Delta^k = \{\mathbf{p} \in [0,1]^{k+1} : \mathbf{p}^\top \mathbf{1} = 1\}$. Under $k = 1$ approximation, the region whose marginal probability $p_1 = \sum_{\mathbf{x}^\top \boldsymbol{\beta} > 0} \pi(\boldsymbol{\beta})$ is closest to 0.5 will provably have the largest IG among all regions of the same size. Thus, we find the region with the highest IG in two steps: (1) compare the p_1 value for all regions for every region size and (2) evaluate the IG of only these regions with the best p_1 values (closest to 0.5) in their region sizes.

4.4 Theoretical Analysis in 1D

The analysis is cleanest when the search space is 1d, where the regions can be any integer intervals that subset $[1, n]$. Without loss of generality (WLOG), assume n is a multiple of k and $n \geq 2k$. Our goal is to find the smallest number of measurements, T, to guarantee a small Bayes risk $\bar{\epsilon}_T = \frac{1}{k}\mathbb{E}|S\Delta\hat{S}_T| \leq \epsilon$. Table 4.2 summarizes our analysis. The sample complexity is best appreciated assuming $\mu \gg 1$, $k \ll n$, and $\epsilon = \mathcal{O}(1)$. A typical choice is $\epsilon = 1/2$, i.e., the number of measurements to guarantee that half of the signal support can be recovered on average.

4.4.1 Baseline Results

Here we provide lower bounds on sample complexity. We show that under region-sensing constraints, all passive methods require $T \geq \Omega(n)$ measurements and active methods require $T \geq \Omega(n/\mu^2 + k)$. When $\mu \gg 1$, active methods have significant potential for improvement over passive methods using region sensing, which contradicts with the view in unconstrained compressive sensing by Arias-Castro et al. [2013], Soni and Haupt [2014].

Theorem 4.2 (Limits of any passive methods using region sensing)**.** *Assume $\boldsymbol{\beta}$ has prior π_0 (uniform random on $\mathcal{S}_\mu\binom{n}{k}$). Any passive method with T noiseless region measurements on 1d must incur Bayes risk $\bar{\epsilon}_T \geq \frac{n-k}{n-1}(1 - \frac{2T}{n})$. To guarantee $\bar{\epsilon}_T \leq \epsilon$, $T \geq \frac{n}{2}(1 - \frac{n-1}{n-k}\epsilon)$ is required.*

The proof is due to model identifiability, neglecting observation noise. More details can be found in the appendix. It applies to any $\mu \geq 0$ and particularly $\mu \to \infty$.

Theorem 4.3 (Limits of any methods, [Arias-Castro et al., 2013])**.** *Assume $\boldsymbol{\beta}$ has a slightly different prior, $\tilde{\pi}_0$, that includes each location in $\mathcal{X}$ in the support of $\boldsymbol{\beta}$ independently with prob-*

Table 4.2: Conditions and conclusions for sample complexity.

Design Type	Region Sensing	Algorithm	Prior for Bayes Risk	Min T to Guarantee $\bar{\epsilon}_T = \frac{1}{k}\mathbb{E}\lvert S\Delta\hat{S}_T\rvert \le \epsilon$	Sample Complexity*
passive	yes	(any)	π_0	Theorem 4.2	$\Theta(n)$
		Point sensing	$(\mu \to \infty)$	Corollary B.2	
active	no	(any)	$\tilde{\pi}_0$	$T \ge \frac{4n}{\mu^2}(1-\epsilon)^2$ (Theorem 4.3)	$\Omega(\frac{n}{\mu^2})^\dagger$
	yes	CASS [2014]	max risk (incl. π_0)	$T \le 20\frac{n}{\mu^2}\log(\frac{8k}{\epsilon}) + 2k\log_2(\frac{n}{k})$	$\tilde{O}(\frac{n}{\mu^2}+k)^\ddagger$
		RSI (ours)	π_0	$\bar{T}_\epsilon \le 50(\frac{n}{\mu^2}+\frac{k^2}{9})\log_2(\frac{2}{\epsilon})\log(\frac{n}{\epsilon})$ (Theorem 4.4)	$\tilde{O}(\frac{n}{\mu^2}+k^2)^\ddagger$

* Assume $\epsilon = O(1)$ and $k \ll n$.
† Compared with unconstrained sensing, bisection search obeys region sensing but also requires $\Omega(\log_2(n)+k)$ measurements.
‡ $\log(n)$ terms are left out. $\bar{T}_\epsilon$ is defined differently; see Section 4.4.2 for details.

ability k/n. Any method (including active and non-region-sensing) must have $\bar{\epsilon}_T \ge 1 - \frac{\mu}{2}\sqrt{T/n}$. To guarantee $\bar{\epsilon}_T \le \epsilon$, $T \ge \frac{4n}{\mu^2}(1-\epsilon)^2$ is required.

The proof can be found under Theorem 3 of [Arias-Castro et al., 2013]. Arias-Castro et al. [2013] gave a minimax risk with similar terms by modifying $\tilde{\pi}_0$ to a *least favorable prior* on all models that are at most k-sparse. However, we only study Bayes risk for technical convenience.

When using Theorem 4.3 for reference, notice the difference between $\tilde{\pi}_0$ and π_0 that the former additionally treats the sparsity to be a random variable $\tilde{k}$ with expectation k. From concentration inequalities, $|\tilde{k}-k| \le O(\sqrt{k})$, with high probability. While $\tilde{k}$ and k are not directly comparable, Theorem 4.3 is still a useful baseline. Under region-sensing constraints, the number of measurements must also be at least $\Omega(k)$ to allow visits to most of the nonzero locations at least once, in a nontrivial draw of S where the signals are separated.

With respect to Theorem 4.2, the point sensing or any non-repeating region sensing will achieve the optimal sample complexity (up to constant factors, see Appendix A for more details). For Theorem 4.3, the CASS method published by Malloy and Nowak [2014] for active sensing with region constraints[3] acheives a nearly optimal rate in theory. Table 4.2 contains a detailed summary of the sample complexities of several algorithms, including our own.

[3] The original result in Malloy and Nowak [2014] is stronger; it considers the maximum probability of support recovery mistakes, $P(S \ne \hat{S}) \le \delta$, for any S that are k-sparse and any signals with at least μ strength.

4.4.2 Main Result

For technical convenience, we directly express our main result in terms of the expected number of measurement that are actually taken so as to realize $\bar{\epsilon}(D_{\mathcal{T}}) \leq \epsilon$ for a given threshold ϵ in an experiment. Taking $\mathcal{T} = \mathcal{T}_\epsilon$ as a random variable, the expected number of actual measurements is different from the pre-determined sampling budget that an algorithm fully consumes to guarantee a desirable averaged risk (see Section 4.4.1). However, it is a comparable alternative in Bayesian analysis, used by e.g., Kaufmann et al. [2012], Lai and Robbins [1985]. When the objective is constant $\epsilon = \mathcal{O}(1)$, our result implies a deterministic budget requirement of the same order of complexity, $T \leq \epsilon_2^{-1}\mathbb{E}\mathcal{T}_{\epsilon_2}$, where $\epsilon_2 = \frac{\epsilon}{2}$, by direct application of Markov's inequality.

Theorem 4.4 (Sample complexity of RSI)**.** *In active search for k sparse signals with strength μ in 1d physical space of size $n \geq 2k$ (WLOG, assume n is a multiple of k), given any $\epsilon > 0$ as tolerance of posterior Bayes risk,* RSI *using region sensing has bounded expected number of actual measurements,*

$$\begin{aligned}\bar{T}_\epsilon &= \mathbb{E}[\min\{\mathcal{T} : \bar{\epsilon}(D_{\mathcal{T}}) \leq \epsilon\}] \\ &\leq 50\Big(\frac{n}{\mu^2} + \frac{k^2}{9}\Big)\log_2\Big(\frac{2}{\epsilon}\Big)\log\Big(\frac{n}{\epsilon}\Big) = \tilde{O}\Big(\frac{n}{\mu^2} + k^2\Big),\end{aligned}$$

where the expectation is taken over the prior distribution and sensing outcomes.

4.4.3 Proof Sketch

The proof for Theorem 4.4 hinges on an observation that the information gain (IG) where RSI makes measurements is consistently large, before active search terminates with minimal Bayes risk. For example, the IG of any measurement in binary search with $k = 1$ and noiseless observations is always $O(\log(2))$. However, IG is harder to approximate when the observations are noisy. Therefore, we first show an intuitive lower bound for IG. Recall notations from (4.3).

Proposition 4.5. *The IG score of a region sensing design has lower bounds with respect to its design parameters $(\lambda, \mathbf{p})$, as*

$$\begin{aligned}I(\gamma; y \mid \lambda, \mathbf{p}) &\geq 2q_c\bar{q}_c\Big(2\Phi\Big(\frac{\lambda}{2}\Big) - 1\Big)^2 \\ &\geq \frac{1}{12}\min\{q_c, \bar{q}_c\}\min\{\lambda^2, 3^2\}, \quad \forall 1 \leq c \leq k,\end{aligned}$$

where $q_c = \Pr(\gamma \geq c) = \sum_{\kappa \geq c} p_\kappa$, $\bar{q}_c = 1 - q_c$, and $\Phi(u)$ is the standard normal cdf.

The proof uses Pinsker's inequality and is given in Section B in the appendix. Notice using the common choice of Jensen's inequality will give bounds in the opposite direction. To formalizes our observation that the IG is bounded:

Lemma 4.6. *WLOG, assume n is a multiple of k and $n \geq 2k$. At any step, if the current Bayes risk $\bar{\epsilon}(D) > \epsilon$, we can always find a region A of size at most $\frac{n}{k}$, such that $\lambda^2 \geq \frac{\mu^2}{a} = \frac{k\mu^2}{n}$ and $\frac{\epsilon}{2} \leq \mathbb{E}[\gamma \mid D] \leq 1 - \frac{\epsilon}{2}$ (we call this* Condition E*), which further yields*

$$I(\gamma; y \mid \lambda, \mathbf{p}) \geq I^*_\epsilon = \frac{\epsilon}{25k} \min\Big\{\frac{k^2\mu^2}{n}, 3^2\Big\}. \tag{4.5}$$

The way to find the region A that satisfies *Condition E* is given in Lemma B.5 in the appendix. The reason that Condition E is sufficient for (4.5) can be derived from Proposition 4.5 for $k = 1$ and Lemma B.6 in the appendix for $k > 1$. □

Eq (4.5) shows the minimum decrease in the model entropy in expectation after each measurement, starting from the maximum entropy of a uniform prior distribution, $k \log(n)$. However, the posterior entropy can never be negative, which implies a bound on the expected number of times that (4.5) can be applied, i.e. the expected number of measurements to reach ϵ Bayes risk is $\frac{25\log(n)}{\epsilon}(\frac{n}{\mu^2} + \frac{k^2}{9})$. Lemma D.5 in the appendix shows some additional improvements to obtain the logarithmic dependency of ϵ in Theorem 4.4.

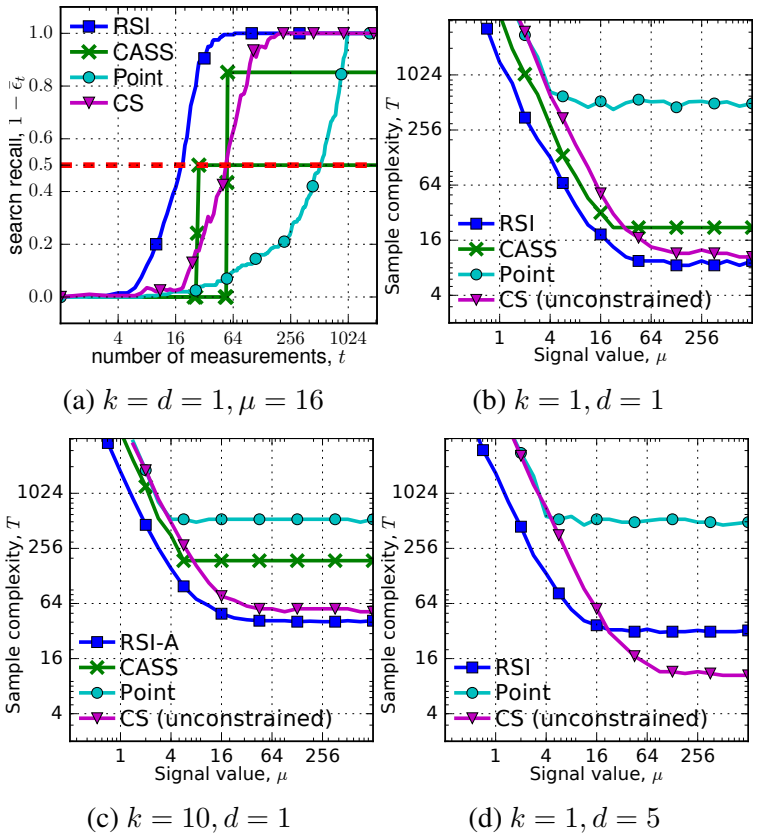
4.5 Simulation Studies

We evaluated **RSI** or its approximation **RSI-A** when $k > 1$. Other baseline algorithms include:

- **CASS** (compressive adaptive sense and search) [Malloy and Nowak, 2014]: a branch-and-bound algorithm that traverses the region hierarchy from top to down using pre-allocated budgets per level. We count each $\mathbf{x}_i$ as $\|\mathbf{x}_i\|_2^2$ region sensing measurements (rounded up to the next integer).
- **Point** sensing: a passive design that uses exhaustive point measurements on all locations.
- **CS** (compressive sensing) [Donoho, 2006]: a non-region-sensing design that draws $\mathbf{x}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and rescales $\|\mathbf{x}_t\|_2^2$ to 1. CS then solves a convex optimization problem to infer the nonzero signals, by minimizing $\sum_t \|y_t - \mathbf{x}_t^\top \boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$ s.t. $\boldsymbol{\beta} \geq 0$, where λ is chosen to produce exactly k nonzero coefficients using the Lasso regularization path.

We picked $n = 1024$ and various k (sparsity) and d (the dimension of the physical space) annotated below the plots. In the $d = 5$ case, we chose the region space to be the Cartesian product of $[4]^5$ and allowed regions from a spatial pyramid [Lazebnik et al., 2006] of granularity 4^5, 2^5, and 1^5. Each method was run with 200 repetitions to find its average performance.

Figure 4.4(a) compares the recall rates of the algorithms as they progressed in a 1d search for a single true signal of strength $\mu = 16$. RSI was the most efficient, finding the correct location in 50% of the cases with as few as $T = 20$ measurements. CASS was comparable only at the step points when all the allocated budgets were used, due to its rather rigid designs. We drew multiple curves for CASS to reflect this fact; the turning points were at $T = 28$ and 56 for $\epsilon = 0.5$ and 0.85, respectively. CS was less effective compared with CASS with equal budgets



(a) $k = d = 1, \mu = 16$ (b) $k = 1, d = 1$

(c) $k = 10, d = 1$ (d) $k = 1, d = 5$

Figure 4.4: Sensing efficiency. (a) Average search progresses as more measurements are taken. (b-d) Minimum sample size T in different SNR scenarios to guarantee $\bar{\epsilon}_T < 0.5$.

(e.g., $\|\mathbf{X}\|_F^2 = 52 > 28$ for $\epsilon = 0.5$) which agrees with the analysis in Arias-Castro et al. [2013]. Point sensing was the least efficient, using $T = n/2 = 512$ measurements, which was worse than the other methods by a factor of $\tilde{\Omega}(\mu^2)$ (ignoring logarithmic terms). Notice, due to non-identifiability, any passive designs would have equal or worse rates.

Figure 4.4(b) extends the comparison on the full spectrum of SNR, $1/4 < \mu < 1024$, showing the minimum number of measurements T to guarantee constant Bayes risk $\bar{\epsilon}_T < 0.5$. RSI led the comparison, showing a sample complexity of $\tilde{O}(n/\mu^2)$ when μ is small and $\tilde{O}(1)$ when μ is large. CASS also had a similar trend. CS ignores the region sensing constraints and was inferior to RSI. Notice CS also has a minimum sample complexity, but in order to meet the incoherence conditions for Lasso sparsistency [Candes and Tao, 2007, Raskutti et al., 2010, Wainwright, 2009], the rank of the covariance matrix of the measurements $\mathbf{X}_S^\top \mathbf{X}_S$ must be at least k. Point sensing and other passive region sensing would always require at least $\Omega(n)$ measurements regardless of μ. Figure 4.4(c-d) show similar conclusions with other choices of k and d. The number of measurements was largely unaffected by $k > 1$ if μ is low, which supports the first term of Theorem 4.4, which is $\tilde{O}(n/\mu^2)$. Comparisons between CS and RSI in high dimensions ($d > 1$) depend on how region constraints are defined. In our high-dimensional simulations, the region choices were rather limiting for RSI, giving more advantage to the unconstrained CS when μ is large.

4.6 Real World Dataset

We performed active needle search with region sensing constraints on 221 satellite image patches of 512×512 pixels each, cropped from National Agriculture Imagery Program (NAIP).[4] The objectives are all pixels with the same blue color as the objectives in the demo image (Figure 4.1) and we used a similar transformation from the RGB values for every measurement. We picked a threshold such that the number of objective pixels in every cropped environment ranges from 0 to 220, with a distribution shown in Figure 4.5. For stability in the evaluation active search performance, we reported only on the 61 environments with at least 10 objective pixels, even though the signal and noise statistics are estimated from all of the 221 images.

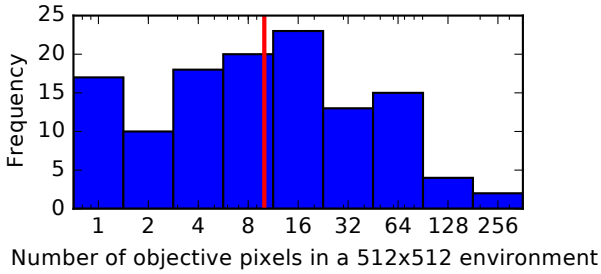


Figure 4.5: Distribution of the number of objective pixels in different experiments. For stability, we reported only on the 61 experiments with at least 10 objective pixels (right of the red bar).

Table 4.3 shows the signal and noise of region measurements at different sizes, estimated from the entire batch of 221 images. The estimated SNRs are similar to our estimates in the demo example. Intuitively, the SNR is roughly unchanged with small region sizes because of spatial correlation at smaller scales, but inversely proportional to the square-root of the region size as the regions become larger, as the pixel values are roughly independent at larger scales (see Remark 4.1). This leads to favors in choosing larger sizes, as our theoretical analysis predicts. We additionally computed the IG for every measurement at step 0, where the distribution of signal locations is uniform and thus $p(\mathbf{x}^\top \boldsymbol{\beta} > 0)$ only depends on the size of a region. The region size with the largest IG will be the first chosen region, which is also a usual choice in subsequent measurements if RSI-A decides to sequentially scan in unexplored areas.

Besides RSI-A, other algorithms under comparison include random point sensing, CS [Donoho, 2006], CASS [Malloy and Nowak, 2014], and its modifications we call CASS*. Among these methods, Point sensing and CS are passive methods, while all the others are active methods. CS further breaks region sensing constraints, for it allows arbitrary weights simultaneously assigned to all points in the search space.

The modifications to CASS are necessary because vanilla CASS relies on repeated measurements in large regions to reduce the effective noise for the final inference. However, with the image dataset, repeated measurements yield identical outcomes. It is thus impossible to allocation

[4] `https://lta.cr.usgs.gov/node/300`

Table 4.3: Signal and noise in NAIP dataset

Region size	1×1	2×2	4×4	8×8	16×16	32×32	64×64	128×128
Average in regions with needles (otherwise zero)	1.33	1.20	0.96	0.59	0.24	0.07	0.02	0.00
Standard deviation of region averages	0.16	0.14	0.12	0.11	0.09	0.07	0.06	0.05
SNR (row1 ÷ row2)	8.48	8.41	7.66	5.55	2.71	0.90	0.29	0.00
Initial IG (4.2)&(4.3)	2e-7	8e-7	3e-6	6e-6	9e-6	4e-6	2e-6	1e-14

sensing budgets correctly and any high probability conclusions via CASS are invalid. To fully represent the branch-and-bound ideas that motivate CASS, we used a modified cass* by fixing the sparsity parameter k to a larger value, in order to fully use the sensing budget, and only measured the same regions once. CASS* starts with a $4k$-partition of the entire search environment and measures each of the partitions to find the top k regions. Then, in each epoch it repeatedly splits the chosen k regions into $4k$ subregions of the next smaller size, measures the $4k$ subregions, and keeps only the top k subregions, until the subregions become single-point regions.

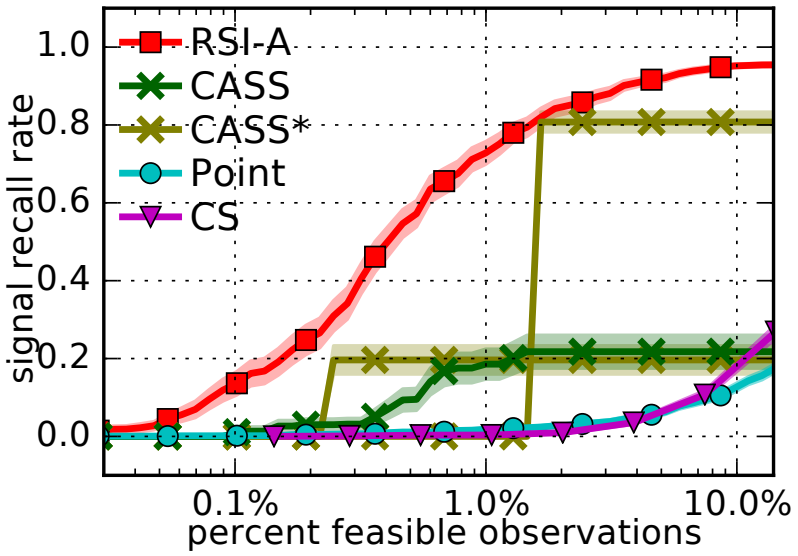


Figure 4.6: Performances on 221 NAIP image crops.

Figure 4.6 compares the recall rate of positive signal sources (there are between 10 and 220 objective pixels in every image) against the budget used in sequential executions of active search. RSI-A achieved the best performance, finding on average 60% blue pixels with as few as 1700 measurements (0.5% of the total number of feasible observations). CASS performed worse, as it could not repeat measurements in large regions to control noise or always branch into the same subregions. CASS* performance highly depended on the parameter choices and produced results only near the end of the experiment. This is due to the very nature its epoch-based approach.

CS did poorly, probably due to the fact the signals were not truly independent (a blue object can contain multiple pixels). According to Table 4.3, signal sources often appear in 4×4 clusters. While we only assumed sparsity in a manner that signal sources are spatially independent, RSI-A effectively used the additional clustering properties of signal sources, while CS ignored them.

4.7 Conclusions

Region sensing is a new setting motivated by robotic search operations where aggregate measurements on spatially contiguous regions are used and where signal sparsity is assumed, that is, the true values in the search space are either zero in most cases or significantly greater than zero at a few signal source hotspots. We model every measurement to be the average value in a rectangular region, perturbed by an additive Gaussian noise that comes from averaging of independent standard Gaussian noises in individual point values. This is equivalent to using a weighted average with uniform ℓ-2 norm and standard Gaussian noise, which we did in order to be comparable with the existing literature.

We proposed algorithms RSI and RSI-A using information-theoretic principles and demonstrated their effectiveness in choosing larger regions at early stages to improve search efficiency. RSI also comes with a guarantee on the expected number of measurements to find all signal locations, on the order of $\tilde{O}(n/\mu^2 + k^2)$, where n is the size of the search domain, μ is the signal-to-noise ratio of the true signal hotspots, k is the number of signal hotspots, and additional $\log(n)$ factors are hidden. Our complexity guarantee is near-optimal and it provides another example showing that active search approaches perform significantly better than passive approaches. In contrast with the unconstrained sensing in Arias-Castro et al. [2013], the passive alternatives under region sensing constraints can never use less than $O(n)$ measurements. RSI-A is an greedy approximation to RSI in order to improve its scalability with large n and large k. We experimented RSI-A with real satellite images to show its empirical usefulness.

In higher dimensions, the analysis may be harder, especially for passive baselines. The number of subregions generated by intersecting the measurement regions may be harder to count, unless measurement regions are restricted to grid regions in a spatial pyramid (such that any pair of regions is either nested or disjoint). On the other hand, it may be possible to improve the bound from k^2 to k, using mathematical techniques. We also want to establish frequentist analysis in the future.

The travel time of active needle search robots are not considered. Here, unlike active area search, the travel distance can be empirically efficient (e.g. in the demo search in Figure 4.2). The empirical efficiency is due to tie-breaking if the robot decides to explore new areas with equal information gains by the smallest (i.e., nearest) choices, as well as locality when larger observations lead to investigations in subregions. (In contrast, smaller observations show negative evidence and are usually ignored in the first pass of active search.) However, trajectory planning can be further optimized to improve locality with guarantees, e.g., using a space-filling curve. Finally, such trajectory may also be useful when we want to generalize the measurement model

beyond the single-pixel camera. With multi-pixel sensing models, the information may be harder to compute and I speculate a better approach based on reinforcement learning, while imitating RSI-A, which uses less information, as a starting point.

5

Conjugate Sampling

5.1 Introduction

Bayesian optimization studies the global optimization of black-box systems. It assumes that for every design of input, the system will respond with a noisy outcome, while incurring cost. The objective is to optimally design the inputs so as to obtain global optimization as quickly as possible. Since little is known about the black-box system, Bayesian optimization assumes it is a random function drawn from a given prior distribution, according to which design choices can be optimized. It is widely applied in hyperparameter tuning, a/b testing, and scientific experiments [Kandasamy et al., 2015, Snoek et al., 2012, Tesch et al., 2013]. The problem of active search is directly connected to and can often solved by modified application of Bayesian optimization.

There are a variety of solutions for Bayesian optimization, for example expected improvement [Jones et al., 1998], upper-confidence-bound [Niranjan et al., 2010], elimination rules [Even-Dar et al., 2006]. Notably, Thompson [1933] has recently rekindled research attention because of its conceptual simplicity and good empirical performance [Chapelle and Li, 2011].

At each step, Thompson sampling draws a random function from the posterior distribution of the black-box function, which comes from Bayesian inference given the collected evidence in previous steps, and then chooses the design that maximizes the expected reward, i.e., the outcome value in our case, assuming that the true function is the sampled function. When integrating out the randomness of the function draw, Thompson sampling effectively chooses a design according to the marginal probability that the chosen design is indeed the optimal design, under the posterior distribution with current information. Thanks to its flexibility, the idea of Thompson sampling can also be applied to complex design objectives, e.g., predictive entropy search

[Hernández-Lobato et al., 2014], reinforcement learning [Osband et al., 2016], and combinatorial optimization [Gopalan et al., 2014].

However, the conceptual simplicity does not directly generate computational benefits. For example, we consider linear models of the black-box function, e.g., Gaussian processes (GP) [Rasmussen and Williams, 2006] and Bayesian linear regressions (BLR). Both types of models induce multivariate normal distributions on any set of design choices; to sample from the posterior distribution, the naive approach will first perform Cholesky decomposition on the kernel matrix of feasible designs (in GP) or the covariance matrix (in BLR) and then sample in the resulting principal directions. This step has $O(n^3)$ time complexity and $O(n^2)$ space complexity, where $n \times n$ is the size of the matrix being decomposed. Caching previous decompositions and running rank-one updates as new evidence arrives can reduce the time complexity to $O(n^2)$ per iteration, but the space complexity remains the same.

Instead, we consider an iterative algorithm we call conjugate sampling that approximately draws from the posterior multivariate normal distribution a sample point, which is then used to choose the optimal design for the next step. The method is inspired by conjugate gradient descent [Hestenes and Stiefel, 1952, Sachdeva and Vishnoi, 2013], which solves a linear system with a positive-definite (PD) design matrix by minimizing its equivalent quadratic form using a very small number of matrix-vector multiplications (MVMs). Similarly, we can also approximately draw a sample point from the corresponding multivariate normal distribution by accumulating the conjugate gradient directions. Due to its iterative nature, the method only uses $O(n)$ extra memory space, in addition to $m_{\mathbf{A}}$, the storage of the precision matrix $\mathbf{A}$ itself. The total time complexity is $O(kt_{\mathbf{A}})$, where $k = O(\sqrt{\kappa_{\mathbf{A}}})$ is square-root of the condition number of $\mathbf{A}$ and $t_{\mathbf{A}}$ is the time for MVM. Conjugate sampling is beneficial when the precision matrix is sparse or structured. In the example by Flaxman et al. [2015], Wilson and Nickisch [2015] and also in our experiments where a Gaussian process (GP) in $D \geq 2$ dimensions has Kronecker-product structures, $m_{\mathbf{A}} \sim O(Dn^{\frac{2}{D}} + n) \ll O(n^2)$ and $t_{\mathbf{A}} \sim O(Dn^{\frac{D+1}{D}} + n) \ll O(n^2)$. Considering $\sqrt{\kappa_{\mathbf{A}}} \sim O(\sqrt{n})$ if proper prior conditioning is provided, the time complexity can still be reduced.

To measure the empirical performance of Bayesian optimization, we use cumulative regret [Bubeck et al., 2012], which is the cumulative difference in terms of the expected outcome between the ideal experiment using the optimal design and the chosen experiment at each step. There have been much theoretical analysis for Thompson sampling [Agrawal and Goyal, 2013a,b, Kaufmann et al., 2012, Russo and Van Roy, 2014] since the empirical findings from [Chapelle and Li, 2011]. However, a negative from Lattimore and Szepesvari [2016] suggest that Thompson sampling may not be optimal when the feasible design choices are finite and non-uniformly distributed. It may be interesting to see if another approximate sampling algorithm like conjugate sampling can have similarly good performance, even though it does not always sample from the exact posterior.

Table 5.1: Complexity of Posterior Sampling

Method	Time	Space
Thompson sampling (naive)	$O(n^3)$, fixed	$O(n^2)$, dense
Thompson sampling (online)	$O(n^2)$, fixed	$O(n^2)$, dense
Low-rank approximation	$O(k^2 t_{\mathbf{A}})$	$O(kn)$
PerturbOpti Orieux et al. [2012]	$O(kt_{\mathbf{A}})$	$O(n)$
Conjugate sampling	$O(kt_{\mathbf{A}})$	$O(n)$

We may take $k = O(\sqrt{\kappa_A})$. The space complexity subtracts the storage of $\mathbf{A}$ itself. PerturbOpti assumes the ability to sample from the prior distribution and likelihood noise.

5.2 Related Work

Our novelty lies in applying conjugate sampling to bandit problems. Conjugate sampling itself has been studied in different contexts before.

Parker and Fox [2012] has a nice overview of conjugate sampling based on the work of Schneider and Willsky [2003], with further theoretical and empirical insights. Their main algorithm is similar to ours, which is an one-line adaptation of conjugate gradient linear solver, using either the covariance matrix or the inverse covariance matrix as its coefficient matrix. One common issue with conjugate methods is the loss of orthogonality or also called *conjugacy* due to numerical errors in finite precision mathematics. A unique insight from Parker and Fox [2012] shows that conjugate sampling loses conjugacy before the corresponding conjugate gradient linear solver converges. As a result, conjugate sampling can only realize covariance matrices that approximate extreme or well-separated eigenvalue-eigenvector pairs. In their experiments, Parker and Fox [2012] showed that conjugate sampling performed well for Gaussian processes in $1d$ or Gaussian random fields based on the connectivity pattern in a $2d$ lattice grid. However, their GP experiments conducted in $1d$ environments. We extend the GP experiments in high-dimensional environments using Kronecker-decomposition of square-exponential kernels for fast matrix operations.

At its core, conjugate sampling realizes a low-rank approximation of the covariance matrix, without explicit eigendecompositions. An alternative is to compute the explicit low-rank approximations using Lanczos algorithms. Parker and Fox [2012] showed a connection between conjugate sampling and Lanczos algorithms, which allows for computation of the ℓ_2-orthogonal bases for Lanczos algorithms under the same complexity with a larger constant. However, an additional multiplicative order of k (the rank) is required in both time and space complexity to realize any of the advanced reothogonalization or spectral manipulation techniques, such as the IRAM algorithm in ARPACK for solving large-scale eigenvalue problems [Lehoucq and Sorensen, 1996]. In practice, ARPACK can be much slower.

Some other work lies between vanilla conjugate sampling and exact Lanczos. Chow and Saad [2014] focused on a few preconditioner for generic matrices. Since our matrix is Kronecker-

decomposable, we may efficiently obtain exact eigendecompositions. Simpson et al. [2008] used the covariance matrix realized by conjugate sampling to obtain the full likelihood of a Gaussian model, including its normalizing constant which involves log-determinant of the covariance matrix. This results in a novel application to sample hyper-parameters of a GP model using Metropolis-Hasting. Aune et al. [2013] implemented several Lanczos algorithms including conjugate sampling in GPU to show their computational efficiency compared to Cholesky decomposition. Li and Marlin [2016] used conjugate sampling for end-to-end Gaussian process learning. In particular, Li and Marlin [2016] use ℓ_2 orthogonal bases for irregular time series regression, while in our paper, we use $\mathbf{A}$ orthogonal bases, which may converge more quickly, for Thompson sampling.

Orieux et al. [2012] studies a slightly different setting where the inverse covariance matrix is a sum of other matrices that allow fast MVMs. As it turned out, the experiments we considered in this work appreciate the same form. Their algorithm is called Perturbation-Optimization (PerturbOpti), which first draws from the component inverse covariance matrices, e.g., the prior and likelihood noise, then solves an optimization problem involving the inverse covariance matrices. This solution separates the sampling and optimization steps, both realizable via conjugate methods. It achieved similar efficiency and appeared even more robust to numerical issues in our experiments, but it also hides away the numerical challenges when sampling from the prior or the likelihood noise.

5.3 Problem Formulation

We consider two types of Bayesian models: Gaussian processes (GP) and Bayesian linear regression (BLR); both models can have the same form of multivariate normal posterior distribution,

$$-\log p(\boldsymbol{\theta} \mid \mathbf{x}_\tau, y_\tau, \forall \tau \le t) \simeq \frac{1}{2}\boldsymbol{\theta}^\top \mathbf{A}\boldsymbol{\theta} - \boldsymbol{\theta}^\top \mathbf{b}, \text{ or equivalently, } \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{A}^{-1}\mathbf{b}, \mathbf{A}^{-1}), \quad (5.1)$$

where $\mathbf{A}$ is a symmetric and positive-definite (PD) matrix.

For BLR, we optimize the black-box function $y = \mathbf{x}^\top \boldsymbol{\theta} + \varepsilon, \forall \mathbf{x} \in \mathbb{R}^n$ s.t. $\|\mathbf{x}\|_2 \le 1$, where $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$ is the observation noise. We assume a priori that $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Theta}^{-1})$, where $\boldsymbol{\Theta}$ is a known information matrix. After observing the function values at $\mathbf{x}_1, \ldots, \mathbf{x}_t$ with outcomes $y_1, \ldots, y_t$, respectively, the posterior distribution of $\boldsymbol{\theta}$ becomes (5.1) where $\mathbf{A} = \boldsymbol{\Theta} + \frac{1}{\sigma_n^2}\sum_{\tau=1}^t \mathbf{x}_\tau \mathbf{x}_\tau^\top$ and $\mathbf{b} = \frac{1}{\sigma_n^2}\sum_{\tau=1}^t \mathbf{x}_\tau y_\tau$. Sampling-based Bayesian optimization then chooses the next observation by sampling $\tilde{\boldsymbol{\theta}} \sim p(\boldsymbol{\theta} \mid \mathbf{x}_\tau, y_\tau, \forall \tau \le t)$ and choosing $\mathbf{x}_{t+1} \in \{\mathbf{x} : \|\mathbf{x}\|_2 \le 1\}$ that maximizes the expected outcome, $\mathbf{x}_{t+1}^\top \tilde{\boldsymbol{\theta}}$.

For GP, the black-box function is modeled a priori by $f : [0,1]^D \to \mathbb{R}$ that is generated from a GP with zero mean and a given kernel function $\kappa(\cdot,\cdot)$, denoted by $f \sim \mathcal{GP}(0, \kappa(\cdot,\cdot))$, such that for any number of variables, $\mathbf{x}_1, \ldots, \mathbf{x}_n$, the outcomes $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))^\top$ always jointly assume a multivariate normal distribution with zero mean and covariance matrix $\mathbf{K}$, where $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. Let $\mathcal{N}(0, \sigma_n^2)$ be the observation noise. After observing f at $\mathbf{x}_1, \ldots, \mathbf{x}_t$ with outcomes

$y_1, \ldots, y_t$, respectively, the posterior GP is given such that for any set of points $\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*$, the function values have multivariate normal distribution with mean vector and covariance matrix,

$$\boldsymbol{\mu} = \mathbf{K}_*^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}, \quad \boldsymbol{\Sigma} = \mathbf{K}_{**} - \mathbf{K}_*^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{K}_*, \tag{5.2}$$

where the elements at (i, j) location in $\mathbf{K}_{**}, \mathbf{K}_*, \mathbf{K}$ are $\kappa(\mathbf{x}_i^*, \mathbf{x}_j^*), \kappa(\mathbf{x}_i, \mathbf{x}_j^*), \kappa(\mathbf{x}_i, \mathbf{x}_j)$, respectively [Rasmussen and Williams, 2006].

If we further assume that the set of feasible designs are fixed throughout Bayesian optimization as $\{\mathbf{x}_1^*, \ldots \mathbf{x}_n^*\}$ and let $s_\tau \in \{1, \ldots, n\}, \forall 1 \leq \tau \leq t$ be the indicator variable such that $\mathbf{x}_\tau = \mathbf{x}_{s_\tau}^*$, direct computation reveals that the posterior has the same formula as (5.1) with $\mathbf{A} = \mathbf{K}_{**}^{-1} + \frac{1}{\sigma_n^2} \sum_{\tau=1}^t \mathbf{s}_\tau \mathbf{s}_\tau^\top, \quad \mathbf{b} = \frac{1}{\sigma_n^2} \sum_{\tau=1}^t \mathbf{s}_\tau y_\tau$, where each $\mathbf{s}_\tau \in \mathbb{R}^n$ is the corresponding indicator vector of index s_τ. This formulation is most useful if the prior kernel matrix $\mathbf{K}_{**}^{-1}$ is easily invertible, e.g., when it is a Kronecker product of 1d kernel matrices [Flaxman et al., 2015].

The problem then is how to sample efficiently from (5.1) using only matrix-vector multiplication between $\mathbf{A}$ and any vector $\boldsymbol{\theta}$ in an efficient manner. Sampling-based Bayesian optimization will then choose the next design location to maximize expected reward based on the obtained sample point.

5.4 Conjugate Sampling

In this section, we present the algorithm to draw one sample point $\boldsymbol{\theta} \in \mathbb{R}^n$ from (5.1), where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a symmetric and positive-definite (PD) matrix. Denote $\|\boldsymbol{\theta}\|_{\mathbf{A}}^2 = \boldsymbol{\theta}^\top \mathbf{A} \boldsymbol{\theta}$ to be the squared A-norm of $\boldsymbol{\theta}$. We assume that the posterior mean of (5.1), $\boldsymbol{\mu} = \mathbf{A}^{-1}\mathbf{b}$, is solved separately (e.g., also using Algorithm 5.1 in a separate run); our goal is to approximate the posterior distribution with similar time and space complexity. Without loss of generality, we set $\mathbf{b} = \mathbf{0}$ to find $\tilde{\mathbf{z}} \sim \mathcal{N}(\mathbf{0}, \mathbf{A}^{-1})$ such that $\tilde{\boldsymbol{\theta}} = \boldsymbol{\mu} + \tilde{\mathbf{z}} \sim$ (5.1).

Algorithm 5.1 shows the steps to draw one sample point $\tilde{\boldsymbol{\theta}}$. It borrows the conjugate directions which are used to solve $\mathbf{A}\mathbf{x}_k = \mathbf{c}$ for random $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We analyze the case when $k_{\max} = n$. The key idea is to realize the following from standard conjugate gradient descent literature:

Theorem 5.1. *The conjugate directions, denoted in matrix form by $\mathbf{P} = (\mathbf{p}_1, \ldots, \mathbf{p}_k)$ are A-orthogonal, such that $\mathbf{P}^\top \mathbf{A} \mathbf{P} = \mathbf{D}$, where $\mathbf{D} = \operatorname{diag}(\|\mathbf{p}_1\|_{\mathbf{A}}^2, \ldots, \|\mathbf{p}_k\|_{\mathbf{A}}^2)$ is positive-definite.*

We skip the proof of Theorem 5.1 by Saad [2003] but only clarify that normally orthogonality requires computing inner products with all previous vectors. However, since the Krylov subspace $\mathcal{K}_k = \{\mathbf{c}, \mathbf{A}\mathbf{c}, \ldots, \mathbf{A}^{k-1}\mathbf{c}\}$ is a power series, we only need to orthogonalize with respect to the last 2 variables. Further $\mathbf{A}$-orthogonality is available using only 1 matrix-vector products for every variable via Algorithm 5.1. □

Our method then follows by sampling $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{D}^{-1})$ and assigning $\tilde{\mathbf{z}} = \mathbf{P}\boldsymbol{\xi} = \sum_{k'=1}^k \xi_{k'} \mathbf{p}_{k'}$, which has distribution $\mathcal{N}(\mathbf{0}, \mathbf{P}\mathbf{D}^{-1}\mathbf{P}^\top)$. There are different cases that produce different distributions.

Algorithm 5.1 Conjugate Sampling Based on $\mathbf{Ax} = \mathbf{c}$

Input MVM operator $\mathbf{A}$, vector $\mathbf{c} \sim \mathcal{N}(0, \mathbf{I})$, and scalar $\epsilon > 0$.
Let $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{p}_0 = \mathbf{r}_0 = \mathbf{c} - \mathbf{A}\mathbf{x}_0$, $\mathbf{z}_0 = \mathbf{0}$.
for $k = 0, \ldots, k_{\max} - 1$ **do**
  perform line search $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$, where $\alpha_k = \frac{\mathbf{r}_k^\top \mathbf{r}_k}{d_k^2}$, $d_k^2 = \mathbf{p}_k^\top \mathbf{A}\mathbf{p}_k$
  accumulate random variables $\mathbf{z}_{k+1} = \mathbf{z}_k + \frac{1}{d_k}\xi_k \mathbf{p}_k$, where $\xi_k \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$
  compute residual (i.e., negative gradient) $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k$
  if $\|\mathbf{r}_{k+1}\|_2 \leq \epsilon$ **then**
    break
  else
    find conjugate direction $\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$, where $\beta_k = \frac{\mathbf{r}_{k+1}^\top \mathbf{r}_{k+1}}{\mathbf{r}_k^\top \mathbf{r}_k}$
  end if
end for
output solution $\mathbf{x} = \mathbf{x}_{k+1}$, sample $\tilde{\mathbf{z}} = \mathbf{z}_{k+1}$

Case 1. Exact sampling if $\mathbf{A}$ has distinct eigenvalues and almost surely Algorithm 5.1 runs for n steps. In this case $\mathbf{P}$ has full rank and Theorem 5.1 suggests $\mathbf{A} = \mathbf{P}^{-\top}\mathbf{D}\mathbf{P}^{-1} = (\mathbf{P}\mathbf{D}^{-1}\mathbf{P}^\top)^{-1}$.

Case 2. Approximate sampling if some eigenvalues of $\mathbf{A}$ have multiplicity greater than 1 and $\mathcal{K}_k$ cannot grow to $\mathbb{R}^n$. An example is $\mathbf{A} = \mathbf{I}$, the identity matrix, which allows the gradient at any location to point directly to the origin and any gradient-based methods to converge in one step with $\mathbf{p}_1 = \mathbf{c}$. In this example, $\tilde{\mathbf{z}} = \xi_1 \mathbf{p}_1$ is a resampled variable in the same direction (or its opposite) as $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. As a result, the angular distribution of $\tilde{\mathbf{z}}$ is the same as an exact sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, but the radial density of the marginal distribution will be different: the exact random variable should allow $\|\mathbf{z}\|_2^2 \sim \chi_n^2$, but the obtained distribution has $\|\tilde{\mathbf{z}}\|_2^2 \sim \chi_1^2$. The resulting covariance matrix will still be diagonal, but the values will be underestimated n times. In general, if $\mathbf{A}$ has unique eigenvalues $0 < \lambda_1 < \cdots < \lambda_k$ with multiplicity $d_1, \ldots, d_k$, respectively, then the covariance matrix of the approximate sample will keep the same eigenvectors but change the eigenvalues from $\{\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_k}\}$ of an exact sample to $\{\frac{1}{\lambda_1 d_1}, \ldots, \frac{1}{\lambda_k d_k}\}$, respectively. To adjust for the bias in the covariance matrix, when approximation is not good in matrix 2-norm (notice good approximation may use less than n iterations if $\mathbf{A}^{-1}$ has low rank), upscaling often helps. However, it is unclear what optimal ratios are. In our simulations, we simply took the upscale ratio of $\tilde{\mathbf{z}} = \sqrt{n/(k+1)}\mathbf{z}_{t+1}$. Better approaches may require additional estimation of the approximation error [Parker and Fox, 2012].

Finally, to draw one sample point $\tilde{\mathbf{z}} = \sum_{k'=1}^{k} \xi_{k'} \mathbf{p}_{k'}$, where $\xi_k \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$, one can keep a running sum of each component without remembering the full matrix $\mathbf{P}$, which reduces the space complexity to $O(n)$, in addition to the (hopefully efficient) storage of $\mathbf{A}$.

5.5 Simulations

5.5.1 BLR Experiments

For BLR models, we randomly drew $\mathbb{R}^n \ni \boldsymbol{\theta}^* \in \mathcal{N}(\mathbf{0}, \mathbf{I})$ and used Bayesian optimization to maximize $\mathbf{x}^\top \boldsymbol{\theta}^*$, s.t. $\|\mathbf{x}\|_2 \leq 1$. Each observation has independent standard Gaussian noise, $\sigma_n = 1$. We also used $\mathcal{N}(\mathbf{0}, \mathbf{I})$ as our prior for $\boldsymbol{\theta}$ and chose designs according to $\mathbf{x}_t = \tilde{\boldsymbol{\theta}}_t / \|\tilde{\boldsymbol{\theta}}_t\|_2$, where $\tilde{\boldsymbol{\theta}}_t$ is a sample from the posterior distribution at step t. The simulation was repeated 100 times.

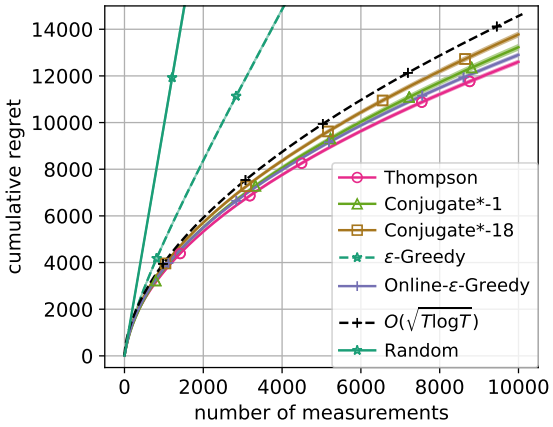


Figure 5.1: Cumulative regret against the number of function observations. BLR with $n = 100$.

Figure 5.1 shows that for $n = 100$, conjugate sampling with one gradient step ($k_{\max} = 1$, called Conjugate*-1)[1] was comparable with Thompson sampling, both in terms of computational complexity and statistical complexity. The key role that the one-step conjugate sampling plays is to scale each random vector by how much the corresponding direction has been explored, finding balance between exploration and exploitation. At step t, the coefficient matrix becomes $\mathbf{A} = \mathbf{I} + \sum_{i<t} \mathbf{x}_i \mathbf{x}_i^\top$. By resampling, the actual random vector along the same direction of an initially proposed variable $\mathbf{c}$ has $\sqrt{n}\|\mathbf{c}\|_2 / \|\mathbf{c}\|_\mathbf{A}$ expected norm, whereas a standard normal variable on$\mathbb{R}^n$ has norm $\sqrt{n}$. Exploration is thus encouraged in underexplored directions, because $\mathbf{c}$ is away from eigenvectors of large eigenvalues, realizing small $\|\mathbf{c}\|_\mathbf{A}$.

Other $k_{\max}(> 1)$ resulted in similar performance. Conjugate*-18 shows the result of an experiment with sufficiently large $k_{\max}$, where conjugate sampling realized an average of $k = 18$ MVM steps. We also included the random selection method (it yielded large regrets) and a typical theoretical rate of optimal cumulative regrets on the order of $O(\sqrt{T \log T})$, which indicates that both conjugate and Thompson sampling converged [Niranjan et al., 2010].

[1] The star variant of conjugate sampling applied the $\sqrt{n/(k+1)}$ upscaling step to overcome larger approximation errors, as we discussed under Case 2: Approximate Sampling.

To further examine the key factors of exploitation, we simplified conjugate sampling to an Online-ϵ-Greedy type algorithm. Similar to Conjugate*-1, this method also draws iid exploration variables $\mathbf{c}$. However, unlike Conjugate*-1, the rescaling is not based on the draw of $\mathbf{c}$, but instead takes a fixed form such that every random variable is uniformly scaled by $\sqrt{\frac{\mathrm{trace}(\mathbf{A}^{-1})}{n}}$ which ensures that the expected norm meets that of Conjugate*-1. The result of Online-ϵ-Greedy suggests that exploration on BLR can be rather simple, but nontrivial, as long as the strength of exploration strikes the right balance at the current time step. In contrast, ϵ-Greedy performed much worse when the scaling is fixed at 1, which is the choice in Online-ϵ-Greedy when $\mathbf{A} = \mathbf{I}$ at the initial step.

5.5.2 GP Experiments

For GP models, we generated functions on the input space $[0, 1]^D$ where $D = 1$ or 3, respectively, and with square-exponential kernel $\kappa(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\{-\frac{1}{2\ell^2}\|\mathbf{x} - \mathbf{x}'\|_2^2\}$ where $\ell = 0.3$ and $\sigma_f = 1$. The feasible designs are chosen from a Cartesian grid of points with n_d points along dimension d. For $D = 1$, we chose $n = 101$ grid points; for $D = 3$, the environment is a Cartesian product of $n = 4 \times 5 \times 6 = 120$. The function values at these n points, $\mathbf{f}^* = (f^*(\mathbf{x}_1), \ldots, f^*(\mathbf{x}_n))^\top$, form a multivariate normal distribution with zero mean and kernel matrix $\mathbf{K}_{**}$.

With proper indexing, $\mathbf{K}_{**}$ and $\mathbf{K}_{**}^{-1}$ can be decomposed by a Kronecker product of 1d matrices, $\mathbf{K}_{**}^{-1} = \otimes_d \mathbf{K}_{(d)}^{-1}$, where $\mathbf{K}_{(d)}$ is a smaller and easily invertible $n_d \times n_d$ kernel matrix built only using the dth coordinate of the input. Per discussion under (5.2), after choosing observations from the Cartesian grid points, such a prior allows for easy computation of the posterior inverse covariance matrix,

$$\mathbf{A} = \mathbf{K}_{**}^{-1} = \otimes_d \mathbf{K}_{(d)}^{-1} + \mathrm{diag}\{\sum_\tau \frac{1}{\sigma_n^2} \mathbf{s}_\tau\}, \tag{5.3}$$

which enables fast sampling approaches using conjugate sampling. Further, Kronecker decomposition allows easy computation of Cholesky decomposition $\mathbf{K}_{**} = \mathbf{L}\mathbf{L}^\top$ via $\mathbf{L} = \otimes_d \mathbf{L}_{(d)}$, where $\mathbf{K}_{(d)} = \mathbf{L}_{(d)}\mathbf{L}_{(d)}^\top$ is the Cholesky decomposition on the Kronecker components. This allows us to use preconditioned conjugate sampling, which equivalently operates on $\hat{\mathbf{A}} = \mathbf{L}^\top \mathbf{A} \mathbf{L}$ with $\hat{\mathbf{c}} = \mathbf{L}^\top \mathbf{c}$ and transforms the result back as $\mathbf{z} = \mathbf{L}\hat{\mathbf{z}}$. Notice the matrix $\mathbf{L}$ is computed only once and may be efficiently stored in its decomposed form.

The form (5.3) also allowed an alternative solution from Orieux et al. [2012], called Perturbed Optimizatoin (PerturbOpti). A simplified version of the algorithm suggests that when the inverse covariance matrix is an additive form of various components $\mathbf{A} = \sum_j \mathbf{A}_j$, instead of directly drawing from $\mathbf{z} \sim \mathcal{N}(0, \mathbf{A}^{-1})$, which may computationally be infeasible, we can equivalently draw from each component $\mathbf{c}_j \sim \mathcal{N}(0, \mathbf{A}_j^{-1})$ and solve an optimization problem:

$$\mathbf{z} \sim \mathcal{N}(0, \mathbf{A}^{-1}) \qquad \Leftrightarrow \qquad \mathbf{z} = \mathbf{A}^{-1}(\sum_j \mathbf{A}_j \mathbf{z}_j) \quad \text{s.t.} \quad \mathbf{z}_j \sim \mathcal{N}(0, \mathbf{A}_j^{-1}).$$

The resulting distribution is identical to the original distribution, because, from the right hand side, $\mathrm{Var}(\mathbf{z}) = \mathbf{A}^{-1} \sum_j (\mathbf{A}_j \mathbf{A}_j^{-1} \mathbf{A}_j) \mathbf{A}^{-1} = \mathbf{A}^{-1}$. In our example, (5.3) appreciates the given

form. Assuming that we can solve for $\mathbf{K}_{**} = \mathbf{L}\mathbf{L}^\top$ a-priori, then PerturbOpti directly samples $\mathbf{z}_0 \sim \mathcal{N}(0, \mathbf{K}_{**})$ and $z_\tau \sim \mathcal{N}(0, \sigma_n^2)$. A sample from the correct posterior distribution can be solved via $\mathbf{A}\mathbf{z} = \mathbf{K}_{**}^{-1}\mathbf{z}_0 + \sum_\tau \sigma_n^{-2} z_\tau \mathbf{e}_{s_\tau}$. Notice that when inverting $\mathbf{A}$ is infeasible due to space or time complexity, the final optimization problem typically requires conjugate gradient descent to accelerate. This solution has the same order of time and space complexity as conjugate sampling and can be more robust against numerical instabilities, if the drawing from the prior and noise model are available.

One-Dimensional GP

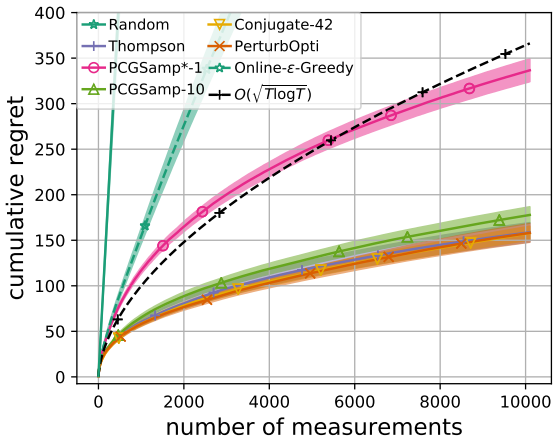


Figure 5.2: Cumulative regret with respect to the number of function observations. GP with square exponential kernel with $\ell = 0.3$ on $n = 101$ uniformly spaced grid points in $[0, 1]$. Methods below the dashed dark line had comparable regrets.

With $n = 101$ gird points on $[0, 1]$ interval with $\ell = 0.3$, the prior kernel matrix is nearly singular. For numerical stability, we added a diagonal jitter with 10^{-6} magnitude. Figure 5.2 shows the cumulative regret of various bandit solutions. The baseline Thompson sampling had an averaged total regret of 161.2 after $10\,100$ steps. Using an equivalent sampling approach, PerturbOpti performed nearly identical. Conjugate sampling, both with and without preconditioning, performed similarly to Thompson sampling, using any sufficiently large $k_{\max}$. When preconditioned by the prior $\mathbf{L}$ (such that the initial coefficient matrix becomes an identity matrix), the PCGsamp-10 algorithm used on average $k = 10$ MVM iterations to converge per decision step, before achieving the stopping criterion of $\epsilon = 10^{-5}$. Simultaneously, the realized covariance matrix of the random variable from the conjugate vectors could approximate 98% of the actual posterior covariance matrix, measured by the 2-norm of the difference between both matrices. Without preconditioning, the Conjugate-42 algorithm needed $k = 42$ MVM steps on average to meet the same stopping criterion and was able to realize the posterior covariance matrix with even smaller error. Notice, similar to Parker and Fox [2012], we used matrix-2-norm as our measure of matrix approximation error. This allowed us to approximate the matrix with fewer iterations $(k < n)$ when the covariance matrix is sufficiently low-rank.

To further study the sensitivity of exploration, we repeated the experiment with reduced $k_{\max}$. With PCGSamp*-1, we only took an iid random variable and used conjugate sampling to find the proper scale along the realized direction. This achieved relatively small regret, because the average cumulative regret is smaller than the dashed line, which indicates a typical theoretical regret bound of $O(\sqrt{T \log T})$. Without preconditioning, convergence requires more iterations. We also implemented Online-ϵ-greedy that ignores the smoothness of a GP function. It did not converge as well.

High-Dimensional Kronecker-Decomposable GP

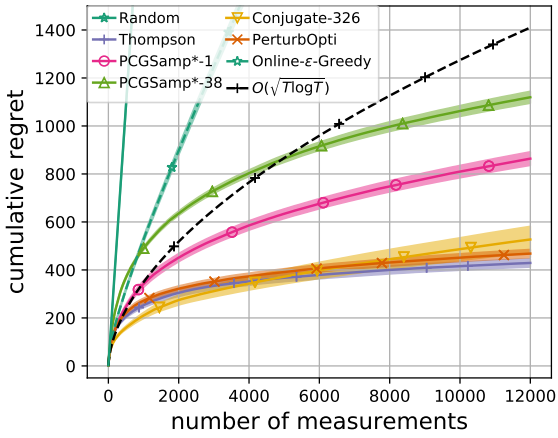


Figure 5.3: Cumulative regret with respect to the number of function observations. GP with square exponential kernel with $\ell = 0.3$ on $n = 3 \times 4 \times 5$ uniformly spaced Cartesian grid points in $[0, 1]^3$. Methods below the dashed dark line had comparable regrets.

In higher dimensions, the square-exponential kernel matrix from Cartesian grid points can be Kronecker-decomposed. As a result, MVM operation with prior $\bar{\mathbf{A}}$ becomes much more feasible, particularly in terms of space complexity. It is thus desirable to perform conjugate sampling for bandits in high-dimensional GPs. Our experiment was in $3d$ with $n = 3 \times 4 \times 5 = 120$ Cartesian grid points. We took kernel lengthscale $\ell = 0.3$. Contrary to the one-dimensional GP, the resulting kernel matrix is not singular, as the grid points are sparser. No diagonal jitter was introduced, since they also break Kronecker-decomposability.

Both Thompson sampling and PerturbOpti [Orieux et al., 2012] had smallest cumulative regrets. This shows that PerturbOpti produces robust sample points comparable to direct sampling via (online) Cholesky factorization used in Thompson sampling. In fact, we used dense matrix solvers because the scale of the problem is rather small. Even so, PerturbOpti used only half the time that Thompson sampling required, probably due to our toolkit choices. Thompson sampling was implemented with choldate package[2] whereas the optimization step in PerturbOpti was

[2] `https://github.com/jcrudy/choldate`

directly solved by a standard linear algebra package. However, on the other hand, PerturbOpti separates the subproblem of sampling and optimization, which limits its ability for extensions.

Conjugate sampling also showed signs of convergence, despite larger constants and variance. Conjugate-326 is the result of vanilla conjugate sampling without preconditioning. To reach a stopping criterion of less than $\epsilon = 10^{-5}$, Conjugate-326 used $k = 326$ MVM iterations on average. Even though $k > n$, the realized covariance matrix is still closing the gap of approximation $\mathbf{P}\mathbf{D}^{-1}\mathbf{P}^\top \approx \mathbf{A}^{-1}$ in matrix 2-norm, similar to the observations in Parker and Fox [2012]. However, the quality of final approximation is rather poor, leaving a gap of

$$\frac{\|\mathbf{P}\mathbf{D}^{-1}\mathbf{P}^\top - \mathbf{A}^{-1}\|_2}{\|\mathbf{A}^{-1}\|_2} \approx 0.7,$$

at most times during the experiment. The realized covariance matrix also has a smaller trace. One plausible explanation is that a Kronecker-product matrix has a rather concentrated eigenvalue distribution compared to kernel matrix in $1d$. The spectral distribution of Kronecker-product is more similar to an identity matrix than a square-exponential kernel matrix on dense grid points: the latter has only few large eigenvalues and is easy to approximate in matrix 2-norms. Regardless, using vanilla conjugate sampling, our experiment showed a seemingly no-regret convergence close to Thompson sampling (even though with larger variance).

Using $\mathbf{L}\mathbf{L}^\top$ from the prior as our preconditioner, PCGSamp also showed no-regret convergence below the dashed line. PCGSamp*-38 used fewer iterations and realized a covariance matrix that had similar (poor) estimation quality as vanilla conjugate sampling. Since the approximation has errors, we experimented with different choices of $\mathbf{c}$ in hope that the sampled variable behaves similarly in expectation. We showed the result when $\hat{\mathbf{c}} \sim \mathcal{N}(0, \mathbf{K}_{**})$ based on the prior distribution. Because samples are generated in the Krylov subspace of matrix powers. It is unclear to us why this choice empirically performed better than the alternatives.

Finally, PCGSamp*-1 realized (preconditioned) conjugate sampling with $k_{\max} = 1$ and iid initial $\hat{\mathbf{c}}$. Despite different behavior of conjugate sampling with Kronecker-product matrices, PCGSamp*-1 performed consistently well relative to the baseline Thompson/PerturbOpti sampling, showing signs of no-regret convergence under the dashed line of theoretical rates. The experiment with PCGSamp*-1 resonates with our assumption that exact Thompson sampling may neither be necessary nor optimal. In fact, when GP is viewed as a linear system with discrete pool of designs, Lattimore and Szepesvari [2016] showed that Thompson sampling can be arbitrary suboptimal, because exploration and exploitation are fundamentally contradicting with each other and the optimal solution may only be found via full system optimization.

5.6 Conclusions

We showed conjugate gradient sampling as a cheap iterative alternative to Thompson sampling for multi-armed bandits and Bayesian optimization. Based on linear algebra, aggregating the conjugate vectors in a conjugate gradient linear solver produces a multivariate normal random

variable, whose covariance matrix approximates the inverse of the coefficient matrix. The solution only uses Matrix-Vector Multiplications (MVMs) involving the coefficient matrix, which is very useful in Bayesian settings where the coefficient matrix (i.e., the desired inverse covariance matrix) has simple forms, e.g., the sum of a sparse GRF prior matrix (or a decomposable GP prior matrix) and other low-rank or diagonal likelihood matrices. Further, as an iterative sampler, experiments showed that the realized sample covariance matrix quickly converged to the true covariance matrix in 2-norm, with superlinear speed, when the true covariance matrix had low ranks. On the other hand, conjugate sampling may suffer greater numerical stabilities than conjugate gradient descent when the eigenvalues of the covariance matrix form dense clusters.

When applied to multi-armed bandits or Bayesian optimization, conjugate sampling performed comparably to Thompson sampling, while using less orders of time and space complexity when cheap MVMs are available. Experiments were conducted both when conjugate sampling yielded a correct sample (GP in $1d$) and when it could only produce approximate samples (BLR and GP in $3d$). In the latter case, the sampled variable usually needs additional scaling to maintain the same level of exploration. Beyond direct application of conjugate sampling, we also experimented with reduced-iteration sampling, e.g. PCGSamp*-1 that uses only one MVM iteration to properly scale an iid exploration variable (in preconditioned spaces) and Online-ϵ-Greedy which further simplifies to truly iid sampling. These experiments may be a starting point to help us understand what properties are more important in exploration and how to realize them efficiently. In a similar line, even though exact Thompson sampling is often preferable in many cases, Lattimore and Szepesvari [2016] showed that Thompson sampling may not yield the minimal regret in linear environments when the decision pool is finite; the true optimal solution requires system optimization that also considers the constraints.

In retrospect, Orieux et al. [2012] provided an alternative Perturbation-Optimization (PerturbOpti) solution that directly solves for the posterior sampling problems in our experiments. PerturbOpti separates the sampling and optimization problems, which empirically improves numerical stability. It may be a desirable algorithm to also accelerate Bayesian optimization (we are unfamiliar with such use cases prior to this work). However, it should be clear that PerturbOpti requires the ability to directly sample from all sub-components of the posterior model and uses conjugate gradient methods to truly have their advantage. It is also more difficult to extend PerturbOpti due to its modular design.

On the other hand, our conjugate sampling is designed with different principles than Orieux et al. [2012]. We directly draw samples based on implicit low-rank approximations of the covariance matrix itself and iteratively produce results. We may extend conjugate sampling to construct other uncertainty measures for exploration. For example, we may use conjugate vectors to approximate V-optimality for GRFs (Section 2.4), so that we can find better alternatives to UCB in large graphs. Garnett et al. [2012], Wang et al. [2013] used the covariance matrix for other purposes as well. We view conjugate sampling as a cheap alternative to explicit low-rank decomposition, which can be realized by IRAM in ARPACK. Expressing full decomposition requires additional $O(k)$ multiplicative time and space complexity to allow additional reorthogonalization or restarting. It is a future direction to find other quantities that may also be well-approximated without the additional k factor.

The numerical error from finite precision mathematics showed several influences in our experiments. When the matrix is near identity with many clustered eigenvalues, conjugate sampling loses conjugacy very quickly. Further, only one eigenvector can be realized in a cluster of eigenvalues, leaving the others exposed as approximation error in matrix 2-norm. Ideally, it is desirable to use preconditioners to spread out the eigenvalues. When preconditioning is infeasible, we must find proper distributions for the initial variable $\mathbf{c}$, such that the expected covariance matrix after integrating out $\mathbf{c}$ may match the objective matrix. We may also upscale the random variable for the same purpose.

The optimal initialization of $\mathbf{c}$ may be hard to find when conjugate sampling does not realize the desired covariance matrix, e.g., in our experiment with GP in $3d$. In a simpler case, e.g., for BLR with iid prior, we followed suggestions from Parker and Fox [2012] to take $\mathbf{c} = (\pm 1, \ldots, \pm 1)^\top$ in order to reduce variance. The choice with GP in $3d$ is empirically taken. As the Krylov subspace is a power space, it is unclear how to propose initial distributions to be closest to the true distribution that we aim to sample. For example, the same variable in Orieux et al. [2012] may not yield the best solution. Our future work includes better understanding of the role of $\mathbf{c}$ in the final covariance estimation when the covariance matrix has large and clustered eigenvalues that cause numerical instabilities.

6
Conclusions

We study in the general topic of active search, where an algorithm explores in an unknown environment, collects and learns from human or environmental interactions, and ultimately search for all positive examples as quickly as possible under limited interaction budgets. It has applications in environmental monitoring, social science, and search and rescue.

There have been many paradigms that may adapt to active search problems, e.g., designs of experiments, Bayesian optimization, and multi-armed bandits. However, these studies usually focus on low-dimensional feature space, where each action can only apply to a single point and each reward will be assigned to the same point. On the other hand, real-world applications may require active search on graphs, with rewards defined on the patterns in group of points, and queries conducted on the aggregate statistics in a region containing multiple points.

In active search on graphs, each node bears a reward, which is unknown at first but can be noisily observed upon query. The aim is to accumulate as large a sum of rewards from the queried nodes as possible under limited budgets. We assume that the graph is known and the node rewards vary smoothly along the graph. The node values are relaxed to real values and modeled with a Gaussian random field prior, which naturally extends label propagation solutions from semi-supervised learning.

Popular GP-UCB-style algorithms use the marginal standard deviation as their exploration criterion, leading to the undesirable tendency of selecting peripheral nodes on a graph, e.g., leaf nodes. Instead, we propose to use variance minimization in GRF posteriors. We analyzed V-optimality from Ji and Han [2012] and Σ-optimality from Garnett et al. [2012] and proposed a novel use of Σ-optimality in active learning and bandit exploration, despite Σ-optimality being originally designed for active surveying. We call our method GP-SOPT, which modifies GP-UCB by using Σ-optimality as its exploration criterion. We also made several theoretical

contributions, including the global optimality of greedy application of V/Σ-optimality and regret analysis for a thresholded variant of GP-SOPT. Empirical evaluations justified Σ-optimality in active learning, active surveying, and active search.

Active area search is a new problem, wherein we wish to identify regions in a continuous space with large average function value. In comparison to typical active learning objectives, this setting is somewhat unusual in that we cannot observe the labels directly. Instead we must infer the labels from observations of a continuous ancillary function. Further, we were able to generalize the problem to active pointillist pattern search, where the goal is to discover specific local patterns exhibited in the regions.

Our approach is to model the function using a Gaussian process and use Bayesian quadrature to infer its average value on the regions of interest. With this setup, we were able to derive a simple expected reward maximization strategy for the active area search problem. Our solution extends to the more complex active pointillist pattern search problem, if the anticipated region pattern can be described as a functional probit model. For the more general region patterns, we relied on Monte-Carlo sampling which empirically showed good promises with moderate number of samples. We used active search for three applications: water quality measurements in a pond, election results prediction, and vortex detection in a fluid flow experiment.

Active needle search for sparse signals with region sensing is motivated by applications where aerial robots are used to detect gas leaks, radiation sources, and human survivors of disasters. Aerial robots are able to sense a region of space whose area depends on their operating altitude. The question we ask is how such a robot can dynamically trade off the ability to make noisier observations of larger regions of space against making higher-fidelity measurements of smaller regions.

We make the simplification that the robots carry a single-pixel camera that records the average value inside a rectangular region of space, corrupted by independent observation noise. We call this observation scheme a *region sensing constraint*, under which, the spacial information inside each region is unattainable. However, efficient solutions can still be found, e.g., using binary search when the observations are noiseless. We use similar principles for noisy binary search and propose an algorithm called Region Sensing Index (RSI). Further, we theoretically show that RSI performs near-optimally in $1d$ search domains and fundamentally faster than passive sensing. The number of measurements is comparable to compressive sensing, despite compressive sensing being incompatible with region sensing constraints. Empirical results on satellite images also showed the efficiency using RSI.

Finally, because many active search methods involve GPs or GRFs, we explored the ability to accelerate Bayesian optimization designs in these domains. We were inspired by the re-emerging popularity of Thompson sampling for approximately optimal designs. However, Thompson sampling does not directly yield computational benefits, despite their conceptual simplicity in that only a single point needs to be drawn from the posterior distribution.

Instead, we considered an iterative algorithm we call conjugate sampling that approximately draws from the posterior multivariate normal distribution in GPs or GRFs. We take advantage of the sparsity in their inverse covariance matrices, which are an additive sum of low-rank matrices

and Kronecker-decomposable matrices for GPs or sparse adjacency matrices for GRFs. In such way, fast approximate samples can be generated with little time and space complexity, while Bayesian optimization still efficiently converges. We work to apply conjugate sampling in large-scale Bayesian optimization problems and make novel discoveries.

6.1 Future Work

6.1.1 Active Search on Graphs

While our current solution is able to find efficient queries in moderately-sized graphs (e.g., using only 10 data points to realize classification on UCI hand-written dataset of a few thousand points/graph nodes), the true power of active search should enable graphs with millions of nodes, e.g. recommending products from the entire catalog book or identifying new classes in ImageNet dataset. Classical designs require the full covariance matrix, which is infeasible to obtain. We may use conjugate sampling to realize D or V-optimality in large graphs, which allows for approximate applications of GP-UCB or GP-VOPT (Section 2.5). Just as spectral clustering via low-rank approximation is a good way to inspect node properties using only graph connectivity, using approximate GP-UCB or GP-VOPT may also yield novel discoveries on large graphs.

We used Gaussian random fields as a generative model for the node label distribution on a graph, which is related to the unnormalized graph Laplacian. However, true node labels are binary and their distribution is infeasible to apply inference with. While allowing feasible posterior inference, the continuous relaxation via GRFs also brings errors, particularly in variance estimation. Nodes on the boundary of the graph with low degrees (e.g. leaf nodes) tend to have both large variance and large errors in their variance estimate. One plausible explanation for the improved performance by Σ-optimality is that it is more robust to this kind of modeling errors. Alternatively, we may consider using the more common V-optimality with normalized graph Laplacian matrices. In our preliminary experiments, normalizing graph Laplacian matrices generated slight better performance for V-optimality, but could not fill the performance gap between Σ and V-optimality.

The optimal generative model should be faithful to the true node label distributions. In real world graphs, edges have features. It is thus possible to learn a linear combination of features to best connect the resulting graph to the node labels. Seeing the weights in the linear combination as hyperparameters, a full Bayesian solution should also optimize for the hyperparameters to maximize the full likelihood of the observed node labels. Again, for large graphs, the normalizing constant, which includes log-determinant of the (augmented) graph Laplacian is hard to compute. Luckily, we may again turn to approximations via conjugate sampling methods [Simpson et al., 2008].

Finally, D, V, and Σ-optimality use (or approximate) functions based on the spectrum of the GRF posterior covariance matrix. We may explore other spectral functions as alternatives for better exploration in active search.

6.1.2 Scientific Applications

Scientific experiments is a classical area that requires optimal designs of experiments. Traditional, fMRI experiments use a fixed sequence of stimuli; it may be challenging for humans to quickly interpret necessary connections between stimuli and brain patterns, especially in real-time environments. Automatic selection of stimuli, on the other hand, may greatly improve efficiency in many fMRI or similar experiments. In Lorenz et al. [2016], Bayesian optimization was used to search for stimuli that evoke a desired target brain state in fMRI studies. This is a search-like objective in high-dimensional environments and it presents opportunities for other active search solutions, e.g., with graph formulations.

6.1.3 Robotic Applications

Both active area search (Chapter 3) and active needle search (Chapter 4) study robotic applications. We may extend the discussions to similar problems. For example, in robotic surgery, we may conduct active search for areas of tumors, blood vessels, etc. The location of the robotic arms cannot be accurately sensed and the organ environment may not be fully modeled. Instead, one may use stiffness feedback from interaction with the organs to identify landmarks and the arm position. Further, it is even possible to sense tumors directly by stiffness when the objective is to cut them. However, how can we efficiently find the stiff regions?

Ayvali et al. [2016], Nichols and Okamura [2015] discussed approaches based on Bayesian optimization via expected improvements or UCB. These are good algorithms that explains exploration/exploitation tradeoff, but not designed for the true objective that is to search for positive regions. It is intuitive to apply active area search directly in these applications, in a similar fashion to the environmental monitoring experiment.

On the other hand, we may extend active needle search with aerial robots using multi-pixel cameras. Here, direct calculation of information gain using the joint output distribution may be infeasible. Instead, we may fix the traveling path as a space-covering curve (Figure 6.1) and at every step only decide whether the robot should travel to next area, a subregion, or a super-region. The space-covering curve is a fractal curve, which recurses itself at shorter scales. As a result, traveling along the curve at various scales eventually surveys the entire space.

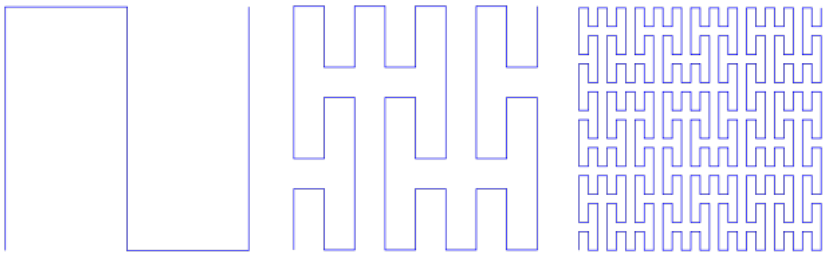


Figure 6.1: Space-covering curve as a fixed travel path for active needle search. Subsampling at fixed intervals realizes the same patterns at a larger scale. (From Wikipedia by Tó campos1.)

Now that we model active needle search with aerial robots as a sequential decision process along the space-covering curve, we may also apply reinforcement learning (RL) to find the optimal solution, using recurrent neural networks to memorize the history of previous measurements as the states. The RL solution relies on many repeated experiments, which are realizable using satellite images. The RL approach may also be initialized by imitating RSI-A.

For both active area search and active needle search, we assume that the travel time is much smaller than the time necessary to take an measurement. This is reasonable for environmental monitoring and gas leak detection because of hysteresis in the sensors. The sensor must remain stationary for a while to collect an accurate measurement. In the case of our actual data, the sensor moves continuously. It brings up two issues: 1. can we correct for the hysteresis in the dataset and 2. in cases where the assumption does not hold, how might we correctly choose experiments when the travel cost is significant? For active needle search, using the space-covering curve may become a solution to minimize traveling distance. For active area search, a simple solution may not be found. Instead, one could include travel cost in the utility function and apply greedy optimization for the expected reward per unit of travel distance, planning for a few steps at a time. However, it is not known whether such a strategy may be empirically or theoretically optimal.

6.1.4 Unified Models for Region Queries and Region Rewards

While we discussed using point sensing for region rewards and using region sensing for sparse point rewards, it may be tempting to combine both region sensing and region rewards, assuming that the regions for queries are different from regions for rewards. An immediate application is in recommender systems if one has to pay to reveal a customer's entire history, while the goal is to maximize sale of a set of products. Moreover, unified models can also be desirable for model robotic applications.

One general approach may involve using a bipartite graph formulation where there are two types of nodes: queryable nodes and reward nodes, as well as edges showing their connections. However, a bipartite graph is different from the general graph that we discussed for active search, in that the subgraph structure, which includes a source node and its closest nodes up to a small degree, may reveal important information about the property of the node. Recently, Defferrard et al. [2016], Kipf and Welling [2016] used neural networks to encode the subgraph in the neighborhood of every node as side information for the inference of node properties. The neural encoding papers usually encode neighborhood using random walk sampling. We may also use more general sampling-based approaches like Thompson sampling for the design of optimal queries.

6.1.5 Active Search in Computation Graph Environment

Besides active search in physical environments, search has a special meaning in computation graphs [Hart et al., 1968]. Search in physical environments is connected to search in computation

graphs. For example, Monte-Carlo Tree Search (MCTS) can be viewed as a sampling-based bandit solution to find the optimal trajectory in a computation tree. Using neural networks as prior, MCTS allowed for the design of AlphaGo [Silver et al., 2016] which advanced the state of artificial intelligence in complex game plays. Recent interests in computation graphs and machine learning also allowed for successes in AI Texas Hold'em [Brown and Sandholm, 2017], combinatorial optimization [Vinyals et al., 2015], etc. Our solutions in active search realize relaxed versions of binary classification problems. It may be interesting to explore active search in computation graphs or other complex binary systems.

A

Active Search on Graphs Proofs

A.1 Submodularity of Σ-Optimality

Our results apply to any step in GRF posterior covariance matrix (2.7) and extend to GPs whose inverse covariance matrix meets Proposition A.1.

Proposition A.1. *L satisfies the following.* [1]

#	*Textual description*	*Mathematical expression*
pA.1.1	*L has proper signs.*	$l_{ij} \geq 0$ *if* $i = j$ *and* $l_{ij} \leq 0$ *if* $i \neq j$.
pA.1.2	*L is undirected and connected.*	$l_{ij} = l_{ji} \forall i, j$ *and* $\sum_{j \neq i}(-l_{ij}) > 0$.
pA.1.3	*Node degree no less than number of edges.*	$l_{ii} \geq \sum_{j \neq i}(-l_{ij}) = \sum_{j \neq i}(-l_{ji}) > 0, \forall i$.
pA.1.4	*L is nonsingular and positive-definite.*	$\exists i : \; l_{ii} > \sum_{j \neq i}(-l_{ij}) = \sum_{j \neq i}(-l_{ji}) > 0$.

□

Although the properties of V-optimality fall into the more general class of *spectral functions* (Friedland and Gaubert [2011]), we have seen no proof of either the suppressor-free condition or the submodularity of Σ-optimality on GRFs.

Lemma A.2. *For any L satisfying (pA.1.1-4), $L^{-1} \geq 0$ entry-wise.*[2]

Proof. Suppose $L = D - W = D(I - D^{-1}W)$, with $D = \operatorname{diag} L$. According to (pA.1.1),

[1] Property pA.1.4 holds after the first query is done or when the regularizor $\delta > 0$ in (2.3).

[2] In the following, for any vector or matrix A, $A \geq 0$ always stands for A being (entry-wise) nonnegative.

$D \geq 0$, $W \geq 0$ and $D^{-1}W \geq 0$. Furthermore, by (pA.1.3),

$$0 \leq D^{-1}W \leq \big(\frac{w_{ij}}{\sum_k w_{ik}}\big)_{i,j=1}^{N},$$

and so the matrix norm $\|D^{-1}W\|_\infty \leq 1$. Thus, any eigenvalue λ_k and its corresponding eigenvector $\mathbf{v}_k$ of $D^{-1}W$ needs to satisfy $|\lambda_k|\|\mathbf{v}_k\|_\infty = \|D^{-1}W\mathbf{v}_k\|_\infty \leq \|\mathbf{v}_k\|_\infty$, i.e. $|\lambda_k| \leq 1, \forall k = 1, ..., N$.

When L is nonsingular, $(I - D^{-1}W)$ is invertible, i.e., has no zero eigenvalue. Hence, $|\lambda_k| < 1, \forall k = 1, ..., N$ and $\lim_{n\to\infty}(D^{-1}W)^n = 0$. The latter yeilds the convergence of Taylor expansion,

$$L^{-1} = [I + \textstyle\sum_{r=1}^{\infty}(D^{-1}W)^r]D^{-1}.$$

It suffices to observe that every term on the right hand side (RHS) is nonnegative. □

Corollary A.3. *The* GRF *prediction operator* $L_{\mathbf{u}}^{-1}L_{ul}$ *maps* $\mathbf{y}_S \in [0,1]^{|S|}$ *to* $\hat{\mathbf{y}}_{\mathbf{u}} = -L_{\mathbf{u}}^{-1}L_{ul}\mathbf{y}_S \in [0,1]^{|\mathbf{u}|}$. *When* L *is singular, the mapping is onto.*

Proof. For $\mathbf{y}_S = \mathbf{1}$, $\big(L_{\mathbf{u}}, L_{ul}\big) \cdot \mathbf{1} \geq 0$ and $L_{\mathbf{u}}^{-1} \geq 0$ imply $\big(I, L_{\mathbf{u}}^{-1}L_{ul}\big) \cdot \mathbf{1} \geq 0$, i.e. $\mathbf{1} \geq -L_{\mathbf{u}}^{-1}L_{ul}\mathbf{1} = \hat{\mathbf{y}}_{\mathbf{u}}$.

As both $L_{\mathbf{u}} \geq 0$ and $-L_{ul} \geq 0$, we have $\mathbf{y}_S \geq 0 \Rightarrow \hat{\mathbf{y}}_{\mathbf{u}} \geq 0$ and $\mathbf{y}_S \geq \mathbf{y}'_S \Rightarrow \hat{\mathbf{y}}_{\mathbf{u}} \geq \hat{\mathbf{y}}'_{\mathbf{u}}$. □

Lemma A.4. *Suppose* $L = \begin{pmatrix} L_{11} & L_{12} \\ S_{21} & L_{22} \end{pmatrix}$, *then* $L^{-1} - \begin{pmatrix} L_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} \geq 0$ *and is positive-semidefinite.*

Proof. When L is nonsingular, by the block matrix inversion theorem,

$$L^{-1} - \begin{pmatrix} L_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} L_{11}^{-1}(-L_{12}) \\ I \end{pmatrix} (L_{22} - L_{21}L_{11}^{-1}L_{12})^{-1} \big((-L_{21})L_{11}^{-1}, I\big)$$

By assumption (pA.1.4), L^{-1} is positive-definite, so is its lower right principal submatrix $(L_{22} - L_{21}L_{11}^{-1}L_{12})^{-1}$. Thus, $L^{-1} - \begin{pmatrix} L_{11} & 0 \\ 0 & 0 \end{pmatrix}$ is positive-semidefinite.

By Lemma A.2, $L^{-1} \geq 0$ and this implies that its lower right $(L_{22} - L_{21}L_{11}^{-1}L_{12})^{-1} \geq 0$. The submatrix L_{11} also satisfies (pA.1.1-4) and by Lemma 1, $L_{11}^{-1} \geq 0$. By the sign rule (pA.1.1), $(-L_{12}) = (-L_{21})^T \geq 0$. Now that every term on the right side of (A.1) is nonnegative, the left side also has to be so. □

As a corollary, the **monotonicity in** (2.21) for both $R(\cdot) = R_V(\cdot)$ or $R_\Sigma(\cdot)$ can be shown. □

Both proofs for **submodularity in** (2.22) and **Theorem 2.4** result from more careful execution of matrix inversions. We first state the key property in these executions of matrix inversions and then prove both results.

Proposition A.5. *Without loss of generality, let* $\mathbf{u} = V - S = \{1, \ldots, k\}$ *and* $v = v_k$*. Partition the matrix:*

$$L_{(V-S)} = \begin{pmatrix} L_{(V-S\cup\{v\})} & L_{(V-S\cup\{v\}),\{v\}} \\ L_{\{v\},(V-S\cup\{v\})} & L_{\{v\}} \end{pmatrix} := \begin{pmatrix} A & b \\ b^T & c \end{pmatrix}$$

By the block matrix inversion theorem,

$$\begin{pmatrix} C & d \\ d^T & e \end{pmatrix} := \begin{pmatrix} A & b \\ b^T & c \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} \frac{A^{-1}bb^TA^{-1}}{c-b^TA^{-1}b} & \frac{-A^{-1}b}{c-b^TA^{-1}b} \\ \frac{-b^TA^{-1}}{c-b^TA^{-1}b} & \frac{1}{c-b^TA^{-1}b} \end{pmatrix}.$$

□

Proof. **submodularity in** (2.22) **for** $R_\Delta(\cdot)$**.** Adopting the notations in Proposition A.5,

$$L^{-1}_{(V-S)} - L^{-1}_{(V-S-\{v\})} = \begin{pmatrix} A & b \\ b^T & c \end{pmatrix}^{-1} - \begin{pmatrix} A^{-1} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} -A^{-1}b \\ 1 \end{pmatrix} \frac{1}{c - b^TA^{-1}b} \left(-b^TA^{-1}, 1\right)$$

For V-optimality,

$$R_\Delta(S \cup \{v\}) - R_\Delta(S) = \operatorname{tr}\left(-L^{-1}_{(V-S-\{v\})} + L^{-1}_{(V-S)}\right) = \frac{((-b^T)A^{-1})(A^{-1}(-b)) + 1}{c - (-b)^TA^{-1}(-b)}.$$

As every term on the RHS has been written as nonnegative entry-wise, by taking submatrices/vectors of consistant rows/columns of A and $-b$, the values of $(-b^T)A^{-1}$ and $(-b^T)A^{-1}(-b)$ decrease.

Notice that both A and b correspond to $(V - S \cup \{v\})$. Thus, as S grows, A and b shrink in size, $R_\Delta(S \cup \{v\}) - R_\Delta(S)$ diminishes.

For Σ-optimality,

$$R_\Delta(S \cup \{v\}) - R_\Delta(S) = \mathbf{1}^T \cdot \left(-L^{-1}_{(V-S-\{v\})} + L^{-1}_{(V-S)}\right) \cdot \mathbf{1} = \frac{((-b^T) \cdot A^{-1} \cdot \mathbf{1})^2}{c - (-b)^TA^{-1}(-b)}.$$

Similar arguments hold. □

Proof. **Theorem 2.4.** Adopt the notations in Proposition A.5. Dividing the vector d by the diagonal number e yields $\forall i \neq k$:

$$\frac{\operatorname{cov}(y_i, y_k|S)}{\operatorname{Var}(y_k|S)} = \frac{(L^{-1}_{(V-S_1)})_{ik}}{(L^{-1}_{(V-S_1)})_{kk}} = \frac{1}{e} \cdot d_i = \frac{(-A^{-1}b)_i}{c - b^TA^{-1}b} \Big/ \frac{1}{c - b^TA^{-1}b} = (A^{-1}(-b))_i.$$

That $-b \geq 0$ and $A^{-1} \geq 0$ leads to $A^{-1}(-b)^T \geq \tilde{A}^{-1}(-\tilde{b}) \geq 0$ if $\tilde{A}$ and $\tilde{b}$ are subsets of consistent columns/rows (Lemma A.4), i.e.,

$$\frac{(L^{-1}_{(V-S)})_{ik}}{(L^{-1}_{(V-S)})_{kk}} \geq \frac{(L^{-1}_{(V-S\cup S_2)})_{ik}}{(L^{-1}_{(V-S\cup S_2)})_{kk}} \geq 0 \quad \forall i \neq k \notin S \cup S_2.$$

Similarly, reordering the indices, $\dfrac{(L^{-1}_{(V-S)})_{ik}}{(L^{-1}_{(V-S)})_{ii}} \geq \dfrac{(L^{-1}_{(V-S\cup S_2)})_{ik}}{(L^{-1}_{(V-S\cup S_2)})_{ii}} \geq 0$. It suffices to multiply both sides of the above. □

A.2 Active Search Regret Bound

We start by stating the following result.

Theorem A.6 (Theorem 6, Srinivas et al. [2012])**.** *Let $\delta \in (0,1)$. Assume the observation noises are uniformly bounded by σ_n and f has RKHS norm B with kernel $\bar{\mathbf{C}}$, which is equivalent to $\mathbf{f}^\top \bar{\mathbf{L}} \mathbf{f} \leq B^2$. Define $\alpha_t = \sqrt{2B^2 + 300\gamma_t \log(t/\delta)^3}$, then*

$$Pr\left(\forall t, \forall v \in V, \;\; |\mu_t(v) - f(v)| \leq \alpha_{t+1}\sigma_t(v)\right) \geq 1 - \delta.$$

We use this result to bound our instantaneous regrets.

Lemma A.7. *Conditioned on the high-probability event in Theorem A.6, the following bound holds:*

$$\forall t, \;\; r_t := f(v_t^*) - f(v_t) \leq 2\alpha_t k \sigma_{t-1}(v_t),$$

where v_t^* is the node with the t-th globally largest function value and v_t is node selected at round t.

Proof. At round t there are two possible situations. If v_t^* was picked at some earlier round, the definition of v_t^* implies that there exists some $t' < t$ such that $v_{t'}^*$ has not been picked yet. According to our selection rule, the fact that $s_t(v) \geq \sigma_t(v)$, and Theorem A.6, the following holds:

$$\begin{aligned}\mu_{t-1}(v_t) + \alpha_t s_{t-1}(v_t) &\geq \mu_{t-1}(v_{t'}^*) + \alpha_t s_{t-1}(v_{t'}^*) \\ &\geq \mu_{t-1}(v_{t'}^*) + \alpha_t \sigma_{t-1}(v_{t'}^*) \geq f(v_{t'}^*) \geq f(v_t^*).\end{aligned}$$

If v_t^* has not been picked yet, a similar argument gives

$$\mu_{t-1}(v_t) + \alpha_t s_{t-1}(v_t) \geq \mu_{t-1}(v_t^*) + \alpha_t s_{t-1}(v_t^*) \geq f(v_t^*).$$

Thus we always have

$$\begin{aligned}f(v_t^*) &\leq \mu_{t-1}(v_t) + \alpha_t s_{t-1}(v_t) \\ &\leq f(v_t) + \alpha_t \sigma_{t-1}(v - t) + \alpha_t s_{t-1}(v_t) \\ &\leq f(v_t) + 2\alpha_t k \sigma_{t-1}(v_t).\end{aligned}$$

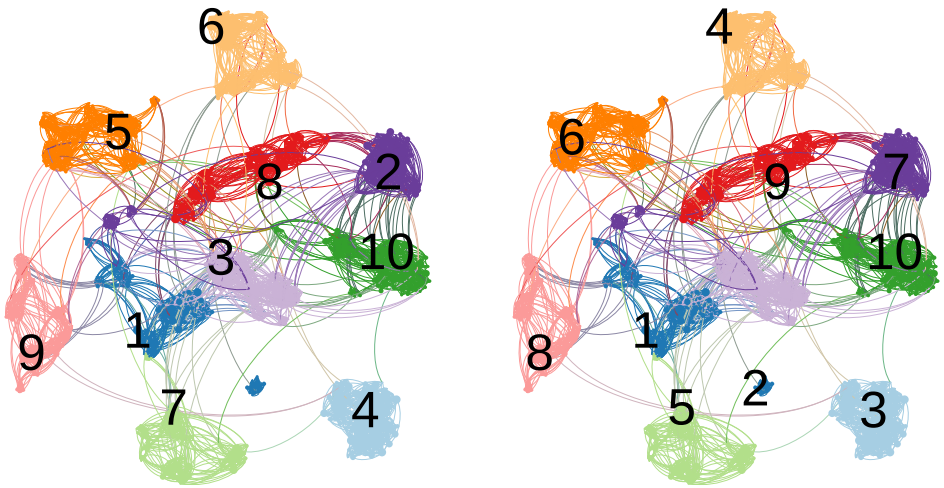
Lemma A.8 (Lemma 5.4, Srinivas et al. [2012])**.** *Let α_t be defined as in Theorem A.6 and c_1 be defined as in Theorem 2.6. Conditioned on the high probability event of Theorem A.6, the following holds:*

$$\forall T \geq 1, \;\; \sum_{t=1}^{T} r_t^2 \leq \alpha_T k^2 c_1 \mathcal{I}(\mathbf{y}_{\mathbf{v}_T}; f_{\mathbf{v}_T}) \leq \alpha_T k^2 c_1 \gamma_T.$$

Finally, Cauchy-Schwarz inequality yields $R_T \leq \sqrt{T \sum_{t=1}^{T} r_t^2} \leq k\sqrt{T c_1 \alpha_T \gamma_T}$.

A.3 Visualization of the Node Choices in Real Graphs

To gain insights of the empirical behavior of Σ- and V-optimality, it is helpful to layout the graphs on the 2D plane and visually inspect the choices of various heuristics. We use the OpenOrd toolbox [Martin et al., 2011] in the Gephi software[3] for this purpose.



(a) Σ-optimality (b) V-optimality

Figure A.1: **digits** 7-nn undirected graph. Labels show the sequence of queries. Colors suggest true (but unseen) class labels.

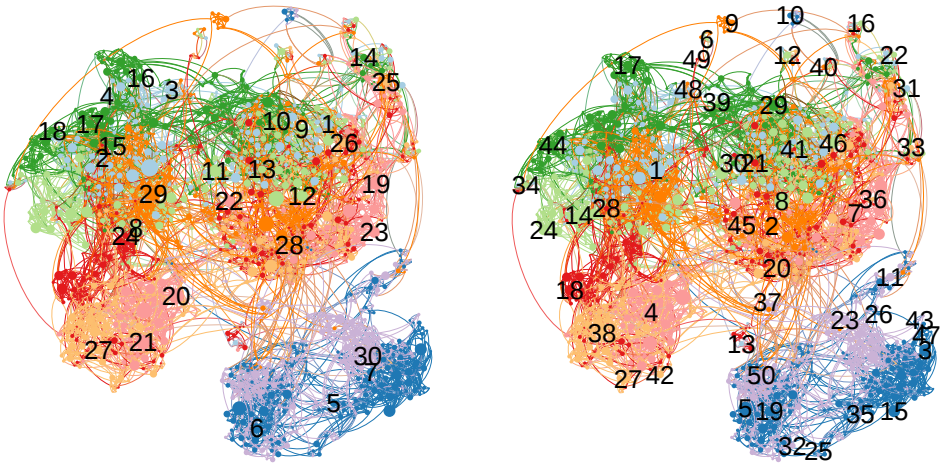
Figure A.1 shows the first few choices of the Σ- and V-optimality in the manifold embedding for **digits**. It is clear that V-optimality made a mistake taking its second query whereas Σ-optimality is able to better balance cluster size over uncertainty to avoid exploiting valueless small clusters.

Figure A.2 contrasts the first few choices of the Σ- and V-optimality in the manifold embedding for **ISOLETe**. The outside of the 2D layout are peripheral points. Notice that the queries for Σ-optimality stays in the central parts of the graph whereas the V-optimality goes after outliers (especially query 9, 10, 16).

An easier network graph is the **Cora** citation graph. In Figure A.3, sparse cuts are prevalent and they highly correlates to class margins. Σ-optimality look for clusters at the right size whereas the clusters queried by V-optimality are too small.

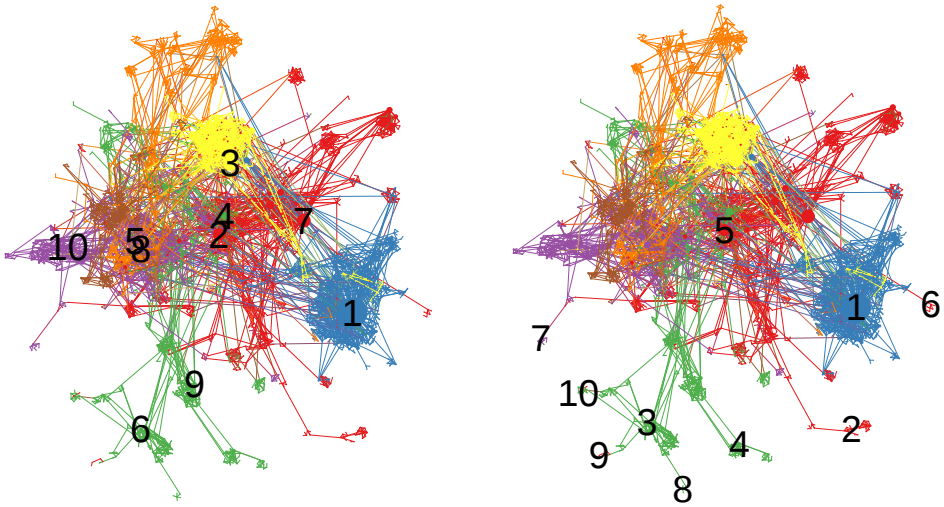
Finally, **DBLP** coauthorship graph is a noisier (and harder) classification problem (Figure A.4). Similar to the ISOLETe 4-nn undirected graph, we can infer that Σ-optimality picks better nodes because it has more queries in the denser regions in the central part of the graph, whereas V-optimality clearly picks many outliers.

[3] https://gephi.org/



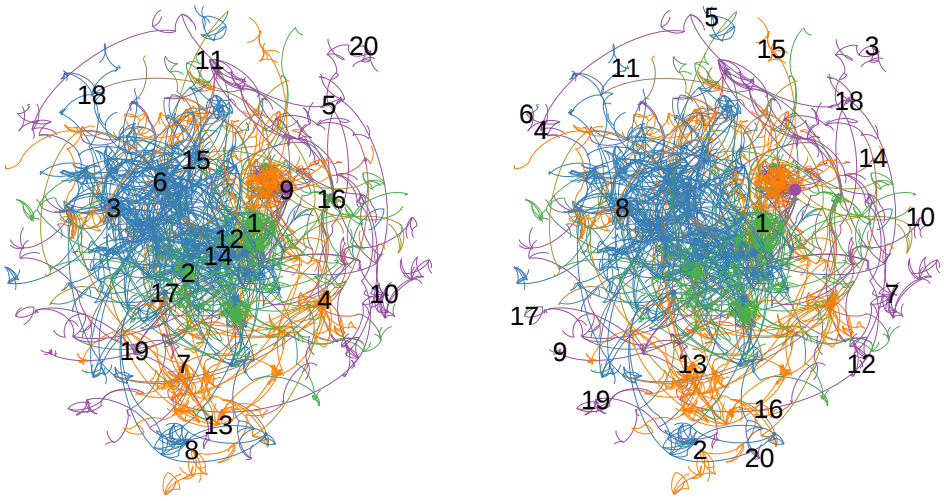
(a) Σ-optimality (b) V-optimality

Figure A.2: ISOLETe 4-nn undirected graph. Labels are the order of queries. Colors mean classes.



(a) Σ-optimality (b) V-optimality

Figure A.3: **Cora** citation graph. First 10 queries. Colors mean classes.



(a) Σ-optimality (b) V-optimality

Figure A.4: **DBLP** coauthorship graph. First 20 queries. Colors mean classes.

B

Active Needle Search Proofs

B.1 Theoretical Properties for Passive Sensing

Theorem B.1 (**Theorem 4.2** in the main document; limits of any passive methods using region sensing)**.** *Assume $\boldsymbol{\beta}$ has prior π_0 (uniform random on $\mathcal{S}_\mu\binom{n}{k}$). Any passive method with T noiseless region measurements on 1D must incur Bayes risk $\bar{\epsilon}_T \geq \frac{n-k}{n-1}(1-\frac{2T}{n})$; to guarantee $\bar{\epsilon}_T \leq \epsilon$, it requires $T \geq \frac{n}{2}(1-\frac{n-1}{n-k}\epsilon)$.*

Proof. We count the number of non-identifiability models with T noiseless observations, particularly when $T < \frac{n}{2}$.

An aggregate measurement on region $[a_i, b_i) \subset [1, n+1)$ cannot identify the sparse support inside $[a_i, b_i)$ (or its complement), unless it intersects with another aggregate measurement. Should two measurement regions intersect, the model is still non-identifiable inside the intersection, set differences, and the complement of the union of both. To find out the set of all disjoint subsets where the model is non-identifiable given any passive design with m region measurements, $\{[a_i, b_i) \subset [1, n+1) : i = 1, \ldots, m\}$, we simply sort the unique end points as $c_1 < \cdots < c_p \in \{a_i, \ldots, a_m\} \cup \{b_1, \ldots, b_m\}$, where $p \leq 2m$, and use the following set of p elementary subsets:

$$\Big\{\underbrace{[c_j, c_{j+1})}_{C_j} : j = 1, \ldots, p-1\Big\} \cup \Big\{\underbrace{[c_p, n+1) \cup [1, c_1)}_{C_p}\Big\}, \tag{B.1}$$

where the last subset is created to ensure that the number of sparse supports in the full set equals k. Notice, (B.1) is also the largest set of disjoint subsets that can be created using intersections, unions, and complements on the regions of measurements.

We will continue our discussion assuming that the measurements are made on the subsets contained in (B.1). When the observations are noiseless, (B.1) is a superior design than the original design, whose outcomes can be inferred as

$$\mathbf{x}_{[a_i,b_i)}^\top \boldsymbol{\beta} = \sum_{j=1}^{p-1} \frac{c_{j+1}-c_j}{b_i-a_i} \mathbf{x}_{[c_j,c_{j+1})}^\top \boldsymbol{\beta}.$$

At this point, it is easy to see that the minimum sample size to guarantee that the signals can be fully identifiable in the worst case is $T \geq \frac{n}{2}$; the necessary (and sufficient) condition is to have $|C_j| = 1, \forall j = 1, \ldots, p$, which requires $2T \geq p \geq n$.

For $\epsilon > 0$, we compute the expected Delta-risk given any fixed design which yields p elementary subsets as shown in (B.1). Let $n_j = |C_j|, j = 1, \ldots, p$. If the model $\boldsymbol{\beta}$ distributes k_j supports in subset C_j, respectively, then on any region where $n_j > k_j > 0$, the inference algorithm can only make a random guess, e.g., for the first k_j elements. Let $\boldsymbol{\beta}_{C_j}$ be the signal vector on subset C_j, the conditional expected error on this subset is:

$$\begin{aligned}
\mathbb{E}\big[|\boldsymbol{\beta}_{C_j} \Delta \hat{\boldsymbol{\beta}}_{C_j}| \mid k_j\big] &= \sum_{e_j=1}^{k_j} \frac{\binom{n_j-k_j}{e_j}\binom{k_j}{k_j-e_j}}{\binom{n_j}{k_j}} e_j = \sum_{e_j=1}^{k_j} \frac{\binom{n_j-k_j-1}{e_j-1}\binom{k_j}{k_j-e_j}}{\binom{n_j}{k_j}} (n_j-k_j) \\
&= \frac{\binom{n_j-1}{k_j-1}}{\binom{n_j}{k_j}} (n_j-k_j) = \frac{k_j(n_j-k_j)}{n_j}.
\end{aligned}$$

The total risk conditioned on all of $k_j : j = 1, \ldots, p$ is:

$$\mathbb{E}\big[|\boldsymbol{\beta} \Delta \hat{\boldsymbol{\beta}}| \mid k_1, \ldots, k_p\big] = \sum_{j=1}^{p} \mathbb{E}\big[|\boldsymbol{\beta}_{C_j} \Delta \hat{\boldsymbol{\beta}}_{C_j}| \mid k_j\big] = \sum_{j=1}^{p} \frac{k_j(n_j-k_j)}{n_j}.$$

Using the law of total expectation assuming $\boldsymbol{\beta}$ to be uniformly distributed, we can compute the expected error of the given passive design as

$$\begin{aligned}
\mathbb{E}|\boldsymbol{\beta} \Delta \hat{\boldsymbol{\beta}}| &= \sum_{k_1+\cdots+k_p=K} \left(\frac{\prod_{j=1}^{p} \binom{n_j}{k_j}}{\binom{n}{K}} \right) \left(\sum_{j=1}^{p} \frac{k_j(n_j-k_j)}{n_j} \right) \\
&= \sum_{j=1}^{p} \sum_{k_1+\cdots+k_p=K} \frac{\prod_{j'=1}^{p} \binom{n_{j'}}{k_{j'}}}{\binom{n}{K}} \frac{k_j(n_j-k_j)}{n_j} \\
&= \sum_{j=1}^{p} \sum_{k_1+\cdots+k_p=K} \frac{(n_j-1)\binom{n_j-2}{k_j-1} \prod_{j' \neq j} \binom{n_{j'}}{k_{j'}}}{\binom{n}{K}} \\
&= \sum_{j=1}^{p} \frac{(n_j-1)\binom{n-2}{K-1}}{\binom{n}{K}} = \frac{(n-p)\binom{n-2}{K-1}}{\binom{n}{K}} = \frac{K(n-K)}{n(n-1)}(n-p) \leq K\frac{(n-K)(n-2T)}{n(n-1)}
\end{aligned} \tag{B.2}$$

To guarantee $\mathbb{E}|\boldsymbol{\beta}\Delta\hat{\boldsymbol{\beta}}| \leq K\epsilon$, by solving (B.2) $\leq K\epsilon$, a passive design requires a minimal sample size of

$$T \geq \frac{p}{2} \geq \frac{n}{2}\Big(1 - \frac{n-1}{n-K}\epsilon\Big).$$

□

Corollary B.2. *Using noiseless region-sensing observations, a passive design in 1D with $T \leq \frac{n}{2}$ region measurements achieves the optimal average-case Delta-risk, if and only if it can separate the search space into $2m$ disjoint subsets using intersections, unions, and complements of the measurement regions. The following example is adapted from Gray code:*

t	$\mathbf{x}^\top$							
1	0	0	0	0	1	1	1	1
2	0	0	1	1	1	1	0	0
3	0	1	1	0	0	0	0	0
4	0	0	0	0	0	1	1	0
$\cdots$	$\cdots$ *(the pattern cycles)*							

Proof. To minimize (B.2), it is sufficient to find passive designs where $p = 2T$, given that the region aggregate measurements are noiseless. The expected risk of (B.2) turns out to be independent of the sizes of each elementary subset C_j (which one can verify with a minimal example where $n = 4$, $K = 2$, and $p = 2$), which suggests that all designs that yield $p = 2T$ have the same average-case Delta-risk with noiseless region aggregate measurements. □

Notice that the Gray-code design may not be optimal when the measurements are noisy. For this reason, we also included point sensing in our main paper, which yields the same order of sample complexity and performs better when the measurement noise is large.

B.2 Theoretical Properties for Active Sensing

The main goal of this section is to show that our main algorithm, *Region Sensing Index* (**RSI**), has the sample complexity guarantees show as Theorem 4.4 in the main paper. The main paper includes a proof sketch with three major steps. We show their details in 3 respective subsections.

B.2.1 Basic Properties of Information Gain (IG)

Recall that the observation model is $y_t = \mathbf{x}_t^\top \boldsymbol{\beta} + \epsilon_t$, where $\boldsymbol{\beta} \in \mathcal{S}_\mu\binom{n}{k}$, $\boldsymbol{\beta} \sim \pi_t$, and $\epsilon_t \sim \mathcal{N}(0, \sigma_t^2)$. Omitting the time index t in this subsection, the information gain (IG) to be maximized

in every step is defined as

$$
\begin{aligned}
& I(\boldsymbol{\beta}; y \mid \mathbf{x}, \pi) = H(y \mid \mathbf{x}, \pi) - \mathbb{E}[H(y \mid \mathbf{x}, \boldsymbol{\beta}) \mid \pi] \\
\Leftrightarrow \quad & I(\gamma; y \mid \lambda, \mathbf{p}) = H(y \mid \lambda, \mathbf{p}) - H(\epsilon), \qquad \text{(B.3)} \\
& \text{where} \quad f(y \mid \lambda, \mathbf{p}) = \sum_{c=0}^{K} p_c \phi(y - c\lambda) \\
& \qquad \lambda = \mu w_{\mathbf{x}}, \quad \gamma = \frac{\mathbf{x}^\top \boldsymbol{\beta}}{\lambda}, \\
& \qquad p_c = \Pr(\gamma = c) = \sum_{\boldsymbol{\beta}: \mathbf{x}^\top \boldsymbol{\beta} = c\lambda} \pi_t(\boldsymbol{\beta}).
\end{aligned}
$$

There are two basic properties: Lemma B.3 that is both directly applied in Section 4.3.1 Accelerations and indirectly used in the later proof sketch; and Proposition B.4 that appears as Proposition 4.5 in the main paper.

Basic Property 1

Lemma B.3 (Concavity and monotonicity)**.** *$I(\gamma; y \mid \lambda, \mathbf{p})$ is concave in $\mathbf{p} \in \mathbb{R}_+^{K+1}$, which includes the convex simplex of $\Delta^K = \{\mathbf{p} \in [0,1]^{K+1} : \mathbf{p}^\top \mathbf{1} = 1\}$, if $0 < \lambda < \infty$ remains constant. On the other hand, $I(\gamma; y \mid \lambda, \mathbf{p})$ with fixed $\mathbf{p} \in \Delta^K$ is monotone-increasing as λ increases.*

Proof. Concavity and monotonicity can be verified using derivatives. Notice the second term in (B.3) is constant. Here are the equations for the first term as well as its first and second order derivatives, omitting the dependency on $\mathbf{p}$ and λ for simplicity:

$$
\begin{aligned}
H(y; \lambda, \mathbf{p}) &= -\int f(y) \log f(y) \mathrm{d}y, \\
\partial H(y; \lambda, \mathbf{p}) &= -\int \big(1 + \log f(y)\big) \partial f(y) \, \mathrm{d}y, \\
\partial^2 H(y; \lambda, \mathbf{p}) &= -\int \left(\frac{\partial f(y) \partial f(y)^\top}{f(y)} + \big(1 + \log f(y)\big) \partial^2 f(y) \right) \mathrm{d}y
\end{aligned}
$$

Part 1. To show concavity in $\mathbf{p}(\geq 0)$, let $\boldsymbol{\phi}_{\boldsymbol{\lambda}}(y) = (\phi(y), \phi(y - \lambda), \ldots, \phi(y - K\lambda))^\top$ and write out the gradient and the Hessian of $H(y; \lambda, \mathbf{p})$ with respect of $\mathbf{p}$ as:

$$
\begin{aligned}
\frac{\partial H(y; \lambda, \mathbf{p})}{\partial \mathbf{p}^\top} &= -\int_{-\infty}^{\infty} \big(1 + \log f(y)\big) \boldsymbol{\phi}_{\boldsymbol{\lambda}}(y) \, \mathrm{d}y \\
\frac{\partial^2 H(y; \lambda, \mathbf{p})}{\partial \mathbf{p} \partial \mathbf{p}^\top} &= -\int_{-\infty}^{\infty} \frac{1}{f(y)} \boldsymbol{\phi}_{\boldsymbol{\lambda}}(y) \boldsymbol{\phi}_{\boldsymbol{\lambda}}(y)^\top \mathrm{d}y
\end{aligned}
$$

Notice $\boldsymbol{\phi}_{\boldsymbol{\lambda}}(y)\boldsymbol{\phi}_{\boldsymbol{\lambda}}(y)^\top$ is a PSD Gram matrix, which is preserved under integration. Further, the integral returns a PD matrix if the distribution is not degenerate ($\lambda > 0$ and $p_k > 0$ for at least two distinct ks)

Part 2.1 For monotonicity in $\lambda(> 0)$, in the case when $K = 1$, the derivative with respect to λ is

$$\begin{aligned}\frac{\partial H(y)}{\partial \lambda} &= -\int \big(1 + \log f(y)\big) \cdot p_1 \phi(y - \lambda) \cdot (y - \lambda)\, \mathrm{d}y \\ &= -p_1 \int \log f(y) \cdot \phi(y - \lambda) \cdot (y - \lambda)\, \mathrm{d}y \\ &= -p_1 \int \log f(y + \lambda) \cdot \phi(y) \cdot (y)\, \mathrm{d}y, \end{aligned} \tag{B.4}$$

where the first line removes constant integrals and the second shifts the variable. In order to show that (B.4) is nonnegative, pair up y and $-y$ for $y > 0$ and notice that, by assuming $\lambda > 0$,

$$\phi(y + \lambda) \le \phi(-y + \lambda) \Rightarrow f(y + \lambda) \le f(-y + \lambda).$$

The bigger λ, the larger derivative it has.

Part 2.2 In general when $K \ge 1$, we can write out the derivative as

$$\begin{aligned}\frac{\partial H(y; \lambda, \mathbf{p})}{\partial \lambda} &= -\int_{-\infty}^{\infty} \big(1 + \log f(y)\big) \sum_{k=0}^{K} p_k \phi(y - k\lambda)(y - k\lambda) k\, \mathrm{d}y \\ &= -\sum_{k=1}^{K} \int_{-\infty}^{\infty} (1 + \log f(y)) \sum_{t=k}^{K} p_t \phi(y - t\lambda)(y - t\lambda)\, \mathrm{d}y \end{aligned} \tag{B.5}$$

Define $h_k(y) = \sum_{t=k}^{K} p_t \phi(y - t\lambda)$; we have

$$0 = -h_k(y) \log h_k(y)\Big|_{-\infty}^{\infty} = \int_{-\infty}^{\infty} (1 + \log h_k(y)) \sum_{t=k}^{K} p_t \phi(y - t\lambda)(y - t\lambda)\, \mathrm{d}y \tag{B.6}$$

Consider each term of k in (B.5) and add the corresponding terms from (B.6); using $\ell_k = \sum_{s=0}^{k-1} p_s \phi(y - s\lambda)$, we get

$$\frac{\partial H(y; \lambda, \mathbf{p})}{\partial \lambda} = -\sum_{k=1}^{K} \int_{-\infty}^{\infty} \log\left(1 + \frac{\ell_k(y)}{h_k(y)}\right) \sum_{t=k}^{K} \phi(y - t\lambda)(y - t\lambda)\, \mathrm{d}y. \tag{B.7}$$

The only remaining task is to show that $r_k(y) = \frac{\ell_k(y)}{h_k(y)}$ is monotone decreasing with respect to y, which is sufficient to guarantee that (B.7) ≥ 0, due to the odd symmetry of the remaining integrand parts around $y = t\lambda$. Take the derivative of $r_k(y)$ with respect to y:

$$\begin{aligned} r_k'(y) &= \frac{\ell_k'(y) h_k(y) - \ell_k(y) h_k'(y)}{h_k^2(y)} = \frac{\sum_{s<k\le t} p_s p_t \big(\phi_s'(y)\phi_t(y) - \phi_s(y)\phi_t'(y)\big)}{h_k^2(y)} \\ &= \sum_{s<k\le t} p_s p_t \frac{\phi_t^2(y)}{h_k^2(y)} \left(\frac{\phi_s(y)}{\phi_t(y)}\right)' = \sum_{s<k\le t} p_s p_t \frac{\phi_t^2(y)}{h_k^2(y)} \left(\frac{\phi_s(y)}{\phi_t(y)}\right) \cdot (s\lambda - t\lambda) \le 0, \end{aligned}$$

where to simplify notations, we denote the composite function $\phi_s(y) = \phi(y - s\lambda)$. The inequality is strict if $\lambda > 0$ and $p_k \ne 0$ for at least two $k \in \{0, \dots, K\}$. □

Basic Property 2

Proposition B.4 (**Proposition 4.5** in the main document; a lower bound for the IG of a design)**.** *The IG score of a region sensing design has lower bounds with respect to its design parameters* $(\lambda, \mathbf{p})$*, as*

$$I(\gamma; y \mid \lambda, \mathbf{p}) \geq 2q_c\bar{q}_c(2\Phi(\tfrac{\lambda}{2}) - 1)^2 \geq \tfrac{1}{12}\min\{q_c, \bar{q}_c\}\min\{\lambda^2, 3^2\}, \quad \forall 1 \leq c \leq K, \tag{B.8}$$

where $q_c = P(\gamma \geq c) = \sum_{\kappa \geq c} p_\kappa$*,* $\bar{q}_c = 1 - q_c$*, and* $\Phi(u)$ *is the standard normal cdf.*

Proof of Proposition B.4. **To show** (B.8)**, first inequality:** Pick any $1 \leq c \leq K$; let $v = 1_{\gamma \geq c}$ and $\hat{v} = 1_{y > (c-1/2)\lambda}$ be two binary truncations of the original variables, γ and y, respectively. These truncations lose information:

$$\begin{aligned} I(\gamma; y \mid \mathbf{p}, \lambda) \geq I(v; \hat{v} \mid \mathbf{p}, \lambda) &= \mathbb{E}^v K\big((\hat{v} \mid v) \,\|\, \hat{v}\big) \\ &\geq 2 \sum_{v_0 \in \{0,1\}} P(v = v_0) \sup_{\hat{v}_0} \underbrace{\Big|P(\hat{v} = \hat{v}_0 \mid v = v_0) - P(\hat{v} = \hat{v}_0)\Big|^2}_{\triangleq \hat{\delta}(v_0, \hat{v}_0)}, \end{aligned} \tag{B.9}$$

where $K(\cdot \,\|\, \cdot)$ is the Kullback–Leibler divergence and the second line comes from Pinsker's inequality.

Consider any realization of $v = v_0$ and choose $\hat{v}_0 = v_0$; using the rule of total probability and direct calculation,

$$\begin{aligned} \hat{\delta}(v_0, v_0) &= \Big|P(\hat{v} = v_0 \mid v = v_0) - P(v = v_0)P(\hat{v} = v_0 \mid v = v_0) - P(v \neq v_0)P(\hat{v} = v_0 \mid v \neq v_0)\Big| \\ &= P(v \neq v_0)\Big|P(\hat{v} = v_0 \mid v = v_0) - P(\hat{v} = v_0 \mid v \neq v_0)\Big| \\ &\geq P(v \neq v_0)\Big[\Phi\Big(\frac{\lambda}{2}\Big) - \Big(1 - \Phi\Big(\frac{\lambda}{2}\Big)\Big)\Big] = P(v \neq v_0)\Big(2\Phi\Big(\frac{\lambda}{2}\Big) - 1\Big), \end{aligned} \tag{B.10}$$

where $\Phi(\frac{\lambda}{2})$ is a lower bound on the probability of correct estimation, based on the worst-case draw of γ such that y cannot be more than $\frac{\lambda}{2}$ away from γ in the direction that leads to estimation errors. Taking (B.10) to (B.9) yields the first part of the result.

To show (B.8)**, second inequality:** So far we have shown an analytical lower bound for $I(\gamma; y \mid \lambda, \mathbf{p})$. To make the result even more interpretable, we can further numerically evaluate the Gaussian tail distribution, to find two constants, C_1 and C_2, such that

$$\Phi(x) - \frac{1}{2} = \int_0^x \phi(u)\,\mathrm{d}u = \int_0^x \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}\,\mathrm{d}u \geq C_1 \min\{x, C_2\}.$$

Since $\Phi(x)$ is monotone-increasing, we can fix C_2 to find the worst difference quotient, ${}^{\Phi(x)}/_{x}$, $\forall x \in (0, C_2]$. In fact, we can directly assign $C_1 \leq {}^{\Phi(C_2)}/_{C_2}$, because $\phi(u)$ is monotone-decreasing as u increases. We choose $C_2 = \frac{3}{2}$ and $C_1 = \frac{1}{\sqrt{12}}$, which yields

$$\Big(2\Phi\Big(\frac{\lambda}{2}\Big) - 1\Big)^2 \geq \frac{1}{12}\min\big\{\lambda^2, 3^2\big\}.$$

□

Proposition B.5 (An upper bound for the IG of a design)**.** *When $K = 1$ and WLOG $p_1 \leq \frac{1}{2}$, the upper bound of IG derived from Jensen's inequality and max-entropy principle is $I(\gamma; y \mid \lambda, \mathbf{p}) \leq \frac{1}{2} p_1 \lambda^2$, which is on the same order of* (B.8) *when $\lambda < O(1)$. In the $\lambda \gg 1$ case, the IG is naturally upper-bounded by a Bernoulli experiment with noiseless observation, $H(\mathcal{B}(p_1)) = -p_1 \log(p_1) - (1-p_1)\log(1-p_1) = \tilde{O}(p_1)$. Therefore, Proposition B.4 is a good approximation to the true IG in all scenarios. (See Figure B.1 for an empirical visualization.) The general upper bound is not tight for general $k > 1$.*

Proof. The upper bound can be shown by Jensen's inequality and max-entropy principle. It is also tight when $k = 1$. Omitting $\mathbf{p}$ and λ,

$$I(\gamma; y) = I(\gamma; \gamma + \epsilon) = H(\gamma + \epsilon) + H(\gamma + \epsilon \mid \gamma) = H(\gamma + \epsilon) - H(\epsilon).$$

We only need to find the largest entropy for $H(\gamma + \epsilon)$ given $\mathbf{p}$ and λ. By Jensen's inequality, under the same mean and variance, a normal distribution has the largest entropy, where we have:

$$\mathbb{E}(\gamma+\epsilon) = \mathbb{E}(\gamma)+\mathbb{E}(\epsilon) = p_1\lambda, \quad \sigma^2_{\text{mar}} = \text{Var}(\gamma+\epsilon) = \text{Var}(\gamma)+\text{Var}(\epsilon)+2\text{Cov}(\gamma,\epsilon) = p_1\lambda^2+1.$$

We can then use a normal distribution with the above mean and variance as a upper bound to:

$$I(\gamma; y) = H(\gamma + \epsilon) - H(\epsilon) \leq \frac{1}{2}\log(2\pi e \sigma^2_{\text{mar}}) - \frac{1}{2}\log(2\pi e) = \frac{1}{2}\log(1 + p_1\lambda^2) \leq \frac{1}{2} p_1 \lambda^2$$

□

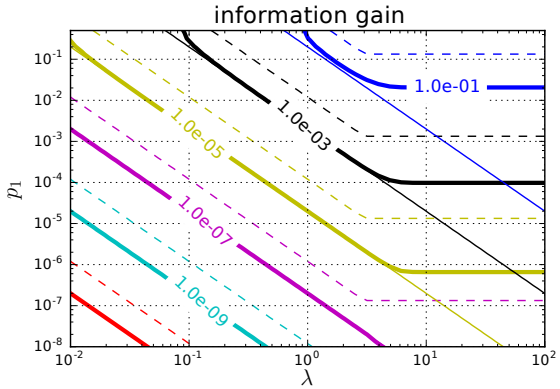


Figure B.1: Level sets of IG $I(\gamma; y \mid \lambda, p_1)$ for different values of p_1 and λ, when $k = 1$. The thin lines below each true value indicate IG upper bounds (Proposition B.5) and the dashed lines are the phase-changing lower bound from Proposition B.4. The phase-changing bound is more useful because it produces insights about optimal region selection, usually at the point of phase-change, whereas the upper bound is non-informatively linear in the log-log plot.

B.2.2 Minimum Information Gain of the Chosen Region in Each Iteration

This subsection aims to formalize the main observation in our main paper, which is that the information gain of all of the chosen measurements from RSI remain consistently large, before active search terminates with minimal Bayes risk. This observation implies a constant speed at which the model uncertainty can be reduced in expectation, leading to the upper bounds on sample complexity in Section B.2.3.

Recall that the Bayes risk is defined by $\bar{\epsilon}_t = \min_{|\hat{S}|=k} \frac{1}{k}\mathbb{E}[|\hat{S}\Delta S| \mid \pi_t]$, where Δ is the symmetric set difference operator. If we include the Bayes inference rule $\pi_t(\boldsymbol{\beta}) \propto \pi_0(\boldsymbol{\beta}) \prod_{\tau=1}^{t} p(y \mid \mathbf{x}_\tau, \boldsymbol{\beta})$, we can see that $\bar{\epsilon}_t$ is essentially a function of the collected data $D_t = \{(\mathbf{x}_\tau, y_\tau) : 1 \le \tau \le t\}$. The following lemma paraphrases Lemma 4.6 in the main document, with the time index t omitted.

Lemma B.6 (Minimum IG of the chosen regions)**.** *WLOG, assume n is a multiple of $2k$. At any step, given the data collection outcomes D and the current Bayes risk $\bar{\epsilon}(D)$, we can always find a region A of size at most $\frac{n}{k}$, such that $\lambda^2 \ge \frac{\mu^2}{a} = \frac{k\mu^2}{n}$ and $\frac{\bar{\epsilon}(D)}{2} \le \mathbb{E}[\gamma_A \mid D] \le 1 - \frac{\bar{\epsilon}(D)}{2}$ (we call it* Condition E*), which further yields*

$$I(\gamma; y \mid \lambda, \mathbf{p}) \ge I^*_{\bar{\epsilon}} = \frac{\bar{\epsilon}(D)}{25k} \min\{k\lambda^2, 3^2\} \ge \frac{\bar{\epsilon}(D)}{25k} \min\{\tfrac{k^2\mu^2}{n}, 3^2\}. \tag{B.11}$$

Condition E

Lemma B.6 states the result in two steps: (a) the fact that the posterior model after collecting data D still has large Bayes risk implies the existence of a very informative region that satisfies *Condition E* and (b) sensing on this region indeed yields nontrivial information, measured in terms of IG (B.11). We will split the proof into these two steps, accordingly.

Lemma B.7 (A region that satisfies *Condition E*)**.** *In 1d search with unit ℓ_2-norm measurements, WLOG, assume n is a multiple of $2k$. At any step, given the collected data D and the current Bayes risk $\bar{\epsilon}(D)$:*

1. There always is a region B of size no larger than $\frac{n}{k}$, such that $\lambda_B^2 \ge \frac{\mu^2}{|B|} = \frac{k\mu^2}{n}$ and $\mathbb{E}[\gamma_B \mid D] \ge \frac{\bar{\epsilon}(D)}{2}$

2. There always is a subregion $A \subset B$ that satisfies Condition E*:*

$$\lambda_A^2 \ge \frac{k\mu^2}{n} \quad \textit{and} \quad \frac{\bar{\epsilon}(D)}{2} \le \mathbb{E}[\gamma_A \mid D] \le 1 - \frac{\bar{\epsilon}(D)}{2}$$

Proof. **Part 1.** Suppose the current minimizer of the posterior Bayes risk is $\hat{S} = \hat{S}(D) = \arg\max_{S'} \sum_{\hat{\jmath} \in S'} \mathbb{E}[\beta_{\hat{\jmath}} \mid D]$. Evenly split the domain into K disjoint and contiguous regions and take their largest disjoint and contiguous subsets that do not intersect with $\hat{S}$. There are at most $G \le 2K$ such sets; let them be $B_1, \ldots, B_G$. We use $\gamma(B_g) = \sum_{j \in B_g} \mathbf{1}_j^\top \boldsymbol{\beta}$ to denote the

corresponding region latent variables in a region B_g. The region $B = \arg\max_{B_{g'}} \mathbb{E}[\gamma(B_{g'}) \mid D]$ yields

$$\mathbb{E}[\gamma(B) \mid D] \geq \frac{\sum_{g'=1}^{G} \mathbb{E}[\gamma(B_{g'}) \mid D]}{2K} = \frac{K - \mathbb{E}[\gamma(\hat{S}) \mid D]}{2K} = \frac{\bar{\epsilon}(D)}{2},$$

due to the additivity, $\sum_g \gamma(B_g) + \gamma(\hat{S}) = \sum_{j\in[n]} \mathbf{1}_j^\top \boldsymbol{\beta} = K$.

Part 2. Let $A \subset B$ be the smallest contiguous subset such that $\mathbb{E}[\gamma(A) \mid D] \geq \frac{\bar{\epsilon}(D)}{2}$. Notice the maximum certainty of any point in $j \in A \subseteq \mathcal{X} \setminus \hat{S}$ is

$$\mathbb{E}[\beta_j \mid D] \leq \min_{\hat{\jmath}\in\hat{S}}(1 - \mathbb{E}[\beta_{\hat{\jmath}} \mid D]) \leq 1 - \bar{\epsilon}(D).$$

We then use the additivity of expectation to obtain

$$\mathbb{E}[\gamma(A) \mid D] \leq \mathbb{E}[\gamma(A\backslash\{j\}) \mid D] + \mathbb{E}[\beta_j \mid D] < \frac{\bar{\epsilon}(D)}{2} + (1 - \bar{\epsilon}(D)) = 1 - \frac{\bar{\epsilon}(D)}{2}, \quad \forall j \in A, \text{ i.e., } j \notin \hat{S}.$$

□

IG of the Chosen Region

The following obtains Lemma B.8 with additive terms of K^2. It provides advantages over the straight-forward calculation in the main paper (which yields results with multiplicative factors of K).

Lemma B.8 (Maximum IG when the outcome expectation is bounded)**.** *For any design on K-sparse models, if there exists $0 < \bar{\epsilon} < 1$ and a design $(\mathbf{x}, A, \lambda, \gamma)$ such that $\frac{\bar{\epsilon}}{2} \leq \mathbb{E}\gamma \leq 1 - \frac{\bar{\epsilon}}{2}$, where $\gamma = \mathbf{x}^\top\boldsymbol{\beta}$ is latent variable of signal counts in the measurement region, then the information of the experiment is lower-bounded by*

$$I(\gamma; y \mid \mathbf{p}, \lambda) \geq \frac{\bar{\epsilon}}{25K} \min\{K\lambda^2, 3^2\}$$

Proof. We use the fact that IG is concave in $\mathbf{p}$ and we only check the vertices of the simplex of feasible probabilities to find its lower bound:

$$\begin{cases} p_k \geq 0, \quad k = 1, \ldots, K, & \text{(Constraint } H_1, \ldots, H_K\text{)}; \\ \sum_{k=1}^{K} p_k \leq 1, & \text{(Constraint } H_0\text{)}; \\ \sum_{k=1}^{K} k p_k \geq \frac{\bar{\epsilon}}{2}, & \text{(Constraint } E_1\text{)}; \\ \sum_{k=1}^{K} k(1 - p_k) \geq \frac{\bar{\epsilon}}{2}, & \text{(Constraint } E_2\text{)}, \end{cases}$$

where $p_0 = 1 - \sum_{k=1}^{K} p_k$ can be decided explicitly. All vertices of the simplex, including infeasible vertices, can be found by solving K linear systems constructed from the $(K + 3)$

linear constraints. Since E_1 and E_2 cannot be satisfied simultaneously for any $\bar{\epsilon} < 1$, we can enumerate all the remaining vertices and write out their respective nonzero values:

$$\begin{array}{lll} p_k = 1, & & \text{from } \cap_{k' \neq k} H_{k'}; \\ p_k + p_\ell = 1, & kp_k + \ell p_\ell = \frac{\bar{\epsilon}}{2}, & \text{from } \cap_{k' \neq k,\ell} H_{k'} \cap E_1; \\ p_k + p_\ell = 1, & kp_k + \ell p_\ell = 1 - \frac{\bar{\epsilon}}{2}, & \text{from } \cap_{k' \neq k,\ell} H_{k'} \cap E_2. \end{array}$$

The first row is infeasible when $\bar{\epsilon} > 0$. We then bound the IG for the other rows. Without loss of generality, assume $\ell < k$. Then, all feasible cases require $\ell = 0$ and yield $\min\{p_k, p_\ell\} \geq \frac{\bar{\epsilon}}{2k}$. Using Proposition 4.5,

$$I(v; u \mid \mathbf{p}, \lambda) \geq \frac{\bar{\epsilon}}{25K} \min\{K^2\lambda^2, 3^2\} \geq \frac{\bar{\epsilon}}{25K} \min\{K\lambda^2, 3^2\}.$$

□

Proof of Lemma B.6. The design from Lemma B.7 satisfies both *Condition E* and $\lambda \geq \frac{K\mu^2}{n}$, where we can then apply Lemma B.8 to obtain the conclusion. □

B.2.3 The Proof of Theorem 4.4

Lemma B.6 implies that the entropy in the posterior distribution, $H(\boldsymbol{\beta} \mid \pi_t) = -\sum_{\boldsymbol{\beta}} \pi_t(\boldsymbol{\beta}) \log \pi_t(\boldsymbol{\beta})$, decreases at least by I^*_ϵ with every measurement in expectation, starting with $H(\boldsymbol{\beta} \mid \pi_0) \leq k \log n$. Since the posterior entropy cannot be negative, RSI must terminates in finite times in expectation.

Theorem B.9 (**Theorem 4.4** in the main document; sample complexity of RSI)**.** *In active search of k sparse signals with strength μ in 1d physical space of size $n(\geq 2k)$, given any $\epsilon > 0$ as tolerance of posterior Bayes risk,* RSI *using region sensing has bounded expected number of actual measurements before stopping,*

$$\bar{T}_\epsilon = \mathbb{E}[\min\{\mathcal{T} : \bar{\epsilon}(D_\mathcal{T}) \leq \epsilon\}] \leq 50\Big(\frac{n}{\mu^2} + \frac{k^2}{9}\Big) \log_2\Big(\frac{2}{\epsilon}\Big) \log\Big(\frac{n}{\epsilon}\Big) = \tilde{O}\Big(\frac{n}{\mu^2} + k^2\Big),$$

where the expectation is taken over the prior distribution and sensing outcomes.

The Simple Approach

Definition B.10 (Stopping time)**.** *Define $T_\epsilon = \min_\mathcal{T}\{\bar{\epsilon}(D_\mathcal{T}) \leq \epsilon\}$ to be a random stopping time for an experiment to first yield less than ϵ posterior risk, $\bar{\epsilon}(D_\tau) = \frac{1}{K}\mathbb{E}[S\Delta\hat{S} \mid D_\tau] \leq \epsilon$. $T_\epsilon = T_\epsilon(\tau)$ can be determined given τ.*

Lemma B.11 (Simple Expectations on the Number of Measurements for Small Errors)**.** *Given any $\epsilon_1 > 0$, $t_0 \geq 0$, and the first t_0 data collection outcomes D_{t_0}, the expected number of*

additional measurements before the RSI *stops with posterior risk less than ϵ_1 is bounded in terms of $H_0 = H(\boldsymbol{\beta} \mid \pi_0)$ and I_{ϵ_1} defined in Lemma B.6, as*

$$\mathbb{E}(T_{\epsilon_1} - T_{\epsilon_0} \mid D_{t_0}) \leq \frac{H_0}{I_{\epsilon_1}} \leq \frac{25 H_0}{\epsilon} \max\Big\{\frac{n}{k\mu^2}, \frac{k}{9}\Big\}.$$

Remark B.12. *By taking $t_0 = 0$ and $H_0 \leq k \log n$, Lemma B.11 implies*

$$\bar{T_\epsilon} \leq \frac{25 \log(n)}{\epsilon_1} \max\Big\{\frac{n}{\mu^2}, \frac{k^2}{9}\Big\}.$$

Proof of Lemma B.11. Let $t = t_0 + s$ for any $s \geq 0$ and D_t be the random variable for the data collection outcomes until step t. According to Lemma B.6,

$$\begin{aligned} (T_{\epsilon_1} \mid D_t) > t \quad &\Rightarrow \quad H(\boldsymbol{\beta} \mid D_t) - \mathbb{E}^y\big[H(\boldsymbol{\beta} \mid D_t \cup \{\mathbf{x}, y\}) \mid D_t, \mathbf{x}_{t+1}\big] \geq I_{\epsilon_1} \\ &\Rightarrow \quad H(\boldsymbol{\beta} \mid D_t) \geq I_{\epsilon_1} + \mathbb{E}^y\big[H(\boldsymbol{\beta} \mid D_t \cup \{\mathbf{x}, y\}) \mid D_t, \mathbf{x}_{t+1}\big] \end{aligned}$$

Taking expectation over

$$\{D_t : (T_{\epsilon_1} \mid D_t) > t, D_{t_0}\} = \{D_t : \bar{\epsilon}(D_{t'}) > \epsilon_1, \forall t' \leq t, D_{t_0}\}$$

yields

$$\mathbb{E}\big[H(\boldsymbol{\beta} \mid D_t) \mid T_{\epsilon_1} > t, D_{t_0}\big] \geq I_{\epsilon_1} + \mathbb{E}\big[H(\boldsymbol{\beta} \mid D_{t+1}) \mid T_{\epsilon_1} > t, D_{t_0}\big], \tag{B.12}$$

where the expectation is taken over $(D_t \mid D_{t_0}, T_{\epsilon_1} > t)$ and $(D_{t+1} \mid D_{t_0}, T_{\epsilon_1} > t)$, respectively.

Next, we hope to apply Lemma B.6 at step $(t+1)$, but we have to make sure that the condition still holds, which is not directly implied by (B.12). To guarantee the conditions, we divide D_{t+1} into two cases and use the nonnegativity of entropy to relax the second case,

$$\begin{aligned} \mathbb{E}\big[H_{t+1}(\boldsymbol{\beta}) \mid T_{\epsilon_1} > t, D_{t_0}\big] &= P\big(T_{\epsilon_1} > t+1 \mid T_{\epsilon_1} > t, D_{t_0}\big)\mathbb{E}\big[H_{t+1}(\boldsymbol{\beta}) \mid T_{\epsilon_1} > t+1, D_{t_0}\big] \\ &\quad + P\big(T_{\epsilon_1} = t+1 \mid T_{\epsilon_1} > t, D_{t_0}\big)\mathbb{E}\big[H_{t+1}(\boldsymbol{\beta}) \mid T_{\epsilon_1} = t+1, D_{t_0}\big] \\ &\geq P\big(T_{\epsilon_1} > t+1 \mid T_{\epsilon_1} > t, D_{t_0}\big)\mathbb{E}\big[H_{t+1}(\boldsymbol{\beta}) \mid T_{\epsilon_1} > t+1, D_{t_0}\big]. \end{aligned}$$

We can then iterate beginning with $t = t_0$ as

$$
\begin{aligned}
&\mathbb{E}\big[H_{t_0}(\boldsymbol{\beta}) \mid D_{t_0}\big] \geq P\big(T_{\epsilon_1} > t_0 \mid D_{t_0}\big)\mathbb{E}\big[H_{t_0}(\boldsymbol{\beta}) \mid T_{\epsilon_1} > t_0, D_{t_0}\big] \\
&\geq P\big(T_{\epsilon_1} > t_0 \mid D_{t_0}\big)\Bigg(I_{\epsilon_1} + P\big(T_{\epsilon_1} > t_0 + 1 \mid T_{\epsilon_1} > t_0, D_{t_0}\big)\mathbb{E}\big[H_{t_0+1}(\boldsymbol{\beta}) \mid T_{\epsilon_1} > t_0 + 1, D_{t_0}\big]\Bigg) \\
&= P\big(T_{\epsilon_1} > t_0 \mid D_{t_0}\big)I_{\epsilon_1} + P\big(T_{\epsilon_1} > t_0 + 1 \mid D_{t_0}\big)\mathbb{E}\big[H_{t_0+1}(\boldsymbol{\beta}) \mid T_{\epsilon_1} > t_0 + 1, D_{t_0}\big] \\
&\geq P\big(T_{\epsilon_1} > t_0 \mid D_{t_0}\big)I_{\epsilon_1} + P\big(T_{\epsilon_1} > t_0 + 1 \mid D_{t_0}\big)\Bigg(I_{\epsilon_1} + \\
&\qquad + P\big(T_{\epsilon_1} > t_0 + 2 \mid T_{\epsilon_1} > t_0 + 1, D_{t_0}\big)\mathbb{E}\big[H_{t_0+2}(\boldsymbol{\beta}) \mid T_{\epsilon_1} > t_0 + 2, D_{t_0}\big]\Bigg) \\
&\geq P\big(T_{\epsilon_1} > t_0 \mid D_{t_0}\big)I_{\epsilon_1} + P\big(T_{\epsilon_1} > t_0 + 1 \mid D_{t_0}\big)I_{\epsilon_1} \\
&\qquad + P\big(T_{\epsilon_1} > t_0 + 2 \mid D_{t_0}\big)\mathbb{E}\big[H_{t_0+2}(\boldsymbol{\beta}) \mid T_{\epsilon_1} > t_0 + 2, D_{t_0}\big] \\
&\geq \dots \\
&\geq I_{\epsilon_1}\sum_{s=0}^{\infty} P\big(T_{\epsilon_1} > t_0 + s \mid D_{t_0}\big) = I_{\epsilon_1}\mathbb{E}\big(T_{\epsilon_1} - T_{\epsilon_0} \mid D_{t_0}\big),
\end{aligned}
$$

which leads to the conclusion given $\mathbb{E}[H_{t_0}(\boldsymbol{\beta}) \mid D_{t_0}] = H(\boldsymbol{\beta} \mid D_{t_0}) = H_0$. □

The Complex Approach

Lemma B.13 (Max entropy given Bayes error)**.** *For a K-sparse model, $\boldsymbol{\beta} \in \mathcal{S}\binom{n}{K}$, given $\bar{\epsilon} \geq \frac{1}{K}\sum_{j\in\hat{S}} P(\beta_j = 0) = \frac{1}{K}\mathbb{E}|S\Delta\hat{S}|$, the posterior entropy is at most*

$$
\begin{aligned}
H(\boldsymbol{\beta}) &\leq KH(\mathcal{B}(\bar{\epsilon})) + K\bar{\epsilon}\log n, && \text{(B.13)}\\
&\leq \frac{K}{2^r}\big(2r\log 2 + \log n\big), \qquad \textit{if } \bar{\epsilon} \leq \frac{1}{2^r}, \forall r = 0, 1, 2, \dots && \text{(B.14)}
\end{aligned}
$$

where $H(\mathcal{B}(\bar{\epsilon})) = -\bar{\epsilon}\log\bar{\epsilon} - (1-\bar{\epsilon})\log(1-\bar{\epsilon})$ is denoted as the entropy of a Bernoulli experiment with $\bar{\epsilon}$ success rate.

Proof. **Part 1.** Let $S = \{S_1, \dots, S_K\}$ be the set of supports of the random variable $\boldsymbol{\beta}$ that is modeled by the posterior distribution given the history data that leads to the current state. We can compute the expectation as

$$
\sum_{k=0}^{K} kP(|S\Delta\hat{S}| = k) = \mathbb{E}|\hat{S}\Delta S| \leq K\bar{\epsilon}. \tag{B.15}
$$

Define $p_k = p_k(\hat{S}) = P(|S\Delta\hat{S}| = k)$; the total entropy can be bounded:

$$H(\boldsymbol{\beta}) = -\sum_{k=0}^{K} \sum_{S:|S\Delta\hat{S}|=k} \pi(\boldsymbol{\beta}_S) \log \pi(\boldsymbol{\beta}_S) \tag{B.16}$$

$$\leq -\sum_{k=0}^{K} p_k \log \left(\frac{p_k}{\binom{K}{K-k}\binom{n-K}{k}} \right) \tag{B.17}$$

$$\leq -\sum_{k=0}^{K} p_k \log p_k + \sum_{k=0}^{K} p_k \log \binom{K}{k} + \sum_{k=0}^{K} k p_k \log n \tag{B.18}$$

$$= -\sum_{k=0}^{K} p_k \log p_k + \sum_{k=0}^{K} p_k \log \binom{K}{k} + K\bar{\epsilon} \log n \tag{B.19}$$

where (B.16) separate the joint probabilities into $(K+1)$ groups according to their values of $|S\Delta\hat{S}|$. Inside every group, (B.17) realizes a uniform distribution, which maximizes the entropy given any value of group marginal probability, p_k. We then relax the number of combination by $\log \binom{x}{K-k} \leq (K-k)\log x$, which yields (B.18). From there, we use the condition, reformulated as (B.15), to obtain (B.19).

The next step uses the principle of maximum entropy to realize the optimizer for (B.19), when the moments are bounded by (B.15). The Lagrangian of the constrained optimization is

$$L(\mathbf{p}; c, \rho) = -\sum_{k=0}^{K} p_k \log p_k + \sum_{k=0}^{K} p_k \log \binom{K}{k} + c\left(\sum_{k=0}^{K} p_k - 1\right) + \rho\left(\sum_{k=0}^{K} k p_k - K\bar{\epsilon}\right).$$

Setting the derivatives to zero yields

$$0 = \frac{\partial L}{\partial p_k} = -\log p_k + \log \binom{K}{k} + 1 + c + k\rho \quad \Rightarrow \quad p_k \propto \binom{K}{k} (e^{\rho})^k,$$

which implies that p_k is the probability of k outcomes in a binomial distribution with K rounds and an iid outcome probability of $p = \frac{1}{1+e^{-\rho}}$ in each round. Since the expectation of the total outcome is $K\bar{\epsilon}$, we have $p = \bar{\epsilon}$. Given the max-entropy binomial distribution and let $(X_1, \ldots, X_K)$ to be the outcome of each round; the entropy of their sum is upper bounded by the sum of their marginal entropies, which is K times the entropy of $H(\bar{\epsilon})$. So, we proved (B.13).

Part 2. To move forward to (B.14), we need an interim result when $\bar{\epsilon} \leq \frac{1}{2}$:

$$H(\bar{\epsilon}) \leq -2\bar{\epsilon}\log\bar{\epsilon} \quad \Rightarrow \quad H(\boldsymbol{\beta}) \leq -2K\bar{\epsilon}\log\bar{\epsilon} + K\bar{\epsilon}\log N, \tag{B.20}$$

To show the interim result, let $\ell(\bar{\epsilon}) = -\bar{\epsilon}\log\bar{\epsilon} + (1-\bar{\epsilon})\log(1-\bar{\epsilon})$; its derivatives are $\ell'(\bar{\epsilon}) = -\log\bar{\epsilon} - \log(1-\bar{\epsilon}) - 2$ and $\ell''(\bar{\epsilon}) = -\frac{1}{\bar{\epsilon}} + \frac{1}{1-\bar{\epsilon}}$. The concavity of $\ell(\bar{\epsilon})$ in $0 \leq \bar{\epsilon} \leq \frac{1}{2}$ where $\ell''(\bar{\epsilon}) \leq 0$ and $\ell(0) = \ell(\frac{1}{2}) = 0$ yield $\ell(\bar{\epsilon}) \geq 0$, i.e., $H(\bar{\epsilon}) \leq -2\bar{\epsilon}\log\bar{\epsilon}, \forall 0 \leq \bar{\epsilon} \leq \frac{1}{2}$.

Finally, (B.14) trivially holds when $r = 0$. Otherwise, substitute $\bar{\epsilon} \leq 2^{-r}$ with $r \geq 1$ in (B.20) yields the final conclusion. □

Proof of the final theorem. Let $\epsilon_r = 2^{-r}$ and $T_{\epsilon_r} = \min_{\mathcal{T}}\{\bar{\epsilon}(D_{\mathcal{T}}) \le \epsilon_r\}$, for $r = 0, 1, \ldots, \lceil \log_2(\frac{1}{\epsilon}) \rceil$. From Lemma B.11, we have

$$\mathbb{E}(T_{\epsilon_{r+1}} - T_{\epsilon_r} \mid D_t, T_{\epsilon_r} \le t) \le \frac{H(\boldsymbol{\beta} \mid D_t)}{I^*_{\epsilon_{r+1}}}. \tag{B.21}$$

We can use Lemma B.6 with $\epsilon_{r+1} = 2^{-r-1}$ to show

$$I^*_{\epsilon_{r+1}} \ge \frac{\epsilon_{r+1}}{25k} \min\Big\{\frac{k^2\mu^2}{n}, 9\Big\} \ge \frac{1}{50k2^r} \min\Big\{\frac{k^2\mu^2}{n}, 9\Big\}$$

and Lemma B.13 with $\bar{\epsilon}(D_t) \le \epsilon_r = 2^{-r}$ to bound

$$H(\boldsymbol{\beta} \mid D_t) \le \frac{k}{2^r}(2r \log 2 + \log n).$$

Put both bounds to (B.21) to get

$$\mathbb{E}(T_{\epsilon_{r+1}} - T_{\epsilon_r} \mid D_t, T_{\epsilon_r} \le t) \le 50 \max\Big\{\frac{n}{\mu^2}, \frac{k^2}{9}\Big\}(2r \log 2 + \log n).$$

Notice the right side is independent of D_t and t, using linearity of expectations,

$$\mathbb{E}(T_{\epsilon_{r+1}} - T_{\epsilon_r}) \le 50 \max\Big\{\frac{n}{\mu^2}, \frac{k^2}{9}\Big\}(2r \log 2 + \log n),$$

which further implies, using $R = \lceil \log_2 \frac{1}{\epsilon} \rceil < 1 + \log_2 \frac{1}{\epsilon}$,

$$\begin{aligned}
\mathbb{E}T_\epsilon &\le \sum_{r=0}^{R-1} \mathbb{E}\big(T_{\epsilon_{r+1}} - T_{\epsilon_r}\big) \\
&\le 50 \max\left\{\frac{n}{\mu^2}, \frac{k^2}{9}\right\} \sum_{r=0}^{R-1} (2r \log 2 + \log n) \\
&\le 50 \max\left\{\frac{n}{\mu^2}, \frac{k^2}{9}\right\} R((R-1) \log 2 + \log n) \\
&\le 50 \max\left\{\frac{n}{\mu^2}, \frac{k^2}{9}\right\} \log_2 \frac{2}{\epsilon} \log \frac{n}{\epsilon}
\end{aligned}$$

□

Bibliography

Yasin Abbasi-Yadkori. Online-to-confidence-set conversions and application to sparse stochastic bandits. 2012. 68

Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, 2005. 2

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *AISTATS*, pages 99–107, 2013a. 86

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *ICML (3)*, pages 127–135, 2013b. 86

Stephen Ansolabehere and Jonathan Rodden. Pennsylvania data files. 62

Ery Arias-Castro, Emmanuel J Candes, and Mark Davenport. On the fundamental limits of adaptive sensing. *Information Theory, IEEE Transactions on*, 2013. 7, 68, 71, 73, 76, 77, 80, 83

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397–422, 2003. 4, 16

Peter Auer, Nicoló Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002. 43

Erlend Aune, Jo Eidsvik, and Yvo Pokern. Iterative numerical methods for sampling from high dimensional gaussian distributions. *Statistics and Computing*, 23(4):501–521, 2013. 88

Elif Ayvali, Rangaprasad Arun Srivatsan, Long Wang, Rajarshi Roy, Nabil Simaan, and Howie Choset. Using bayesian optimization to guide probing of a flexible environment for simultaneous registration and stiffness mapping. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 931–936. IEEE, 2016. 102

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001. 10

Gábor Braun, Sebastian Pokutta, and Yao Xie. Info-greedy sequential adaptive compressed sensing. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):601–611, 2015. 71

Eric Brochu, Vlad M Cora, and Nando de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, 2010. 3, 43

Jason L Brown, Alison Cameron, Anne D Yoder, and Miguel Vences. A necessarily complex

model to explain the biogeography of the amphibians and reptiles of Madagascar. *Nat. Commun.*, 5:5046, January 2014. ISSN 2041-1723. 41

Noam Brown and Tuomas Sandholm. Safe and nested subgame solving for imperfect-information games. *arXiv preprint arXiv:1705.02955*, 2017. 104

Sébastien Bubeck. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Found. Trends Mach. Learn.*, 5(1):1–122, 2012. ISSN 1935-8237. doi: 10.1561/2200000024. 4

Sébastien Bubeck, Gilles Stoltz, Csaba Szepesvári, and Rémi Munos. Online optimization in X-armed bandits. In *Adv. Neural Inf. Process. Syst.*, pages 201–208, 2009. 10

Sébastien Bubeck, Nicolò Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012. 86

Emmanuel Candes and Terence Tao. The dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 2007. 71, 80

Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, 2008. 68

Alexandra Carpentier and Rmi Munos. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *AISTATS*, volume 22, pages 190–198, 2012. 68

Mattia Carpin, Stefano Rosati, Mohammad Emtiyaz Khan, and Bixio Rimoldi. Uavs using bayesian optimization to locate wifi devices. *arXiv preprint arXiv:1510.03592*, 2015. 68

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011. 85, 86

Siheng Chen, Aliaksei Sandryhaila, José MF Moura, and Jelena Kovacevic. Signal denoising on graphs via graph filtering. In *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pages 872–876. IEEE, 2014. 16

Edmond Chow and Yousef Saad. Preconditioned krylov subspace methods for sampling multivariate gaussian distributions. *SIAM Journal on Scientific Computing*, 36(2):A588–A608, 2014. 87

Emile Contal, Vianney Perchet, and Nicolas Vayatis. Gaussian process optimization with mutual information. In *31th Int. Conf. Mach. Learn.*, 2014. 29

Dennis D Cox and Susan John. Sdo: A statistical method for global optimization. *Multidiscip. Des. Optim. state art*, pages 315–329, 1997. 16

W Bruce Croft, Donald Metzler, and Trevor Strohmann. *Search engines*. Pearson Education, 2010. 2

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *COLT*, pages 355–366, 2008. 4, 10

Abhimanyu Das and David Kempe. Algorithms for subset selection in linear regression. In *Proc. 40th Annu. ACM Symp. Theory Comput.*, pages 45–54. ACM, 2008. 28

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks

on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3837–3845, 2016. 103

David L Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006. 68, 79, 81

Peter G. Doyle and J. Laurie Snell. *Random Walks and Electric Networks*, volume 22. Mathematical Association of America, 1 edition, 1984. ISBN 9780883850244. 19

D. J. Eisenstein, D. H. Weinberg, E. Agol, H. Aihara, C. Allende Prieto, and et al. SDSS-III: Massive spectroscopic surveys of the distant universe, the Milky Way, and extra-solar planetary systems. *The Astronomical Journal*, 142:72, September 2011. 41

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun):1079–1105, 2006. 85

Seth Flaxman, Andrew Gordon Wilson, Daniel B Neill, Hannes Nickisch, and Alexander J Smola. Fast kronecker inference in gaussian processes with non-gaussian likelihoods. In *International Conference on Machine Learning*, volume 2015, 2015. 7, 86, 89

S Friedland and S Gaubert. Submodular spectral functions of principal submatrices of a hermitian matrix, extensions and applications. *Linear Algebra Appl.*, 2011. 15, 105

Akshay Gadde and Antonio Ortega. A probabilistic interpretation of sampling theory of graph signals. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 3257–3261. IEEE, 2015. 16

Roman Garnett, Yamuna Krishnamurthy, Xuehan Xiong, Jeff Schneider, and Richard P Mann. Bayesian optimal active search and surveying. In *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, 2012. 1, 3, 4, 6, 9, 12, 13, 15, 43, 45, 49, 61, 96, 99

J. D. Gergonne. Application de la méthode des moindre quarrés a l'interpolation des suites. *Annales des Math Pures et Appl*, 1815. 67

John C Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 148–177, 1979. 4

Aditya Gopalan, Shie Mannor, and Yishay Mansour. Thompson sampling for complex online problems. In *ICML*, volume 14, pages 100–108, 2014. 86

Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause. Active learning for level set estimation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2013. xiii, 21, 22, 43, 57, 60

Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. 103

Jarvis D Haupt, Richard G Baraniuk, Rui M Castro, and Robert D Nowak. Compressive distilled sensing: Sparse recovery using adaptivity in compressive measurements. In *Signals, Systems and Computers*. IEEE, 2009. 68

Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(Jun):1809–1837, 2012. 3

Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive entropy search for multi-objective bayesian optimization. In *International Conference on Machine Learning*, pages 1492–1501, 2016. 3

José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, 2014. 68, 86

Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS, 1952. 86

Bruno Jedynak, Peter I Frazier, Raphael Sznitman, et al. Twenty questions with noise: Bayes optimal policies for entropy loss. *Journal of Applied Probability*, 49(1):114–136, 2012. 68, 71

Ming Ji and Jiawei Han. A variance minimization criterion to active learning on graphs. In *AISTAT*, 2012. 6, 13, 15, 23, 27, 31, 99

Pengsheng Ji and Jiashun Jin. Coauthorship and citation networks for statisticians. *The Annals of Applied Statistics*, 10(4):1779–1812, 2017. xiii, 10, 11

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998. 3, 68, 85

Kwang-Sung Jun and Robert Nowak. Graph-based active learning: A new look at expected error minimization. In *IEEE GlobalSIP Symposium on Non-Commutative Theory and Applications*. IEEE, 2016. 16

Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning (ICML)*, 2015. 85

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory*, pages 199–213. Springer, 2012. 78, 86

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 103

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proc. fortieth Annu. ACM Symp. Theory Comput.*, pages 681–690. ACM, 2008. 10

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008. 3, 6, 13, 14, 15, 21, 28, 31

O B Kroemer, R Detry, J Piater, and J Peters. Combining active learning and reactive control for robot grasping. *Rob. Auton. Syst.*, 58(9):1105–1116, September 2010. 43

John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Proba-

bilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001. 10, 38

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985. 78

Tor Lattimore and Csaba Szepesvari. The end of optimism? an asymptotic analysis of finite-armed linear bandits. *arXiv preprint arXiv:1610.04491*, 2016. 86, 95, 96

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition*. IEEE, 2006. 69, 72, 79

Richard B Lehoucq and Danny C Sorensen. Deflation techniques for an implicitly restarted arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications*, 17(4):789–821, 1996. 87

Steven Cheng-Xian Li and Benjamin M Marlin. A scalable end-to-end gaussian process adapter for irregularly sampled time series classification. In *Advances In Neural Information Processing Systems*, pages 1804–1812, 2016. 88

Qiang Liu, Alexander Ihler, and John Fisher. Boosting crowdsourcing with expert labels: Local vs. global effects. In *Information Fusion (Fusion), 2015 18th International Conference on*, pages 9–14. IEEE, 2015. 16

Andrea Locatelli, Maurilio Gutzeit, and Alexandra Carpentier. An optimal algorithm for the thresholding bandit problem. In *Proceedings of The 33rd International Conference on Machine Learning*, 2016. 55

Romy Lorenz, Ricardo Pio Monti, Inês R Violante, Christoforos Anagnostopoulos, Aldo A Faisal, Giovanni Montana, and Robert Leech. The automatic neuroscientist: A framework for optimizing experimental design with closed-loop real-time fmri. *NeuroImage*, 129:320–334, 2016. 102

Kian Hsiang Low, Jie Chen, John M Dolan, Steve Chien, and David R Thompson. Decentralized active robotic exploration and mapping for probabilistic field classification in environmental sensing. In *Proc. 11th Int. Conf. Auton. Agents Multiagent Syst. - Vol. 1*, AAMAS '12, pages 105–112, Richland, SC, 2012. 43

Yifei Ma, Roman Garnett, and Jeff Schneider. Σ-optimality in active learning on Gaussian random fields. In *NIPS*, 2013. 5

Yifei Ma, Roman Garnett, and Jeff Schneider. Active area search via Bayesian quadrature. In *Proc. 17th Int. Conf. Artif. Intell. Stat. (AISTATS 2014)*, 2014. 5, 7, 60

Yifei Ma, Tzu-Kuo Huang, and Jeff G Schneider. Active search and bandits on graphs using sigma-optimality. In *UAI*, pages 542–551, 2015a. 4, 5, 7, 68

Yifei Ma, Dougal Sutherland, Roman Garnett, and Jeff Schneider. Active pointillistic pattern search. 2015b. Share Two Lead Authors. 5

Yifei Ma, Roman Garnett, and Jeff Schneider. Active search for sparse signals with region sensing. In *AAAI Conference on Artificial Intelligence*, 2017a. 5

Yifei Ma, Roman Garnett, Jeff Schneider, and Andrew Gordon Wilson. Fast bayesian optimization via conjugate sampling. In *NIPS 2016 Workshop on Practical Bayesian Nonparametric*, 2017b. 5

Matthew L Malloy and Robert D Nowak. Near-optimal adaptive compressed sensing. *Information Theory, IEEE Transactions on*, 2014. 71, 73, 77, 79, 81

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. Scoring, term weighting and the vector space model. *Introduction to information retrieval*, 100:2–4, 2008. 2

Shawn Martin, W Michael Brown, Richard Klavans, and Kevin W Boyack. Openord: an open-source toolbox for large graph layout. In *IS&T/SPIE Electron. Imaging*, page 786806. International Society for Optics and Photonics, 2011. 22, 109

H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013. 2

Jonas Mockus. On bayesian methods for seeking the extremum. In *Optimization Techniques, IFIP Technical Conference, Novosibirsk, USSR, July 1-7, 1974*, pages 400–404, 1974. 3

D.C. Montgomery. *Design and Analysis of Experiments*. Wiley, 2012. 3

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functionsi. *Math. Program.*, 14(1):265–294, 1978. 15, 28

Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006. 11

Kirk A Nichols and Allison M Okamura. Methods to segment hard inclusions in soft tissue during autonomous robotic palpation. *IEEE Transactions on Robotics*, 31(2):344–354, 2015. 102

Srinivas Niranjan, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. 27th Int. Conf. Mach. Learn. (ICML 2010)*, 2010. 43, 85, 91

François Orieux, Olivier Féron, and J-F Giovannelli. Sampling high-dimensional gaussian distributions for general linear inverse problems. *IEEE Signal Processing Letters*, 19(5):251–254, 2012. 87, 88, 92, 94, 96, 97

Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2377–2386, 2016. 86

Michael A. Osborne, Roman Garnett, and Stephen J. Roberts. Gaussian processes for global optimization. In *Proceedings of the 3rd Learning and Intelligent Optimization Conference (LION 3)*, 2009. 43

Albert Parker and Colin Fox. Sampling gaussian distributions in krylov spaces with conjugate gradients. *SIAM Journal on Scientific Computing*, 34(3):B312–B334, 2012. 87, 90, 93, 95, 97

Eric Perlman, Randal Burns, Yi Li, and Charles Meneveau. Data exploration of turbulence

simulations using a database cluster. In *Proc. 2007 ACM/IEEE Conf. Supercomupting*, 2007. 63

Purnima Rajan, Weidong Han, Raphael Sznitman, Peter Frazier, and Bruno Jedynak. Bayesian multiple target localization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015. 68, 71

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Restricted eigenvalue properties for correlated gaussian designs. *The Journal of Machine Learning Research*, 2010. 80

Carl E Rasmussen and Zoubin Ghahramani. Bayesian monte carlo. In *Adv. Neural Inf. Process. Syst. 15 (NIPS 2002)*, 2003. 55

Carl Edward Rasmussen and Christopher K I Williams. *Gaussian processes for machine learning*. The MIT Press, 2006. ISBN 0-262-18253-X. 15, 20, 42, 86, 89

Herbert Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Sel. Pap.*, pages 169–177. Springer, 1985. 10

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *Journal of Machine Learning Research*, 2014. 86

Yousef Saad. *Iterative methods for sparse linear systems*. Siam, 2003. 89

Sushant Sachdeva and Nisheeth K Vishnoi. Faster algorithms via approximation theory. *Theoretical Computer Science*, 9(2):125–210, 2013. 86

Michael K Schneider and Alan S Willsky. A krylov subspace method for covariance approximation and simulation of random processes and fields. *Multidimensional systems and signal processing*, 14(4):295–318, 2003. 87

Sambu Seo, M. Wallat, T. Graepel, and K. Obermayer. Gaussian process regression: active data selection and test point rejection. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 3, pages 241–246 vol.3, 2000. 57

Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010. 1, 6, 10, 14, 15, 31, 43

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587): 484–489, 2016. 104

Daniel P Simpson, Ian W Turner, and Anthony N Pettitt. Fast sampling from a gaussian markov random field using krylov subspace approaches. 2008. 88, 101

Alexander J. Smola and Risi Kondor. Kernels and regularization on graphs. In *COLT/Kernel*, pages 144–158, 2003. 38

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012. 85

Archana Soni and Jarvis Haupt. On the fundamental limits of recovering tree sparse vectors from

noisy linear measurements. *Information Theory, IEEE Transactions on*, 2014. 76

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. International Conference on Machine Learning (ICML)*, 2010a. 21, 55

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. *Proceedings of International Conference on Machine Learning*, pages 1015–1022, 2010b. 4, 6, 14, 16

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *Inf. Theory, IEEE Trans.*, 58(5):3250–3265, 2012. 29, 30, 37, 108

Dougal J Sutherland, Liang Xiong, Barnabás Póczos, and Jeff Schneider. Kernels on sample sets via nonparametric divergence estimates, 2012. 63, 64

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998. 3

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science (80-.).*, 290(5500):2319–2323, 2000. 10, 34

Matthew Tesch, Jeff Schneider, and Howie Choset. Expensive function optimization with stochastic binary outcomes. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, May 2013. 43, 85

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. 85

United States Census Bureau. 2010 census, 2010. URL `http://www.census.gov/2010census/data/`. 62

Abhinav Valada, Christopher Tomaszewski, Balajee Kannan, Prasanna Velagapudi, George Kantor, and Paul Scerri. An intelligent approach to hysteresis compensation while sampling using a fleet of autonomous watercraft. In *Intelligent Robotics and Applications*, volume 7507 of *Lecture Notes in Computer Science*. 2012. 41, 42, 59

Michal Valko, Rémi Munos, Branislav Kveton, and Tomáš Kocák. Spectral bandits for smooth graph functions. In *31th Int. Conf. Mach. Learn.*, 2014. xiii, 14, 16, 21, 29, 30, 37

Hastagiri P Vanchinathan, Andreas Marfurt, Charles-Antoine Robelin, Donald Kossmann, and Andreas Krause. Adaptively selecting valuable diverse sets via Gaussian processes and submodularity. In *NIPS Workshop on Discrete and Combinatorial Problems in Machine Learning (DISCML) 2013: Theory and Applications*, 2013. xiii, 3, 9, 12, 14, 15, 26, 29, 30, 37

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700, 2015. 104

Martin J Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using-constrained quadratic programming (lasso). *Information Theory, IEEE Transactions on*, 2009. 68, 71, 80

David A Walker. Suppressor variable (s) importance within a regression model: an example of

salary compression from career services. *J. Coll. Stud. Dev.*, 44(1):127–133, 2003. 13, 28

Dilin Wang, John Fisher III, and Qiang Liu. Efficient observation selection in probabilistic graphical models using bayesian lower bounds. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 755–764. AUAI Press, 2016. 16

Xuezhi Wang, Roman Garnett, and Jeff Schneider. Active search on graphs. In *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. data Min.*, pages 731–738. ACM, 2013. 1, 3, 4, 9, 12, 15, 35, 36, 37, 96

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1775–1784, 2015. 7, 86

Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems*, pages 2483–2491, 2011. 68

Jie Zhong, Yijun Huang, and Ji Liu. Asynchronous parallel empirical variance guided algorithms for the thresholding bandit problem. *arXiv preprint arXiv:1704.04567*, 2017. 55

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 20, page 912, 2003a. 6, 11, 12, 16

Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 Work. Contin. from labeled to unlabeled data Mach. Learn. data Min.*, pages 58–65, 2003b. 6, 14