NEW INSIGHTS ON THE POWER OF ACTIVE LEARNING

A Dissertation
Presented to
The Academic Faculty

by

Christopher Grant Berlind

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science

Georgia Institute of Technology
August 2015

NEW INSIGHTS ON THE POWER OF ACTIVE LEARNING

Approved by:

Maria-Florina Balcan, Co-advisor
School of Computer Science
Carnegie Mellon University

Le Song, Co-advisor
School of Computer Science and Engineering
Georgia Institute of Technology

Santosh Vempala
School of Computer Science
Georgia Institute of Technology

Charles L. Isbell, Jr.
School of Interactive Computing
Georgia Institute of Technology

Avrim Blum
School of Computer Science
Carnegie Mellon University

Date Approved: June 24, 2015

To Mom and Dad

ACKNOWLEDGEMENTS

The circumstances allowing for the creation of this thesis were shaped and molded by so many influential people that I would be greatly remiss not to express my sincerest gratitude to as many of them as I can. At the same time, this virtually guarantees I will neglect to mention someone important, and for that I apologize.

First and foremost, I would like to thank Nina Balcan. As my advisor and mentor for the last four years, Nina's impeccable vision and endless enthusiasm have been crucial to my success. She has kept me on track whenever I start to falter, and she always has interesting and exciting problems for me to explore. Her ability to maintain a top-notch work ethic, advise students, and raise a family at the same time (while doing all of them well!) is truly astounding, and I greatly admire her for that.

I am also very thankful to my co-advisor, Le Song. Discussions with Le in any situation are always enlightening, and I have learned a great deal from him. I am also privileged to have Santosh Vempala, Charles Isbell, and Avrim Blum making up the remainder of my committee. Their insights have always been valuable and their suggestions helpful, and I thank them for serving on my committee.

I owe an enormous debt of gratitude to all my co-authors as well. Steven Ehrlich, Yingyu Liang, Ruth Urner, Kaushik Patnaik, and Emma Cohen have all been a pleasure to work with and I could not ask for a better team. In addition to my co-authors, Ying Xiao, Shang-Tse Chen, and Ben Cousins have always been willing to read early drafts, listen to practice talks, and ask really good questions that only serve to make my work better. I can hardly begin to describe how intelligent, thoughtful, hardworking, and forgiving everyone has been.

I must also thank the many influential teachers and mentors I have had throughout

the course of my education. In particular, I thank Peter Barbella, who taught me that I am a mathematician; Mike Vanier, who taught me that I am also a computer scientist; Yaser Abu-Mostafa, who inspired my love for machine learning; and Erik Winfree, who taught me what it means to do great research.

I would not have made it through the last four years in one piece if it was not for so many friends and lab-mates relentlessly distracting me from work. Special thanks to Emma, Ying, Sarah, Sara, and Eric for always finding time for chamber music; to Ying for getting me up early for tennis; to Nolan, Robert K., Steven, Ben, and Emma for the card games; to Brendan for the book recommendations I never took; and to Hank, Prateek, Robert P., and others for helping me study for my second bachelor's. I also want to thank the happy hour, trivia, and frisbee crews for helping to keep me somewhat sane. My biggest distraction of all turned out to be an old friend convincing me to start a company with him. I thank David Lindsay for being patient with me while I finish my Ph.D.

I would like to express my deepest thanks to my family for making me into the person I am today. My parents, Linda and Brian, have been the greatest parents imaginable, providing me with enriching opportunities at every turn. They and my big sister Stephanie have always been my role models and friends and continue to support me in everything I do (even though I don't call them enough).

Finally, I need to thank my amazing fiancée, Emma Cohen. Emma is truly exceptional in so many ways and has made numerous contributions, both technical and emotional, to the completion of this thesis. Thank you, Emma, for your love, friendship, and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Supervised machine learning is the process of algorithmically learning how to make future predictions by training on labeled examples of past occurrences. While traditionally a learning algorithm has access to a large corpus of labeled examples, the recent proliferation of data made possible by modern computing power and the Internet has made unlabeled data much easier to come by than accompanying labels. For example, billions of images are readily available for download on the Internet, but annotations of the objects present in an image are much more difficult to acquire.

Two main methods have been proposed by the machine learning community for taking advantage of relatively low-cost unlabeled examples in an effort to reduce the number of expensive labeled examples needed for learning. One method is semi-supervised learning, which includes a large quantity of unlabeled examples into the training data in addition to a smaller number of labeled examples. Another is active learning, in which the algorithm itself can select which examples it would like labeled out of a large pool of unlabeled examples. Prior research on active learning has focused almost entirely on the issue of reducing labeling effort (over that of passive learning) through intelligent querying strategies.

In this dissertation, we demonstrate that the power to make adaptive label queries has benefits beyond reducing labeling effort over passive learning. We develop and explore several novel methods for active learning that exemplify these new capabilities. Some of these methods use active learning for a non-standard purpose, such as computational speedup, structure discovery, and domain adaptation. Others successfully apply active learning in situations where prior results have given evidence of its ineffectiveness.

Specifically, we first give an active algorithm for learning disjunctions that is able to overcome a computational intractability present in the semi-supervised version of the same problem. This is the first known example of the computational advantages of active learning. Next, we investigate using active learning to determine structural properties (margins) of the data-generating distribution that can further improve learning rates. This is in contrast to most active learning algorithms which either assume or ignore structure rather than seeking to identify and exploit it. We then give an active nearest neighbors algorithm for domain adaptation, the task of learning a predictor for some target domain using mostly examples from a different source domain. This is the first formal analysis of the generalization and query behavior of an active domain adaptation algorithm. Finally, we show a situation where active learning can outperform passive learning on very noisy data, circumventing prior results that active learning cannot have a significant advantage over passive learning in high-noise regimes.

CHAPTER I

INTRODUCTION

Active learning involves machine learning algorithms that make adaptive label queries during the learning process. The thesis of this work is that the power to make adaptive label queries has benefits beyond those traditionally considered to be within the scope of active learning. While traditionally the purpose of the query ability is to reduce labeling effort, especially in low-noise scenarios, we present several novel uses of active learning that add significant breadth to its repertoire. These new capabilities include speeding up computation time over semi-supervised learning, discovering and exploiting margin structure in data, adapting to a changing data distribution, and improving prediction accuracy over passive learning in the presence of very noisy data.

1.1 Learning from Labeled and Unlabeled Data

Supervised machine learning is the process of algorithmically learning how to make future predictions by training on examples of past occurrences. Training examples are of the form (x, y) where x is an *instance* (e.g. an image or email) and y is a *label* (e.g. the name of an item in the image or the classification of an email as "spam" or "not spam"). Traditionally, a learning algorithm has access to a large corpus of such examples, all of them labeled. While this paradigm is appropriate for many applications, the recent proliferation of data made possible by modern computing power and the Internet has changed the relative availability of different types of data. For example, billions of images are readily available for download on the Internet, but accompanying annotations of objects present in an image are much more difficult to come by. Similarly, large numbers of emails can be found in just about anyone's inbox, but the task of manually labeling each email is daunting at best. As a result, modern machine learning frequently adheres to a new paradigm in which labeled examples are expensive as usual, but unlabeled examples are cheap or even free.

Several methods have been proposed by the machine learning community for taking advantage of relatively low-cost unlabeled examples in an effort to reduce the number of expensive labeled examples needed for learning. One method is *semi-supervised learning* which includes a large quantity of unlabeled examples into the training data in addition to a (usually smaller) number of labeled examples. At a high level, we may expect unlabeled data to improve prediction performance because it gives the learner information about what kinds of instances to expect (the data-generating distribution), and this in turn allows the learner to focus its search for a good predictor to those predictors which "make sense" given the kinds of examples it expects to see.

Another method for learning with few labeled examples, and the main focus of this thesis, is *active learning*. Active learning extends the classical paradigm of machine learning by starting with access to a large supply of unlabeled data and intelligently choosing which unlabeled examples should be labeled. The learner sequentially makes label queries one at a time and can see each resulting label before proceeding to make a query in the next iteration (see Section 2.1 for the formal definition). Applications of active learning therefore require methods for acquiring labels. Often these methods come in the form of feedback from experts (such as asking a doctor or technician to label a medical image) or non-experts (such as Amazon's Mechanical Turk or Zooniverse projects like Galaxy Zoo), but labels can also come from expensive procedures like chemistry experiments or computer simulations.

Research on active learning has generally focused on demonstrating that the ability to query the label of any unlabeled training example allows active learning algorithms to achieve improved *label complexity* over passive methods (see Section 2.2 for a more comprehensive review). In other words, most works attempt to show that active learning can achieve the same prediction accuracy while using fewer labels than methods which are given randomly selected labeled examples. This is true of both applied and theoretical work. Many methods have been proposed for selecting examples to query: some are information theoretic, some are based on probabilistic models, some rely on Bayesian priors, and still others are geometric. Despite the recent theoretical advancements in active learning, relatively few active learning methods have theoretical guarantees, and of those that do, few are simultaneously practical.

1.2 Contributions

In this dissertation, we demonstrate that the power to make adaptive label queries has benefits beyond reducing labeling effort over passive learning. We develop and explore several novel methods for active learning that exemplify these new capabilities. Some of these methods use active learning for a non-standard purpose, such as computational speedup, structure discovery, and domain adaptation. Others successfully apply active learning in situations—such as in the presence of very noisy data—where prior results have given evidence of its ineffectiveness.

Our focus throughout is on theoretically sound algorithms with provable guarantees. However, we simultaneously demonstrate the practicality of our methods through experiments on both synthetic and real data for the majority of our contributions.

Here we give an overview of our main contributions.

1.2.1 Actively Learning Disjunctions

In a seminal work on the theory of semi-supervised learning, Balcan and Blum [13] formalize the idea that unlabeled data is useful because for many learning problems, the natural regularities of the problem involve not only the form of the function being learned but also how this function relates to the distribution of data. For example, a natural assumption in linear classification is that the separating hyperplane passes through a low-density region rather than cutting through the middle of a dense cluster. Unlabeled data is useful in this context because in principle, it allows one to reduce the search space from the entire set of hypotheses down to the set of *compatible* hypotheses, those satisfying the regularity condition with respect to the unlabeled data. While such insights have been exploited for deriving a variety of sample complexity results [40, 63, 84, 13], the algorithmic problems involved in semi-supervised learning become much more challenging. The scarcity of efficient semi-supervised learning algorithms was noted in [24], where several open problems were posed.

One of these open problems was to design an algorithm to learn the class of two-sided disjunctions (the class of monotone disjunctions under a natural notion of compatibility) in the semi-supervised model. We design two semi-supervised algorithms for this problem under an additional restriction on the data distribution. One outputs a disjunction that is both consistent with the labeled data and compatible with the unlabeled examples, but it runs in polynomial time only for a subset of possible target functions. The other is always efficient, but it is not a proper learning algorithm (it does not always output a disjunction). In some sense, these restrictions on the semi-supervised algorithms are necessary, as we also show that the problem of finding a consistent and compatible hypothesis in the semi-supervised model is NP-hard.

However, by taking advantage of the additional power available in active label queries, we are able to design an efficient active learning algorithm that outputs a consistent and compatible disjunction without any restrictions on the data distribution. This represents the first known example of how active learning can be used to avert computational difficulties present in semi-supervised learning.

1.2.2 Passive and Active Learning with Large $L_q L_p$ Margins

The notion of "margin" arises naturally in many areas of machine learning. Margins have long been used to motivate the design of algorithms [33, 14], to give sufficient conditions for fast learning rates [18, 70], and to explain unexpected behavior of algorithms in practice [90]. In the context of learning linear separators, we can define an $L_q L_p$ margin, where L_q and L_p are dual norms placed on the weight vector space and instance space respectively. Prior work on passive algorithms and generalization bounds for learning with large $L_q L_p$ margins has only been able to demonstrate the benefits of large margins in the $p >= 2$ case. Margins ($L_2 L_2$ in particular) have been used in active learning to guide the design and analysis of algorithms, but little is known about active algorithms designed to exploit large margins present in training data.

We first give a bound on the generalization error of passively learning linear separators with large $L_q L_p$ margins for any finite $p >= 1$. The bound extends and improves upon previous results and leads to a simple data-dependent sufficient condition for fast learning rates. We also give examples showing the relative power of different types of margins in different settings. This leads to a setting in which making use of margins with $p < 2$ has a provable advantage over margins with $p >= 2$. Both parts include experimental results on real data showing the relevance of our theoretical results in practice.

We then show how active learning can be used to discover structure in data. We modify traditional margin-based active learning algorithms by replacing the usual margin with an $L_q L_p$ margin. We first show that knowing the correct value of p enables these algorithms to outperform traditional margin-based AL approaches. Our main contribution here is the use of active learning to automatically determine the appropriate value of p for optimal margin-based active learning. We do this by giving an algorithm that estimates from small batches of labeled examples the sufficient condition for fast passive learning rates. Estimating this quantity at early stages allows the algorithm to query more intelligently in future stages. We show that this algorithm indeed has benefits over other methods, and effectively uses queries to learn the structure of the data.

1.2.3 Active Learning for Domain Adaptation

Traditional machine learning paradigms operate under the unrealistic assumption that the data generating process remains stable; that is, training and test data are assumed to be from the same task. The main challenge in domain adaptation (or transfer learning) [80] is to develop learning algorithms that adapt to and perform well in changing environments. In a common model for domain adaptation, the learner receives large amounts of labeled data from a *source* distribution and unlabeled data from the actual *target* distribution (and possibly a small amount of labeled data from the target task as well). The goal of the learner is to output a good model for the target task.

Designing methods for this scenario that are statistically consistent with respect to the target task is challenging, even in the so-called covariate shift setting [96, 98], where the marginal distribution over the covariates changes but the regression functions (the labeling rules) of the source and target distributions are identical. Performance guarantees in the literature usually involve an extra additive term that measures the difference between source and target tasks (that is, the loss does not converge to the optimal target error) [19, 75], or they rely on strong assumptions, such as the target support being a subset of the source support and the density ratio between source and target being bounded from below [99, 20]. Generally, the case where the target is partly supported in regions that are not covered by the source, is considered to be particularly challenging [32].

We provide the first formal demonstration that active learning yields a way to address these challenges[1]. We propose a non-parametric algorithm, ANDA, that combines an active nearest neighbor querying strategy with nearest neighbor prediction. ANDA chooses target examples to query for labels according to how many source examples lie in a k'-nearest neighbor ball around them (this serves as an indication for how well the area of a point is covered by the source). ANDA then predicts with a k-nearest neighbor classifier on the combined source and target labeled data.

We provide both an analysis of finite sample convergence rates of the resulting classifier and an analysis of ANDA's querying behavior. Remarkably, the bounds on the expected loss we provide do not depend on the source/target relatedness. The convergence rates only depend on the size of the input unlabeled target sample. The number of queries that the algorithm will make, however, does depend the closeness of the involved tasks. ANDA will automatically make more or less queries to the target sample depending on how well the target is "covered" by the source, that is depending on whether the source provides sufficiently informative examples or not.

ANDA's intelligent querying behavior and its advantages are further demonstrated by our visualizations and experiments. We visually illustrate ANDA's query strategy and show empirically that ANDA successfully corrects for dataset bias in a challenging image classification task.

1.2.4 Sensor Consensus Game for High-Noise Active Learning

To date, research on active learning has focused only on the single-agent setting and primarily on low-noise scenarios, although several notable works on active learning in the presence of noise can be found as well [48, 9, 5]. However, it is known that the effectiveness of active learning quickly degrades as noise rates become high [64, 22]. In this work, we introduce and analyze a novel setting where label information is held by

[1] While the idea of incorporating active learning into domain adaptation strategies has recently received some attention in the machine learning community [30, 29, 89], to the best of our knowledge, there has not been any formal analysis of the possibilities of incorporating active learning to facilitate being adaptive to distribution changes.

highly-noisy low-power sensor agents. These agents are each noisily measuring some quantity which assigns them an initial (binary) state and can communicate locally with each other. A central agent far from the sensors can query an agent for its state in an effort to learn a spacial boundary between the two states. Learning this boundary directly would require prohibitively many queries due to the high noise rate in the system. We show how by first using simple game-theoretic dynamics among the agents we can quickly approximately denoise the system. This allows us to exploit the power of active learning (especially, recent advances in agnostic active learning), leading to efficient learning from only a small number of expensive queries.

1.3 Overview of the Dissertation

This dissertation is organized as follows. We first establish the traditionally accepted uses of active learning, and then we provide evidence for four novel capabilities of active learning.

Chapter 2 begins with a formal definition of the active learning models referenced herein. We then review the major lines of research in the field that have arisen since its inception. This chapter serves to establish that research in active learning has typically focused on its ability to reduce label complexity over passive learning in settings with little or no noise.

In Chapter 3 we describe efficient semi-supervised and active algorithms for learning two-sided disjunctions. We also demonstrate that the problem of finding a consistent, compatible disjunction in the semi-supervised setting is NP-hard, while our active learning algorithm is able to accomplish the same task efficiently. This shows how active learning can be used to circumvent computational difficulties that arise in semi-supervised learning. This chapter is based on work that appears in ICML 2013 [11].

Chapter 4 contains new generalization guarantees for passive learning with large L_qL_p margins and an example proving the usefulness of the $L_\infty L_1$ margin. We also give experiments showing not only that our theory for passive learning carries over to practical settings, but also that active learning can be used effectively to discover and exploit margin structure in the data. A portion of this chapter is based on work that appears in AISTATS 2014 [10].

In Chapter 5 we describe our active nearest neighbors algorithm for domain adaptation and give theoretical analyses of its generalization performance and querying behavior. Our experiments show that our algorithm can successfully correct for dataset bias in image classification. Appendix A contains some additional proofs of the the theorems in this chapter. This chapter is largely based on work that appears in ICML 2015 [21].

Finally, in Chapter 6 we propose a novel setting for learning from many noisy low-power agents. We then describes our multi-agent scheme for denoising the system based on a consensus game and prove its effectiveness in several settings. The denoising scheme allows active learning to achieve exponential improvement over passive learning despite the high noise. This chapter is based on work that appears in NIPS 2014 [16].

CHAPTER II

ACTIVE LEARNING

In this chapter we first formally define the active learning setting and then give a brief survey of the active learning literature, focusing on the advancements relevant to this thesis. More detailed surveys can be found in [92], which contains full coverage of the topic from a more empirical perspective, and [57], which includes theory only and focuses on disagreement-based methods.

2.1 Formal Setup

We begin by formally defining the PAC (*probably approximately correct*) learning model [104] for passive supervised learning on which active learning theory is based. In the PAC model, we have an instance space X, label space $Y = \{-1, 1\}$, and a fixed but unknown distribution[1] P_X over X. Instances x are drawn independently from P_X and associated labels y are given by a fixed target concept $f^* : X \to Y$, where $f^* \in C$, the concept class. A learning algorithm is given a concept class and a set of n labeled examples $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and asked to efficiently produce a hypothesis $h : X \to Y$ such that its error rate $\mathsf{err}(h) = \Pr_{x \sim P_X}[h(x) \neq f^*(x)]$ is small (that is, h will rarely make mistakes on future labeled examples formed in the same manner). Specifically, the algorithm is given an accuracy parameter ϵ and confidence parameter δ and is asked to produce in polynomial time, with probability at least $1-\delta$, a hypothesis h with $\mathsf{err}(h) < \epsilon$. The sample complexity of the algorithm (given ϵ and δ) is the minimum number of examples n such that for any $f^* \in C$, the algorithm will output a hypothesis h with $\mathsf{err}(h) < \epsilon$ with probability at least $1 - \delta$.

[1] In later chapters, we will frequently use the notation D instead of P_X when there is no need to distinguish it from other probability distributions.

The PAC model defined above is completely noise-free. That is, there exists a function f^* with $\mathsf{err}(f^*) = 0$ and the algorithm is given a description of a set C known to contain f^*. We will refer to this setting as the realizable case. This model can be extended to include learning in the presence of noise (the non-realizable case) by assuming the examples (x, y) are independently drawn from a fixed but unknown distribution P over $X \times Y$. Now the error rate is given by $\mathsf{err}(h) = \Pr_{(x,y)\sim P}[h(x) \neq y]$. Alternatively (and equivalently), we could draw the instances x from a marginal distribution P_X over X and the corresponding labels y will be drawn from a binomial distribution determined by the regression function $\eta(x) = \Pr[y = 1|x]$. This is just a re-parameterization of the distribution P into P_X and η.

In this noisy setting, there may be no function $f* \in C$ such that $\mathsf{err}(f^*) = 0$, so it now may be impossible to find a hypothesis with error rate less than ϵ for any $\epsilon > 0$. There are two ways in which we can adjust the learner's goal to take this into account. The nonparametric approach is to require that the learner outputs a hypothesis that can predict nearly as well (within error ϵ) as the best possible predictor for the data generating distribution. This predictor, known as the Bayes classifier, is defined as $f^* = \text{argmin}_{f:X\to Y} \mathsf{err}(f)$ and is known to be given by $f^*(x) = 2\mathbb{1}[\eta(x) > 1/2] - 1$. The quantity $\mathsf{err}(h) - \mathsf{err}(f^*)$ is known as the excess error (or excess risk).

An alternative is a parametric approach known as agnostic learning [66]. In this model, the algorithm is only expected to produce a hypothesis that can compete with the best predictor in the given concept class. That is, given concept class C and accuracy parameter ϵ, an agnostic learner must, with high probability, output h such that $\mathsf{err}(h) < \min_{f\in C} \mathsf{err}(f) + \epsilon$. Finding polynomial time agnostic learners is often a very challenging problem, even for concept classes that are straightforward to learn efficiently in the PAC model.

Active learning can be formalized as follows. The same set S of labeled examples is drawn as defined above, but the labels $L = \{y_1, \ldots, y_n\}$ are separated from the unlabeled examples $U = \{x_1, \ldots, x_n\}$. The active learner receives the set U of unlabeled examples and is given access to a label oracle that returns the (possibly noisy) label y_i when given an instance x_i. The active learning protocol then proceeds in rounds, where in each round the algorithm can perform computations on the unlabeled data and choose a single unlabeled example to send to the oracle for labeling. After some number of rounds (possibly given to the learner in advance in the form of a label budget) the learner outputs a hypothesis h that should have small error with high probability.

Active learning is distinct from a related query model for learning known as the *membership query* model [2]. In this model, the learner is not given unlabeled examples upfront but is instead expected to synthesize an unlabeled example on its own each round and ask an oracle for the label. The major drawback of membership query learning is that for many real-world tasks, such as image classification and natural language processing, membership query algorithms produce instances for labeling that have no natural semantic meaning to human labelers.

2.2 Literature Review

Active learning originated in the early 1990s as a method for learning more accurately using fewer labeled examples. In the modern paradigm of unlabeled examples being relatively inexpensive compared to the cost of labels, this mode of learning is especially relevant.

In one of the earliest works on active learning, Cohn, Atlas, and Ladner [4] give a simple and general approach for active learning in the realizable case that became the basis for a line of work known as *disagreement-based* active learning. Their idea, stemming from the earlier work of Mitchell [77] on passive learning, is to maintain two sets known as the version space and region of disagreement, respectively. The version space $V \subseteq C$ contains all hypotheses that are consistent with all labeled examples seen so far, while the region of disagreement $R \subseteq X$ contains all unlabeled examples which would be labeled differently by two hypotheses in V. The query strategy is simply to iterate through the unlabeled examples and make a query if the example is in R, updating V and R after each query. The authors gave a method for implementing this strategy to train neural networks on relatively simple concept classes, but they gave no characterization of convergence rates and did not address the computational challenges inherent in maintaining these sets or how to deal with the presence of noise.

Since the publication of this seminal work, many strategies have been proposed for choosing which examples to query. Most of these methods optimize an objective function over the unlabeled examples in each round to select the next query point. For example MacKay [74] proposes several information theoretic objective functions that quantify the informativeness of each example based on a probabilistic model of the data. Roy and McCallum [87] employ a Bayesian assumption to directly minimize expected future error. Tong and Koller [103] propose several methods for selecting examples based on a geometric interpretation of support vector machines [33]. All of these works demonstrate the efficacy of their methods by testing them experimentally on real-world data. The empirical advantage of active learning over passive learning in terms of the trade-off between generalization error and labeled examples is made very clear.

Perhaps the earliest theoretical work on active learning is the “query by committee” algorithm of Freund et al. [44]. This is the first example of a formal analysis of query complexity and the first proven example of an algorithm with exponential improvement over passive learning for non-trivial cases. The analysis is for the realizable case and depends on having a correct Bayesian prior. The bulk of the modern work on active learning theory began with that of Dasgupta [35, 36] which also gave a query complexity analysis in the realizable case, now in a non-Bayesian setting.

Balcan et al. [12] developed the first active learning algorithm proven to satisfy the stringent requirements of the agnostic model. The algorithm is not computationally efficient in general, but it does have exponential improvement over passive learning in some natural settings (such as learning linear separators over the uniform distribution on the unit sphere). This work led to further analyses of the query complexity of active learning algorithms by Hanneke [55] and Koltchinskii [69] among others. Hanneke's work introduced a parameter known as the *disagreement coefficient* which characterizes the rate of convergence of disagreement-based methods. The disagreement coefficient quantifies the relationship between the growth rates of the version space and region of disagreement. Later works on agnostic active learning include [41, 23, 15, 5, 109]. In a very recent result, Huang et al. [59] give a new efficient agnostic active learning algorithm that is both general and aggressive.

Several other works have shown the power of active learning for reducing labeling effort. Tong & Koller [102] show that active queries can reduce label complexity over random sampling when learning structure in Bayesian networks. Hanneke [56] gives a general strategy for turning any passive learning algorithm into an active one with reduced label complexity. Gonen et al. [49] give an alternative aggressive approach for actively learning linear separators. Sabato & Munos [88] give an active learning algorithm for linear regression that provably improves over passive learning.

Despite the multitude of positive results for active learning, there are limits to its capabilities. One of the first negative results was a lower bound proved by Kääriäinen [64] which showed that in the agnostic setting with overall noise rate $\eta > 0$, the label complexity of active learning is $\Omega((\eta^2/\epsilon^2)\log(1/\delta))$. In other words, the dependency on the accuracy parameter ϵ is the same as that of passive learning. This does not preclude active learning from having exponential label complexity improvement settings with very low noise, but it does indicate that when the noise rate is significantly greater than the desired accuracy, active learning can achieve no better than a constant improvement over passive learning. This lower bound was improved to $\Omega(d\eta^2/\epsilon^2)$, where d is the VC-dimension of the hypothesis class, by Beygelzimer et al. [22].

CHAPTER III

ACTIVELY LEARNING DISJUNCTIONS

In this chapter we provide efficient algorithms with nearly optimal sample complexity for semi-supervised and active learning of disjunctions under a natural regularity assumption introduced by Balcan & Blum [13]. In particular we consider the so called two-sided disjunctions setting, where we assume that the target function is a monotone disjunction satisfying a margin-like regularity assumption. In the simplest case resolved in [13], the notion of "margin" is as follows: every variable is either a positive indicator for the target function (i.e., the true label of any example containing that variable is positive) or a negative indicator (i.e., the true label of any example containing that variable is negative), and no example contains both positive and negative indicators. In this work, we consider the much more challenging setting left open by Blum & Balcan [24] where *non-indicators* or *irrelevant variables*, i.e., variables that appear in both positive and negative examples, are also present.

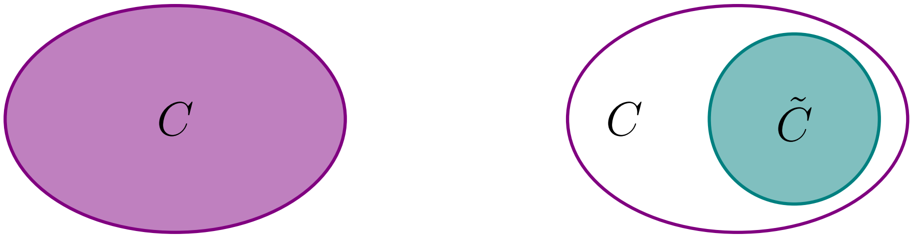
One practical motivation for using two-sided disjunctions is the problem of text classification. If each instance is a document in bag-of-words representation (each feature is an indicator variable for some word in the dictionary appearing in the given document) then our regularity assumption is satisfied when some subset of dictionary words (positive indicators) appear only in documents of the first class, another subset of words (negative indicators) appear only in the second class, and the remaining words (non-indicators) may appear in documents in either class. The assumption is similar to the separability assumption used in the topic modeling literature in which every topic is assumed to have an *anchor word* that only appears in documents of that topic [3].

In the semi-supervised learning setting, we present an algorithm that finds a consistent hypothesis that furthermore is compatible (in the sense that it satisfies our regularity assumption). This algorithm is proper (it outputs a disjunction), has near-optimal labeled data sample complexity provided that each irrelevant variable appears with non-negligible probability, and it is efficient when the number of irrelevant variables is $O(\log n)$. We next present a non-proper algorithm that PAC learns two-sided disjunctions with nearly the same sample complexity and whose running time is polynomial for any k. We also prove that, in general, it is NP-hard to find a consistent and compatible two-sided disjunction in the semi-supervised setting, which gives some justification for why our semi-supervised algorithms have additional dependencies (one is not proper, one is only efficient for a subclass of problems, and both depend on the minimum probability of non-indicators appearing).

In the active learning setting, we present an efficient proper active learning algorithm for two-sided disjunctions. This algorithm outputs a consistent, compatible two-sided disjunction, with sample complexity linear in the number of irrelevant variables and independent of the probability of irrelevant variables appearing, the quantity that appears in both bounds in the semi-supervised setting. Combined with our NP-hardness result for the semi-supervised setting, this shows that the active query ability allows the learner to overcome a computational difficulty. This is the first known example of such a benefit for active learning and our first example of how active learning can be used for something other than reducing label complexity over passive learning.

3.1 Related Work

Conceptually, what makes unlabeled data useful in the semi-supervised learning context [13, 112], is that for many learning problems, the natural regularities of the problem involve not only the form of the function being learned but also how this



(a) All hypotheses

(b) Highly compatible hypotheses

Figure 1: Compatibility assumptions shrink the effective hypothesis space, reducing label complexity from $O(\frac{1}{\epsilon}\log|C|)$ examples to $O(\frac{1}{\epsilon}\log|\tilde{C}|)$.

function relates to the distribution of data; for example, that it partitions data by a wide margin as in Transductive SVM [62] or that data contains redundant sufficient information as in Co-training [25]. Unlabeled data is useful in this context because it allows one to reduce the search space from the whole set of hypotheses, down to the set of hypotheses satisfying the regularity condition with respect to the underlying distribution (see Figure 1). Such insights have been exploited for deriving a variety of sample complexity results [40, 63, 84, 13]. However, while in principle semi-supervised learning can provide benefits over fully supervised learning [13, 112], the corresponding algorithmic problems become much more challenging. As a consequence there has been a scarcity of efficient semi-supervised learning algorithms.

While several semi-supervised learning methods have been introduced [28, 113, 62], much of the theoretical work has focused either on sample complexity (e.g., [40, 63, 84]) or on providing polynomial time algorithms with error bounds for surrogate losses only (e.g., [85]). The few existing results with guarantees on the classification error loss hold under very stringent conditions about the underlying data distribution (e.g., independence given the label [25]). In contrast, we provide (PAC-style) polynomial time algorithms for learning disjunctions with general guarantees on the classification error loss.

We note that while a lot of the research on active learning [36, 12, 53, 54, 41, 23, 69] has *not* made an explicit regularity assumption as in semi-supervised learning, this

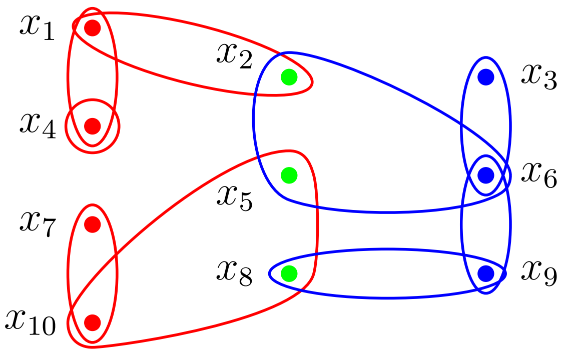


Figure 2: A schematic diagram of the two-sided disjunction defined by $h_+(x) = x_1 \vee x_4 \vee x_7 \vee x_{10}$ and $h_-(x) = x_3 \vee x_6 \vee x_9$. Vertices represent variables (features) and hyperedges represent examples (the variables included in the edge are set to 1 in the example). Edge color indicates the label of the example while vertex color represents the indicator type of the variable (red for positive, blue for negative, and green for non-indicator).

is an interesting direction to study. As our results reveal, active learning could help overcome computational hardness limitations over (semi-supervised) passive learning in these settings.

3.2 Preliminaries

Let $X = \{0,1\}^n$ be the instance space, $Y = \{-1, 1\}$ be the label set, and D denote any fixed probability distribution over X. Following [13], a two-sided disjunction h is defined as a pair of monotone disjunctions[1] (h_+, h_-) such that $h_+(x) = -h_-(x)$ for all $x \sim D$, and h_+ is used for classification. Let the concept class C be the set of all pairs[2] of monotone disjunctions and for any hypothesis $h = (h_+, h_-) \in C$, define $h(x) = h_+(x)$.

For a two-sided disjunction (h_+, h_-), variables included in h_+ are the *positive indicators*, and variables in h_- are *negative indicators*. Variables appearing neither in h_+ nor in h_- are called *non-indicators*, as the value of any such variable has no

[1] Recall that a monotone disjunction is an OR function of positive literals only, e.g. $h(x) = x_1 \vee x_3 \vee x_4$.

[2] Although we are actually interested in learning a single monotone disjunction, we need to associate each disjunction with a second disjunction in order to test compatibility.

effect on the label of any example. To simplify the discussion, we will often identify binary strings in $X = \{0,1\}^n$ with subsets of the variables $V = \{x_1, \ldots, x_n\}$. In other words, we say an example x contains x_i if the i-th coordinate of x is set to 1. This allows us to speak of variables "appearing in" or "being present in" examples rather than variables being set to 1. We will use similar language when referring to hypotheses, so that a two-sided disjunction $h = (h_+, h_-)$ consists of a set h_+ of positive indicators and a set h_- of negative indicators (which completely determine a third set of non-indicators). See Figure 2 for a schematic of a two-sided disjunction that will be used an example throughout this chapter.

In the semi-supervised learning setting, we will assume that both labeled examples L and unlabeled examples U are drawn i.i.d. from D and that examples in L are labeled by the target concept h^*, where h^* is a two-sided disjunction with at most k non-indicators. We will let $|L| = m_l$ and $|U| = m_u$; both m_l and m_u will be polynomial throughout this chapter. In the active setting, the algorithm first receives a polynomially sized unlabeled sample U and it can adaptively ask for the label $\ell(x) = h^*(x)$ of any example $x \in U$.

The generalization error of a hypothesis h is given by $\mathsf{err}(h) = \Pr_{x \sim D}[h(x) \neq h^*(x)]$, the probability of h misclassifying a random example drawn from D. For a set L of labeled examples, the empirical error is given by $\mathsf{err}_L(h) = |L|^{-1} \sum_{x \in L} I[h(x) \neq h^*(x)]$. If $\mathsf{err}_L(h) = 0$ for some h we say that h is *consistent* with the data.

To formally encapsulate the regularity or compatibility assumption for two-sided disjunctions described in the introduction, we consider the regularity or *compatibility function* χ: $\chi(h, x) = I[h_+(x) = -h_-(x)]$ for any hypothesis h and example $x \in X$. In addition, we define (overloading notation) the compatibility between h and the distribution D as $\chi(h, D) = \mathbb{E}_{x \sim D}[\chi(h, x)] = \Pr_{x \sim D}[h_+(x) = -h_-(x)]$. For a set U of unlabeled examples, define the empirical compatibility between h and U as $\chi(h, U) = |U|^{-1} \sum_{x \in U} I[h_+(x) = -h_-(x)]$. If $\chi(h, U) = 1$ we say that h is *compatible* with the data. Thus a hypothesis is consistent and compatible with a set of examples if every example contains exactly one type of indicator and every labeled example contains an indicator of the same type as its label. We will assume throughout this chapter that the target function is compatible.

We define, for any $\epsilon > 0$, the reduced hypothesis class $C_{D,\chi}(\epsilon) = \{h \in C : 1 - \chi(h, D) \leq \epsilon\}$, the set of hypotheses with "unlabeled error" at most ϵ. Similarly, for an unlabeled sample U, we define $C_{U,\chi}(\epsilon) = \{h \in C : 1 - \chi(h, U) \leq \epsilon\}$. The key benefit of using unlabeled data and our regularity assumption is that the number of labeled examples will only depend on $\log(C_{D,\chi}(\epsilon))$ which for helpful distributions will be much smaller than $\log(C)$.

3.2.1 The Commonality Graph

The basic structure used by all of our algorithms is a construct we call the *commonality graph*. As mentioned in the introduction, the commonality graph is the graph on variables that contains an edge between any two vertices if the corresponding variables appear together in a common example. That is, given the set U of unlabeled examples, define the commonality graph $G_{\text{com}}(U) = (V, E)$ where $V = \{x_1, \ldots, x_n\}$ and E contains an edge (x_i, x_j) if and only if there is some $x \in U$ such that x_i and x_j are both set to 1 in x. Furthermore, given the set L of labeled examples, let V_L^+ be the set of variables appearing in positive examples and V_L^- be the set of variables appearing in negative examples.

The edge structure of the commonality graph and the labeled examples will allow us to draw inferences about which vertices in the graph correspond to positive indicators, negative indicators, and non-indicators in the target concept. Any variable that appears in a labeled example cannot be an indicator of the type opposite of the label. In addition, an edge between two variables implies they cannot be indicators

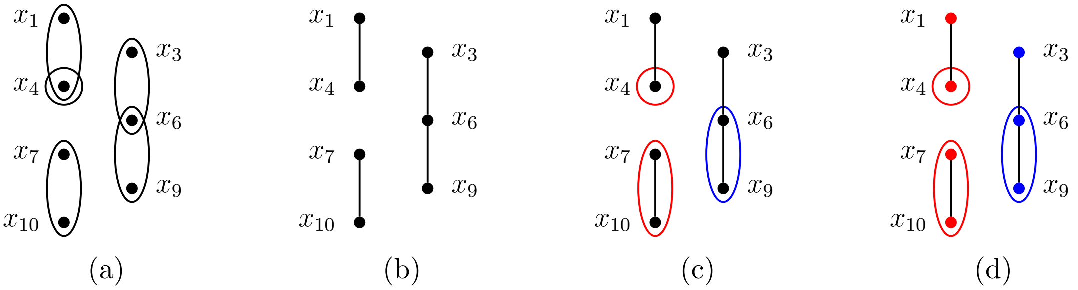


Figure 3: Schematic of learning two-sided disjunctions in the absence of non-indicators. First unlabeled examples (a) are used to create a commonality graph (b). Then labels are queried for each connected component (c) and a consistent, compatible hypothesis is output (d).

of different types. This means that any path in the commonality graph between positive and negative indicators must contain a non-indicator. Similarly, paths that pass only through indicator variables can be used to propagate labels to the unlabeled examples.

To see why the presence of irrelevant variables significantly complicates the algorithmic problem, consider the case in which there are no non-indicators (the case studied in [13]). If the target function indeed satisfies our regularity assumption, then no component will get multiple labels, so all we need to learn is a single labeled example in each component. Furthermore, if the number of components in the underlying graph is small, then both in the semi-supervised and active learning setting we can learn with many fewer labeled examples then in the passive supervised case. See Figure 3 for an illustration of learning in this setting.

Introducing non-indicators into the target concept complicates matters because components can now have multiple labels. We could think of the non-indicators as forming a vertex cut in the commonality graph separating variables corresponding to positive indicators from those corresponding to negative ones. To learn well, one could try to find such a cut with the necessary properties to ensure compatibility with the unlabeled data (i.e. no examples are composed only of non-indicators). Unfortunately, this is a difficult combinatorial problem in general.

Interestingly, we will be able to find such a cut in the semi-supervised setting for $k = O(\log n)$ and for general k we will be still be able to learn with nearly optimal rates, if each non-indicator appears with non-negligible probability; we do this by identifying a superset of non-indicators and carefully making inferences using it. Furthermore, since classification mistakes reveal vertices in both sides of the cut, the adaptive query ability in the the active learning model will allow us to actively search for vertices in the cut, without any conditions on the distribution.

3.2.2 Finding a Consistent Compatible Hypothesis is NP-hard

The following theorem formalizes the computational difficulty of finding a fully consistent and compatible two-sided disjunction in the semi-supervised setting.

Theorem 1. *Given data sets L and U as input, it is* NP*-hard to find a hypothesis $h \in C$ that is both consistent with L and compatible with U.*

Proof. The proof is by reduction from 3-SAT. Given a 3-SAT instance φ on variables $x_1, \ldots, x_n$ we produce the following data sets L and U containing examples on the $4n$ variables $x_1^+, x_1^-, \bar{x}_1^+, \bar{x}_1^-, \ldots, x_n^+, x_n^-, \bar{x}_n^+, \bar{x}_n^-$. The labeled set L contains examples of the form $(\{x_i^+, \bar{x}_i^+\}, +1)$ and $(\{x_i^-, \bar{x}_i^-\}, -1)$ for $1 \leq i \leq n$. In addition, for each clause in φ of the form $(\ell_i \vee \ell_j \vee \ell_k)$ where ℓ_i, ℓ_j, ℓ_k can each be a positive or negative literal, L contains the example $(\{\ell_i^+, \ell_j^+, \ell_k^+\}, +1)$. The unlabeled set U contains examples of the form $\{x_i^+, x_i^-\}$ and $\{\bar{x}_i^+, \bar{x}_i^-\}$ for $1 \leq i \leq n$. The labelings that are consistent and compatible with all the non-clause examples correspond precisely to assignments of $x_1, \ldots, x_n$, and the clauses are compatible with a given hypothesis only if they are satisfied in the underlying assignment. The set of positive indicators of any hypothesis $h = (h_+, h_-) \in C$ that is both consistent with L and compatible with U corresponds to a truth assignment to $x_1, \ldots, x_n$ that satisfies φ, therefore finding such a hypothesis is NP-hard. □

3.3 Semi-supervised Learning

Our general strategy is to identify non-indicators and remove them from the commonality graph, reducing this problem to the simpler case. Notice that each non-indicator that appears in the unlabeled data is significant; failing to identify it can lead to incorrect inferences about a large probability mass of examples. A variable is obviously a non-indicator if it appears in both positive and negative examples. A naïve approach would be to draw enough labeled examples so that every significant non-indicator appears in examples with both labels. The problem with this approach is that some non-indicator can appear much more frequently in positive examples than in negative examples. In this case the number of examples needed by the naïve approach is inversely proportional to the probability of that non-indicator appearing in negative examples. This sample complexity can be worse than in the fully supervised case.

In our approach, it is enough to ensure each non-indicator appears in a labeled example, but not necessarily in both positive and negative examples. The number of examples needed in this case will now depend on the minimum probability of a non-indicator appearing. This allows the sample complexity to be significantly smaller than that of the naïve approach; for example, when a non-indicator appears in positive examples with constant probability while in negative examples with probability ϵ/n.

Our approach can still identify non-indicators, now by examining paths in the commonality graph. In paths whose interior vertices appear only in unlabeled examples (i.e. are indicators) and whose endpoints appear in oppositely labeled examples, one of the endpoints must be a non-indicator. When $k = O(\log n)$ we can enumerate over all consistent compatible hypotheses efficiently by restricting our attention to a small set of paths.

If the number of non-indicators is larger, we can still find a good hypothesis efficiently by finding the non-indicators one at a time. Each time our working hypothesis makes a mistake this reveals a path whose endpoint is a non-indicator.

The number of labeled examples we require will depend on the *minimum non-indicator probability* defined by

$$\epsilon_0(D, h^*) = \min_{x_i \notin h^*_+ \cup h^*_-} \Pr_{x \sim D}[x_i = 1].$$

For notational convenience denote it simply by ϵ_0 without ambiguity. To guarantee with high probability that each non-indicator appears in some labeled example, it suffices to use $\tilde{O}(\frac{1}{\epsilon_0} \log k)$ labeled examples.

3.3.1 Finding a Consistent, Compatible Hypothesis Efficiently when $k = O(\log n)$

We now give an algorithm, along with the intuition behind it, for finding a two-sided disjunction that is consistent and compatible with a given training set. Our algorithm will not run efficiently on every possible input (since, as shown above, this problem is NP-hard in general), but the algorithm is efficient for a large class of possible inputs. Given example sets L and U, the algorithm begins by constructing the commonality graph $G = G_{\text{com}}(U)$ and setting G to $G \setminus (V_+ \cap V_-)$. This removes any variables that appear in both positive and negative examples as these are guaranteed to be non-indicators.

To identify the rest of the non-indicators, we consider a new graph. Using $u \leftrightarrow_G v$ to denote the existence of a path in the graph G between vertices u and v, we define the *indicator graph* $G_{\text{ind}}(G, V_+, V_-)$ to be the bipartite graph with vertex set $V_+ \cup V_-$ and edge set $\{(u, v) \in V_+ \times V_- : u \leftrightarrow_{G \setminus (V_+ \cup V_-)} v\}$. The key idea is that an edge in this graph implies that at least one of its endpoints is a non-indicator, since the two variables appear in oppositely labeled examples but are connected by a path of indicators.

Note that the target set of non-indicators must form a vertex cover in the indicator graph. By iterating over all minimal vertex covers, we must find a subset of the target non-indicators whose removal disconnects positive examples from negative

Algorithm 1 Finding a consistent compatible two-sided disjunction

```
Input: unlabeled set $U$, labeled set $L$
Set $G = G_{\text{com}}(U)$, $V_+ = V_L^+$, $V_- = V_L^-$
Set $G = G \setminus (V_+ \cap V_-)$
Set $V_+ = V_+ \cap G$, $V_- = V_- \cap G$
Set $G_I = G_{\text{ind}}(G, V_+, V_-)$
for each minimal vertex cover $S$ of $G_I$ do
    Set $G' = G \setminus S$, $V'_+ = V_+ \setminus S$, $V'_- = V_- \setminus S$
    Set $h_+ = \{v \in G' : \exists u \in V'_+,\ u \leftrightarrow_{G'} v\}$
    if $(h_+, G' \setminus h_+)$ is consistent and compatible then
        break
Output: hypothesis $h = (h_+, G' \setminus h_+)$
```

examples, and this corresponds to a consistent compatible hypothesis. The algorithm is summarized in Algorithm 1.

The key step in Algorithm 1 is enumerating the minimal vertex covers of the indicator graph G_I. One way to do this is as follows. First find a maximum matching M in G_I, and let m be the number of disjoint edges in M. Enumerate all 3^m subsets of vertices that cover M (for every edge in M, one or both of the endpoints can be included in the cover). For each such cover S, extend S to a minimal vertex cover of G_I by adding to S every variable not covered by M that has no neighbors already in S. This extension can always be done uniquely, so there is a one-to-one correspondence between covers of M and minimal vertex covers of G_I.

This enumeration method gives us both a concrete way to implement Algorithm 1 and a way to bound its running time. We prove in Theorem 2 that given enough data, Algorithm 1 correctly outputs a consistent compatible hypothesis with high probability. We then bound its time and sample complexity.

Theorem 2. *For any distribution D over $\{0,1\}^n$ and target concept $h^* \in C$ such that $\chi(h^*, D) = 1$, h^* has at most k non-indicators, and the minimum non-indicator probability is ϵ_0, if*

$$m_u \geq \frac{1}{\epsilon}\left[\log \frac{2|C|}{\delta}\right] \qquad \textit{and} \qquad m_l \geq \max\left\{\frac{1}{\epsilon_0}\log\frac{k}{\delta}, \frac{1}{\epsilon}\left[\log\frac{2|C_{D,\chi}(\epsilon)|}{\delta}\right]\right\}$$

then with probability at least $1-2\delta$*, Algorithm 1 outputs a hypothesis* $h \in C$ *such that* $\mathsf{err}_L(h) = 0$*,* $\chi(h,U) = 1$*, and* $\mathsf{err}(h) \leq \epsilon$*. Furthermore, when* $k = O(\log n)$ *the algorithm runs in time at most poly*(n)*.*

Proof. We separate the proof into three sections, first proving consistency and compatibility of the output hypothesis, then giving the sample sizes required to guarantee good generalization, and finally showing the overall running time of the algorithm.

Consistency and Compatibility. The exit condition for the loop in Algorithm 1 guarantees that the algorithm will output a consistent compatible hypothesis, so long as a suitable minimal vertex cover of G_I is found. Thus, it suffices to show that such a vertex cover exists with high probability when L is large enough.

By the definition of ϵ_0 along with the independence of the samples and a union bound, if $m_l > \frac{1}{\epsilon_0} \log \frac{k}{\delta}$, then with probability at least $1-\delta$, all non-indicator variables appear in some labeled example. We will assume in the remainder of the proof that all variables not in $V_L^+ \cup V_L^-$ are indicators.

Since an edge in G between indicators forces both endpoints to be of the same type, a path through indicators does the same. Edges in G_I correspond to such paths, but the endpoints of such an edge cannot be indicators of the same type because they appear in differently labeled examples. It follows that at least one endpoint of every edge in G_I must be a non-indicator.

Now let V_0 be the set of non-indicators in the target hypothesis. The above observations imply that V_0 contains a vertex cover of G_I. It follows that there must exist a subset $\tilde{S} \subseteq V_0$ that is a minimal vertex cover of G_I. Let $\tilde{h} = (\tilde{h}_+, \tilde{h}_-)$ be the hypothesis h formed from the minimal vertex cover $S = \tilde{S}$. We only need to show that $\tilde{h}$ is both consistent and fully compatible.

Every indicator of h^* is also an indicator of $\tilde{h}$ since only true non-indicators were removed from G and all remaining variables became indicators in $\tilde{h}$. Since every example contains an indicator of h^*, every example must contain an indicator of $\tilde{h}$ of the correct type. Furthermore, if an example contained both positive and negative indicators, this would imply an edge still present in G_I. But removing a vertex cover removes all edges, so this is impossible. Hence $\tilde{h}$ is a consistent, fully compatible hypothesis.

Generalization Error. If

$$m_u \geq \frac{1}{\epsilon}\left[\log\frac{2|C|}{\delta}\right] \qquad \text{and} \qquad m_l \geq \max\left\{\frac{1}{\epsilon_0}\log\frac{k}{\delta},\ \frac{1}{\epsilon}\left[\log\frac{2|C_{D,\mathcal{X}}(\epsilon)|}{\delta}\right]\right\},$$

the above analysis states that Algorithm 1 will fail to produce a consistent compatible hypothesis with probability at most δ. Furthermore, an algorithm with true error rate greater than ϵ will be fully consistent with a labeled set of size m_l with probability at most $\delta/C_{D,\mathcal{X}}(\epsilon)$. Union bounding over all compatible hypotheses we see that a consistent compatible hypothesis will fail to have an error rate less than ϵ with probability at most δ. By a union bound over the two failure events, the overall probability of failure is $\leq 2\delta$.

Running Time. Since checking consistency and compatibility can be done in time polynomial in the number of examples, the limiting factor in the running time is the search over minimal vertex covers of G_I. In a bipartite graph, the size of the minimum vertex cover is equal to the size of the maximum matching. The set of k non-indicators in the target hypothesis includes a vertex cover of G_I, so the size m of the maximum matching is at most k. There is one minimal vertex cover for each of the 3^m covers of the maximum matching, so the number of minimal vertex covers to search is at most 3^k.

3.3.2 A General Semi-supervised Algorithm

Algorithm 1 is guaranteed (provided the labeled set is large enough) to find a hypothesis that is both consistent and compatible with the given data but is efficient only when k is logarithmic in n. When k is instead polylogarithmic in n, our algorithm is no longer efficient but still achieves a large improvement in sample complexity over supervised learning. We now present an efficient algorithm for finding a low-error (but not necessarily consistent and compatible) hypothesis which matches the sample complexity of Algorithm 1.

The algorithm, summarized in Algorithm 2, begins by constructing the commonality graph from the unlabeled examples and identifying potential indicators from a subset of the labeled examples. We use sample(m, S) to denote a random sample of m elements from set S. An initial hypothesis is built and tested on the sequence of remaining labeled examples. If the hypothesis makes a mistake, it is updated and testing continues. Each update corresponds to either identifying a non-indicator or labeling all indicators in some connected component in the commonality graph, so the number of updates is bounded. Furthermore, if the hypothesis makes no mistakes on a large enough sequence of consecutive examples, then with high probability it has a small error rate overall. This gives us a stopping condition and allows us to bound the number of examples seen between updates.

The hypotheses in Algorithm 2 use a variation on nearest neighbor rules for classification. Given a commonality graph G and a set L of labeled examples, the associated nearest neighbor hypothesis $h_{G,L}$ classifies an example x the same as the nearest labeled example in L. The distance between two examples x and x' is the measured by the minimum path distance in G between the variables in x and the variables in x'. If no examples in L are connected to x, then $h_{G,L}$ classifies x negative by default. For convenience, we use $\mathrm{nn}_{G,S}(x)$ to denote the vertex in the set S nearest to a variable in the example x via a path in G. If no such vertex exists, $\mathrm{nn}_{G,S}(x)$ returns the empty set. Using hypotheses of this form ensures that the neighbor variable used to classify an example x is connected to x by a path through indicators, which allows us to propagate its label to the new example. If the example is misclassified, we must have been fooled by a non-indicator.

Algorithm 2 Learning a Low-error Hypothesis for Two-Sided Disjunctions

```
Input: data sets $U$ and $L$, parameters $\epsilon$, $\delta$, $k$
Set $L' = \text{sample}(\frac{1}{\epsilon_0} \log \frac{k}{\delta}, L)$ and $L = L \setminus L'$
Set $G = G_{\text{com}}(U) \setminus (V_{L'}^{+} \cap V_{L'}^{-})$
Set $P = G \cap (V_{L'}^{+} \cup V_{L'}^{-})$
Set $h = h_{G,L'}$ and $c = 0$
while $L \neq \emptyset$ and $c \leq \frac{1}{\epsilon} \log \frac{k+T}{\delta}$ do
  Set $x = \text{sample}(1, L)$
  Set $L = L \setminus \{x\}$, and $L' = L' \cup \{x\}$
  if $h(x) \neq \ell(x)$ then
    Set $G = G \setminus \text{nn}_{G,P}(x)$
    Set $h = h_{G,L'}$ and $c = 0$
  else
    Set $c = c + 1$
Output: the hypothesis $h$
```

The number of examples used by Algorithm 2 depends on T, the number of connected components in the commonality graph after removing all non-indicators. Lemma 1 bounds this quantity by the number of compatible hypotheses.

Lemma 1. *Let G be the graph that results from removing all non-indicators from $G_{\text{com}}(U)$, and suppose G is divided into T connected components. If $m_u \geq \frac{2n^2}{\epsilon} \log \frac{n}{\delta}$, then $T \leq \log_2 |C_{D,\mathcal{X}}(\epsilon)|$ with probability at least $1 - \delta$.*

Proof. Since G has no non-indicators, a hypothesis is compatible with U if and only if every component is made entirely of indicators of the same type. There are two possible choices for each component, so the number of fully compatible hypotheses is $|C_{U,\mathcal{X}}(0)| = 2^T$.

To complete the proof, it is sufficient to show that $C_{U,\mathcal{X}}(0) \subseteq C_{D,\mathcal{X}}(\epsilon)$. Since any hypothesis in $C_{U,\mathcal{X}}(0)$ is compatible with any example containing variables from only one component, we only need to show that there is at most ϵ probability mass of examples that contain variables from multiple components. All such examples correspond to edges that are absent from $G_{\text{com}}(U)$, so we only need to show that $G_{\text{com}}(U)$ was constructed with enough examples so that nearly all significant edges appear in the graph.

To see this, fix any pair of variables x_i, x_j. If $\Pr_{x\sim D}[x_i = 1 \wedge x_j = 1] < \epsilon/n^2$, we can ignore this pair since all such pairs together constitute a probability mass strictly less than ϵ. Now suppose $\Pr_{x\sim D}[x_i = 1 \wedge x_j = 1] \geq \epsilon/n^2$. The probability that x_i and x_j do not appear together in any of the examples in U is at most $(1 - \frac{\epsilon}{n^2})^{m_u}$, so if $m_u \geq \frac{n^2}{\epsilon} \log \frac{n^2}{\delta}$ then this failure probability is at most δ/n^2. By a union bound over all such pairs, with probability at least $1 - \delta$ all corresponding edges appear in $G_{\text{com}}(U)$, and the probability mass of examples containing variables from multiple components is at most ϵ. This means that every fully compatible hypothesis has unlabeled error at most ϵ, so we have $T = \log_2 |C_{U,\chi}(0)| \leq \log_2 |C_{D,\chi}(\epsilon)|$. □

The following theorem bounds the number of examples sufficient for Algorithm 2 to output a low-error hypothesis.

Theorem 3. *For any distribution D over $\{0,1\}^n$ and target concept $h^* \in C$ such that $\chi(h^*, D) = 1$, h^* has at most k non-indicators, and the minimum non-indicator probabilityis ϵ_0, if $m_u \geq \frac{2n^2}{\epsilon} \log \frac{n}{\delta}$ and*

$$m_l \geq \frac{1}{\epsilon_0} \log \frac{k}{\delta} + \frac{k + \log |C_{D,\chi}(\epsilon)|}{\epsilon} \left[\log \frac{k + \log |C_{D,\chi}(\epsilon)|}{\delta} \right]$$

then with probability at least $1 - 3\delta$, Algorithm 2 outputs a hypothesis h in polynomial time such that $\mathsf{err}(h) \leq \epsilon$.

Proof. **Generalization Error.** First note that according to the loop exit condition, Algorithm 2 outputs the first hypothesis it encounters that correctly classifies a sequence of at least $\frac{1}{\epsilon} \log \frac{k+T}{\delta}$ i.i.d. examples from D. If $\mathsf{err}(h) > \epsilon$ for some hypothesis h, then the probability that h correctly classifies such a sequence of examples is at most $(1 - \epsilon)^{\frac{1}{\epsilon} \log \frac{k+T}{\delta}} \leq \frac{\delta}{k+T}$. Assuming Algorithm 2 updates its hypothesis at most $k+T$ times, a union bound over the $k+T$ hypotheses considered guarantees that with probability at least $1-\delta$, the hypothesis output by Algorithm 2 has error rate at most

ϵ. In the remainder of the proof, we will bound the total number of samples required by Algorithm 2 and show that it makes at most $k + T$ updates to its hypothesis.

Mistake Bound. By the definition of ϵ_0, the initial set of m_l labeled examples ensures that with probability at least $1 - \delta$ all non-indicators are included in the potential indicator set P, so all variables outside P (call this set Q) are indicators. We will assume such an event holds throughout the remainder of the proof. In particular, this means that any paths through Q must consist entirely of indicators of the same type.

Suppose at some point during the execution of Algorithm 2, the intermediate hypothesis h misclassifies an example x. There are two types of such mistakes to consider. If the variables in x are not connected to any variables in P, then by the above observation, all variables connected to x are indicators of the same type, and in particular, they are indicators of the type corresponding to the label of x. This means that this type of mistake can occur only when h knows of no labeled examples connected to x. Once h is updated to be $h_{G,L'}$ where $x \in L'$, h can make no further mistakes of this type on any examples connected to x. Thus, Algorithm 2 can make at most T mistakes of this type before all connected components have labeled examples.

The hypothesis $h_{G,L'}$ labels x with the label of the example of L' containing $\mathrm{nn}_{G,P}(x)$. If x is labeled incorrectly, then this must be an example with label opposite that of x. But since the path between $\mathrm{nn}_{G,P}(x)$ and x consists only of vertices not in P, i.e. indicators, we conclude that $\mathrm{nn}_{G,P}(x)$ must be a non-indicator. Algorithm 2 can make at most k mistakes of this type before all non-indicators are removed from G.

Sample Complexity and Running Time. We have shown that after Algorithm 2 makes $k+T$ updates, all non-indicators have been removed from G and all connected components in G contain a variable that has appeared in a labeled example. Since at most $\frac{1}{\epsilon} \log \frac{k+T}{\delta}$ examples can be seen between updates, the total number of labeled examples needed by Algorithm 2 is at most

$$\frac{1}{\epsilon_0}\log\frac{k}{\delta}+\frac{k+T}{\epsilon}\log\frac{k+T}{\delta}.$$

Straightforward algebra and an application of Lemma 1 yields the bound given in the theorem statement, and a union bound over the three failure events of probability δ yields the stated probability of success. The time complexity is clearly polynomial in n per example and therefore polynomial overall. □

3.4 Active Learning

We now consider the problem of learning two-sided disjunctions in the active learning model, where the learner has access to a set U of unlabeled examples and an oracle that returns the label of any example in U it submits. The additional power provided by this model allows us to use the same strategy as in the semi-supervised algorithm in Section 3.3.2 while achieving sample complexity bounds independent of ϵ_0.

As in Section 3.3.2, the goal will be to identify and remove non-indicators from the commonality graph and obtain labeled examples for each of the connected components in the resulting graph. In the semi-supervised model we could identify a mistake when there was a path connecting a positive labeled example and a negative labeled example. To identify non-indicators we guaranteed that they would lie on the endpoints of these labeled paths. In the active learning setting, we are able to check the labels of examples along this path, and thus are able to remove our dependence on minimum non-indicator probability parameter.

The algorithm we propose can be seen as a slight modification of Algorithm 2. The idea is to maintain a set of at least one labeled example per connected component and to test the corresponding nearest neighbor hypothesis on randomly chosen examples. If the hypothesis misclassifies some example, it identifies a path from the example to its nearest neighbor. Since these examples have opposite labels, a non-indicator must be present at a point on the path where positive indicators switch to negative

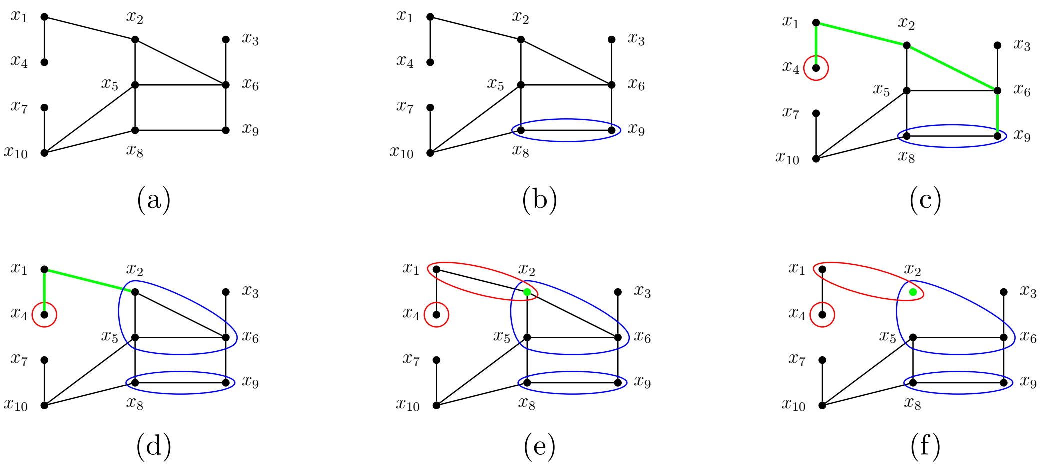


Figure 4: Schematic diagram of Algorithm 3. The commonality graph (a) is formed from unlabeled data. Labels are queried for each connected component in (b). Binary search is used to identify a non-indicator in (c), (d), and (e), followed by the removal of variable x_2 from the graph (f).

indicators, and such a non-indicator can found in logarithmically many queries by actively choosing examples to query along this path in a binary search pattern. The search begins by querying the label of an example containing the variable at the midpoint of the path. Depending on the queried label, one of the endpoints of the path is updated to the midpoint, and the search continues recursively on the smaller path whose endpoints still have opposite labels. Let $\text{BinarySearch}_{G,L}(x)$ return the non-indicator along the path in G from a variable in x to $\text{nn}_{G,L}(x)$. As with Algorithm 2, the algorithm halts after removing all k non-indicators or after correctly labeling a long enough sequence of random examples.

The details are described in Algorithm 3, and the analysis is presented in Theorem 4.

Theorem 4. *For any distribution D over $\{0,1\}^n$ and target concept $h^* \in C$ such that $\chi(h^*, D) = 1$ and h^* has at most k non-indicators. If $m_u \geq \frac{2n^2}{\epsilon} \log \frac{n}{\delta}$ then after at most*

$$m_q = O\left(\log |C_{D,\chi}(\epsilon)| + k\left[\log n + \frac{1}{\epsilon}\log\frac{k}{\delta}\right]\right)$$

Algorithm 3 Actively Learning Two-Sided Disjunctions

```
Input: unlabeled data $U$, parameters $\epsilon$, $\delta$, $k$
Set $G = G_{\text{com}}(U)$ and $L = \emptyset$
for each connected component $R$ of $G$ do
    Choose $x \in U$ such that $x \subseteq R$
    Set $L = L \cup \{(x, \ell(x))\}$
Set $h = h_{G,L}$ and $c = 0$
while $c \leq \frac{1}{\epsilon} \log \frac{k}{\delta}$ do
    Set $x = \text{sample}(1, U)$ and $L = L \cup \{(x, \ell(x))\}$
    if $h(x) \neq \ell(x)$ then
        Set $v = \text{BinarySearch}_{G,L}(x)$
        Set $G = G \setminus \{v\}$
        for each new connected component $R$ of $G$ do
            Choose $x \in U$ such that $x \subseteq R$
            Set $L = L \cup \{(x, \ell(x))\}$
        Set $h = h_{G,L}$ and $c = 0$
    else
        Set $c = c + 1$
Output: the hypothesis $h$
```

label queries, with probability $\geq 1 - 2\delta$, Algorithm 3 outputs a hypothesis h in polynomial time such that $\mathsf{err}(h) \leq \epsilon$.

Proof. **Generalization Error.** According to the exit condition of the loop in Step 3, Algorithm 3 outputs the first hypothesis it encounters that correctly classifies a sequence of at least $\frac{1}{\epsilon} \log \frac{k}{\delta}$ i.i.d. examples from D. If $\mathsf{err}(h) > \epsilon$ for some hypothesis h, then the probability that h correctly classifies such a sequence of examples is at most $(1-\epsilon)^{\frac{1}{\epsilon}\log\frac{k}{\delta}} \leq \frac{\delta}{k}$. Assuming Algorithm 3 updates its hypothesis at most k times, a union bound over the k hypotheses considered guarantees that with probability at least $1 - \delta$, the hypothesis output by Algorithm 3 has error rate at most ϵ. In the remainder of the proof, we will bound the total number of samples required by Algorithm 3 and show that it makes at most k updates to its hypothesis.

Queries per Stage. In the loops over connected components of G, one label is queried for each component. The components are those formed by removing from G a subset of the non-indicators, so the total number of queries made in these loops is at most T, the number of components after removing all non-indicators.

Now suppose the hypothesis h misclassifies an example x. Let x' be the nearest labeled example to x, and let x_i and x_j be the endpoints of the shortest path from x to x' in G. If each variable along the path appears in examples of only one label, then there could be no path between x_i and x_j, which appear in examples with different labels. Thus, there must exist a variable along the path from x_i to x_j that appears in both positive and negative examples, i.e. a non-indicator. Since the commonality graph was constructed using the examples in U, we can query the labels of examples that contain variables between x_i and x_j in order to find the non-indicator. Using binary search, the number of queries is logarithmic in the path length, which is at most n.

Query Complexity and Running Time. Each mistake results in removing a non-indicator from the G, so at most k mistakes can be made. For each mistake, $O(\log n)$ queries are needed to find a non-indicator to remove and at most $\frac{1}{\epsilon} \log \frac{k}{\delta}$ more queries are used before another mistake is made. Combined with the queries for the connected components, we can bound the total number of queries by $O\left(T + k\left[\log n + \frac{1}{\epsilon} \log \frac{k}{\delta}\right]\right)$. We can further bound T by $\log |C_{D,\mathcal{X}}(\epsilon)|$ via Lemma 1, and pay the price of an additional δ probability of failure. The running time for this algorithm is clearly polynomial. □

3.5 Discussion

One drawback of our semi-supervised algorithms is that their dependence on the minimum non-indicator probability restricts the class of distributions under which they can be used for learning. Additionally, the class of target concepts for which Algorithm 1 can efficiently learn a consistent and compatible hypothesis is restricted, and our reduction proves that some such restriction is necessary since the general problem is NP-hard. The surprising result of our work is that both restrictions can be lifted entirely in the active learning setting while improving label complexity at the same time. The ability to adaptively query the labels of examples allows us to execute a strategy for identifying non-indicators that would require too many labeled examples in the semi-supervised setting. As this represents the first known example of how active learning can be used to avert computational difficulties present in semi-supervised learning, we hope this work will lead to more such examples and to a more general understanding of when active learning provides this type of advantage.

It is important to note that the problem of learning two-sided disjunctions can be viewed as learning under a large-margin assumption. We can represent a two-sided disjunction h as a linear threshold function $h(x) = \mathrm{sign}(w \cdot x)$ where $w_i = +1$ for positive indicators, $w_i = -1$ for negative indicators, and $w_i = 0$ for each of the k non-indicators. If h is fully compatible with the distribution D, every $x \sim D$ has at least one indicator set to 1 and does not have any indicators of the opposite sign set to 1. This means that $|w \cdot x| \geq 1$, so when $\|x\|_1 \leq k$ we immediately have

$$\frac{|w \cdot x|}{\|w\|_\infty \|x\|_1} \geq \frac{1}{k}$$

and when $\|x\|_1 > k$, $|w \cdot x|$ is minimized when x has k non-indicators, so we have

$$\frac{|w \cdot x|}{\|w\|_\infty \|x\|_1} \geq \frac{\|x\|_1 - k}{\|x\|_1} \geq \frac{1}{k+1}.$$

Combining these two cases gives us an $L_\infty L_1$ margin of $O(1/k)$. This is a different notion of margin than the $L_2 L_2$ margin appearing in the mistake bounds for the Perceptron algorithm [86] and the $L_1 L_\infty$ margin appearing in the bounds for Winnow [73]. Providing generic algorithms with bounds depending on the $L_\infty L_1$ margin (and more generally, the $L_q L_p$ margin) is the main topic discussed in Chapter 4.

CHAPTER IV

PASSIVE AND ACTIVE LEARNING WITH LARGE L_qL_p MARGINS

The notion of "margin" arises naturally in many areas of machine learning. Margins have long been used to motivate the design of algorithms [33, 14], to give sufficient conditions for fast learning rates [18, 70], and to explain unexpected behavior of algorithms in practice [90]. Here we are concerned with learning the class of homogeneous linear separators in $\mathbb{R}^d$ over distributions with large margins. We use a general notion of margin, the L_qL_p margin, that captures, among others, the notions used in the analyses of Perceptron ($p = q = 2$) and Winnow ($p = \infty$, $q = 1$). For $p, q \in [1, \infty]$ with $1/p + 1/q = 1$, the L_qL_p margin of a linear classifier $x \mapsto \text{sign}(w \cdot x)$ with respect to a distribution D is defined as

$$\gamma_{q,p}(D, w) = \inf_{x \sim D} \frac{|w \cdot x|}{\|w\|_q \|x\|_p}.$$

While previous work has addressed the case of $p \geq 2$ both theoretically [50, 91, 46] and experimentally [110], the $p < 2$ case has been mentioned but much less explored. This gap in the literature is possibly due to the fact that when $p < 2$ a large margin alone does not guarantee small sample complexity (see Example 1 below for such a case). This leads to the question of whether large L_qL_p margins with $p < 2$ can lead to small sample complexity, and if so, under what conditions will this happen? Furthermore, are there situations in which taking advantage of margins with $p < 2$ can lead to better performance than using margins with $p \geq 2$?

In this chapter, we answer these three questions using both theoretical and empirical evidence. We first give a bound on the generalization error of linear separators with large L_qL_p margins that holds for any finite $p \geq 1$. The result is proved through a new bound on the fat-shattering dimension of linear separators with bounded L_q norm. The bound improves upon previous results by removing a factor of $\log d$ when $2 \leq p < \infty$ and extends the previously known bounds to the case of $1 \leq p < 2$. A highlight of this theoretical result is that it gives a simple sufficient condition for fast learning even for the $p < 2$ case. The condition is related to the $L_{2,p}$ norm of the data matrix and can be estimated from the data.

We then give a concrete family of learning problems in which using the $L_\infty L_1$ margin gives significantly better sample complexity guarantees than for $L_q L_p$ margins with $p > 1$. We define a family of distributions over labeled examples and consider the sample complexity of learning the class W_p of linear separators with large $L_q L_p$ margins. By bounding covering numbers, we upper bound the sample complexity of learning W_1 and lower bound the complexity of learning W_p when $p > 1$, and we show that the upper bound can be significantly smaller than the lower bound.

In addition, we give experimental results supporting our claim that taking advantage of large $L_\infty L_1$ margins can lead to faster learning. We observe that in the realizable case, the problem of finding a consistent linear separator that maximizes the $L_q L_p$ margin is a convex program (similar to SVM). An extension of this method to the non-realizable case is equivalent to minimizing the L_q-norm regularized hinge loss. We apply these margin-maximization algorithms to both synthetic and real-world data sets and find that maximizing the $L_\infty L_1$ margin can result in better classifiers than maximizing other margins. We also show that the theoretical condition for fast learning that appears in our generalization bound is favorably satisfied on many real-world data sets.

Finally, we show how active learning can be used to adapt to the optimal margin parameters. We activize the $L_q L_p$ SVM via the simple margin strategy of Tong & Koller [103] and empirically demonstrate that the resulting algorithm selects examples to query that allow it not only to determine which margin parameters to use but to exploit that knowledge by querying the appropriate labels. This is an example of using active learning to discover and exploit structure in data and serves as our second example of how the power to make adaptive label queries has benefits beyond reducing labeling effort over passive learning.

We note that much of this chapter can be equivalently interpreted as results on norm-based regularization rather than learning with large-margins. As can be seen from the formulation of the support vector machine optimization problem (see Section 4.2.1), maximizing $L_q L_p$ margin is equivalent to performing L_q-norm regularization on examples with unit (or bounded) L_p-norm. Under this interpretation, our results say that we can adaptively determine which type of regularization will lead to the fastest learning rates, and then we can query the labels of the examples that will lead to the best label complexity under the chosen form of regularization.

4.1 Related Work

It has long been known that the classic algorithms Perceptron [86] and Winnow [73] have mistake bounds of $1/\gamma_{2,2}^2$ and $\tilde{O}(1/\gamma_{1,\infty}^2)$, respectively. A family of "quasi-additive" algorithms [50] interpolates between the behavior of Perceptron and Winnow by defining a Perceptron-like algorithm for any $p > 1$. While this gives an algorithm for any $1 < p \leq \infty$ the mistake bound of $\tilde{O}(1/\gamma_{q,p}^2)$ only applies for $p \geq 2$. For small values of p, these algorithms can be used to learn non-linear separators by using factorizable kernels [47]. A related family [91] was designed for learning in the PAC model rather than the mistake bound model, but again, guarantees were only given for $p \geq 2$.

Other works [65, 32, 68, 76] bound the Rademacher complexity of classes of linear separators under general forms of regularization. Special cases of each of these regularization methods correspond to L_q-norm regularization, which is closely related to maximizing $L_q L_p$ margin. Specifically, Kakade et al. [65] directly consider the case of L_q-norm regularization but only give Rademacher complexity bounds for the case of $p \geq 2$. Kloft & Blanchard [68] give Rademacher complexity bounds that cover the entire range $1 \leq p \leq \infty$ in the context of multiple kernel learning[1], but their discussion of excess risk bounds for different choices of p is limited to the $p \geq 2$ case while our work discusses the generalization error over the entire range. Cortes et al [32] also give Rademacher complexity bounds for multiple kernel learning which hold only for even integers p. Maurer & Pontil [76] consider the more general setting of block-norm regularized linear classes but only give bounds for the case of $p \geq 2$. A work of Zhang [110] deals with algorithms for L_q-norm regularized loss minimization and discusses cases in which L_∞-norm regularization may be appropriate. In contrast to our work, none of the above works give lower bounds on the sample complexity or give concrete evidence of when some values of p will result in faster learning than others.

A more recent work of Neyshabur et al. [79] studies the use of norm-based regularization for capacity control of feedforward neural networks. Their Rademacher complexity bounds on the class of neural networks depend on a bound on the Rademacher complexity of L_q-norm regularized linear functions such as the bound we give in Section 4.3.1. Interestingly, they seek to find, among other things, conditions on when their bounds will have no or weak dependence on the size of the network, just as we are concerned with limiting dependence on data dimensionality.

4.2 Preliminaries

Let D be a distribution over a bounded instance space $X \subseteq \mathbb{R}^d$. A linear separator over X is a classifier $h(x) = \text{sign}(w \cdot x)$ for some weight vector $w \in \mathbb{R}^d$. We use h^* and w^* to denote the target function and weight vector, respectively, so that $h^*(x) =$

[1] Our setting is a special case of multiple kernel learning in which there are d kernels, one for each dimension of the instance space, where each kernel simply acts as a one-dimensional projection. That is, the i-th kernel K_i is defined as $K_i(x, x') = x_i x'_i$ and the corresponding feature transformation can be taken to be $\Phi_i(x) = x_i$.

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ x^1 & x^2 & \dots & x^n \\ | & | & & | \end{bmatrix} \begin{array}{lll} \rightarrow & L_2 & \downarrow \\ \rightarrow & L_2 & \downarrow \\ \rightarrow & L_2 & \downarrow L_p \end{array}$$

Figure 5: The $d \times n$ data matrix $\mathbf{X}$ is oriented with one unlabeled example in each column. The $L_{2,p}$-norm is found by first taking the L_2-norm of each row in $\mathbf{X}$, resulting in a vector of L_2-norms for each feature variable. The $\|\mathbf{X}\|_{2,p}$ is the L_p-norm of this resulting vector.

$\text{sign}(w^* \cdot x)$ gives the label for any instance x and $\mathsf{err}_D(h) = \Pr_{x \sim D}[h(x) \neq h^*(x)]$ is the generalization error of any hypothesis h. We will often abuse notation and refer to a classifier and its corresponding weight vector interchangeably. We will overload the notation $\mathbf{X}$ to represent either a set of n points $\{x^1, \dots, x^n\}$ in $\mathbb{R}^d$ or the $d \times n$ matrix of containing one point per column.

For any point $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ and $p \geq 1$, the L_p-norm of x is

$$\|x\|_p = \left(\sum_{i=1}^{d} |x_i|^p \right)^{\frac{1}{p}}$$

and the L_∞-norm is $\|x\|_\infty = \max_i |x_i|$. Let $\|X\|_p$ denote $\sup_{x \in X} \|x\|_p$, which is finite for any p by our assumption that X is bounded. The L_q-norm is the dual of the L_p-norm if $1/p + 1/q = 1$ (so the L_∞-norm and L_1-norm are duals). In this work, p and q will always denote dual norms.

For any weight vector w, the $L_q L_p$ margin of w with respect to D is defined as

$$\gamma_{q,p}(D, w) = \inf_{x \sim D} \frac{|w \cdot x|}{\|w\|_q \|x\|_p}.$$

We can similarly define $\gamma_{q,p}(\mathbf{X}, w)$ for a set $\mathbf{X}$. We will drop the first argument when referring to the distribution-based definition and the distribution is clear from context. We assume that D has a positive margin; that is, there exists w such that $\gamma_{q,p}(D, w) > 0$. Note that by Hölder's inequality, $|w \cdot x| \leq \|w\|_q \|x\|_p$ for dual p and q, so $\gamma_{q,p}(D, w) \leq 1$.

We also define the $L_{a,b}$ matrix norm

$$\|\mathbf{M}\|_{a,b} = \left(\sum_{i=1}^{r} \left(\sum_{j=1}^{c} |m_{ij}|^a \right)^{b/a} \right)^{1/b}$$

for any $r \times c$ matrix $\mathbf{M} = (m_{ij})$. In other words, we take the L_a-norm of each row in the matrix and then take the L_b-norm of the resulting vector of L_a-norms. We will primarily be concerned with the $L_{2,p}$-norm of the data matrix $\mathbf{X}$ as shown in Figure 5.

4.2.1 L_qL_p Support Vector Machines

Given a linearly separable set $\mathbf{X}$ of n labeled examples, we can solve the convex program

$$\begin{aligned} \min_{w} \quad & \|w\|_q \\ \text{s.t.} \quad & \frac{y^i(w \cdot x^i)}{\|x^i\|_p} \geq 1, \quad 1 \leq i \leq n. \end{aligned} \tag{1}$$

to maximize the L_qL_p margin. Observe that a solution $\hat{w}$ to this problem has $\gamma_{q,p}(\mathbf{X}, \hat{w}) = 1/\|\hat{w}\|_q$. We call an algorithm that outputs a solution to (1) an L_qL_p SVM due to the close relationship between this problem and the standard support vector machine.

If $\mathbf{X}$ is not linearly separable, we can introduce nonnegative slack variables in the usual way and solve

$$\begin{aligned} \min_{\substack{w, \\ \xi \geq 0}} \quad & \|w\|_q + C \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & \frac{y^i(w \cdot x^i)}{\|x^i\|_p} \geq 1 - \xi_i, \quad 1 \leq i \leq n. \end{aligned} \tag{2}$$

which is equivalent to minimizing the hinge loss with respect to an L_p-normalized data set using L_q-norm regularization on the weight vector space.

4.3 Generalization Bounds

In this section we give an upper bound on the generalization error of learning linear separators over distributions with large L_qL_p margins. The proof follows from combining a theorem of [18] with a new bound on the fat-shattering dimension of the class of linear separators with small L_q-norm. We begin with the following definitions.

Definition. For a set $\mathcal{F}$ of real-valued functions on X, a finite set $\{x^1, \dots, x^n\} \subseteq X$ is said to be *γ-shattered by $\mathcal{F}$* if there are real numbers $r_1, \dots, r_n$ such that for all $b = (b_1, \dots, b_n) \in \{-1, 1\}^n$, there is a function $f_b \in \mathcal{F}$ such that

$$f_b(x^i) \begin{cases} \geq r_i + \gamma & \text{if } b_i = 1 \\ \leq r_i - \gamma & \text{if } b_i = -1. \end{cases}$$

The *fat-shattering dimension of $\mathcal{F}$ at scale* γ, denoted $\text{fat}_{\mathcal{F}}(\gamma)$, is the size of the largest subset of X which is γ-shattered by $\mathcal{F}$.

Our bound on the fat-shattering dimension will use two lemmas analogous to Lemmas 11 and 12 in [91].

Lemma 2. *Let $\mathcal{F} = \{x \mapsto w \cdot x : \|w\|_q \leq \|W\|_q\}$ with $1 \leq p \leq \infty$. If the set $\{x^1, \dots, x^n\} \subseteq X^n$ is γ-shattered by $\mathcal{F}$ then every $b = (b_1, \dots, b_n) \in \{-1, 1\}^n$ satisfies $\|\sum_{i=1}^n b_i x^i\|_p \geq \frac{\gamma n}{\|W\|_q}$.*

Proof. The proof is identical to that of Lemma 11 in [91], replacing the radius $1/\|X\|_p$ of $\mathcal{F}$ in their lemma with $\|W\|_q$. □

The next lemma will depend on the following classical result from probability theory known as the Khintchine inequality.

Theorem 5 (Khintchine)**.** *If the random variable $\sigma = (\sigma_1, \dots, \sigma_n)$ is uniform over $\{-1, 1\}^n$ and $0 < p < \infty$, then any finite set $\{z_1, \dots, z_n\} \in \mathbb{C}$ satisfies*

$$A_p \sqrt{\sum_{i=1}^n |z_i|^2} \leq \left(\mathbb{E} \left[\Big| \sum_{i=1}^n \sigma_i z_i \Big|^p \right] \right)^{\frac{1}{p}} \leq B_p \sqrt{\sum_{i=1}^n |z_i|^2}$$

where A_p and B_p are constants depending only on p.

The precise optimal constants for A_p and B_p were found by [51], but for our purposes, it suffices to note that when $p \geq 1$ we have $1/2 \leq A_p \leq 1$ and $1 \leq B_p \leq \sqrt{p}$.

Lemma 3. *For any set* $\mathbf{X} = \{x^1, \ldots, x^n\} \subseteq X^n$ *and any* $1 \leq p < \infty$*, there is some* $b = (b_1, \ldots, b_n) \in \{-1, 1\}^n$ *such that* $\|\sum_{i=1}^n b_i x^i\|_p \leq B_p \|\mathbf{X}\|_{2,p}$.

Proof. We will bound the expectation of $\|\sum_{i=1}^n b_i x^i\|_p$ when $b = (b_1, \ldots, b_n)$ is uniformly distributed over $\{-1, 1\}^n$. We have

$$\begin{aligned}\mathbb{E}\left[\left\|\sum_{i=1}^n b_i x^i\right\|_p\right] &= \mathbb{E}\left[\left(\sum_{j=1}^d \Big|\sum_{i=1}^n \epsilon_i x_j^i\Big|^p\right)^{1/p}\right] \\ &\leq \left(\sum_{j=1}^d \mathbb{E}\left[\Big|\sum_{i=1}^n \epsilon_i x_j^i\Big|^p\right]\right)^{1/p} \\ &\leq \left(\sum_{j=1}^d B_p^p \left(\sum_{i=1}^n |x_j^i|^2\right)^{p/2}\right)^{1/p} \\ &= B_p \|\mathbf{X}\|_{2,p}\end{aligned}$$

where the first inequality is an application of Jensen's inequality and the second uses the Khintchine inequality. The proof is completed by noting that there must be some choice of b for which the value of $\|\sum_{i=1}^n b_i x^i\|_p$ is smaller than its expectation. □

We can use these two lemmas to give an upper bound on the fat-shattering dimension for any finite p.

Theorem 6. *Let* $\mathcal{F} = \{x \mapsto w \cdot x : \|w\|_q \leq \|W\|_q\}$ *with* $1 \leq p < \infty$*. If there is a constant* $C = C(d, p)$ *independent of* n *such that* $\|\mathbf{X}\|_{2,p} \leq C n^\alpha \|X\|_p$ *for any set* $\mathbf{X}$ *of* n *examples drawn from* D*, then*

$$\mathrm{fat}_{\mathcal{F}}(\gamma) \leq \left(\frac{C B_p \|W\|_q \|X\|_p}{\gamma}\right)^{\frac{1}{1-\alpha}}.$$

Proof. Combining Lemmas 2 and 3, we have that any set $\mathbf{X} = \{x^1, \ldots, x^n\} \subseteq X^n$ that is γ-shattered by $\mathcal{F}$ satisfies $\frac{\gamma n}{\|W\|_q} \leq B_p \|\mathbf{X}\|_{2,p} \leq C B_p n^\alpha \|X\|_p$. Solving for n gives us $n \leq (\frac{C B_p \|W\|_q \|X\|_p}{\gamma})^{1/(1-\alpha)}$ as an upper bound on the maximum size of any γ-shattered set. □

This bound extends and improves upon Theorem 8 in [91]. In their specific setting, $\|W\|_q = 1/\|X\|_p$, so we can directly compare their bound

$$\text{fat}_{\mathcal{F}}(\gamma) \leq \frac{2\log 4d}{\gamma^2} \tag{3}$$

for $2 \leq p \leq \infty$ to our bound

$$\text{fat}_{\mathcal{F}}(\gamma) \leq \left(\frac{CB_p}{\gamma}\right)^{1/(1-\alpha)} \tag{4}$$

for $1 \leq p < \infty$. Observe that by Minkowski's inequality, any set $\mathbf{X}$ satisfies $\|\mathbf{X}\|_{2,p} \leq n^{1/2}\|X\|_p$ if $p \geq 2$. In this case, (4) simplifies to $(B_p/\gamma)^2$ which is dimension-independent and improves upon (3) by a factor of $\log d$ when p is constant. When $1 \leq p < 2$, (3) does not apply, but (4) still gives a bound that can be small in many cases depending on the relationship between $\|\mathbf{X}\|_{2,p}$ and γ. We will give specific examples in Section 4.3.2.

The fat-shattering dimension is relevant due to the following theorem of [18] that relates the generalization performance of a classifier with large margin to the fat-shattering dimension of the associated real-valued function class at a scale of roughly the margin of the classifier.

Theorem 7 (Bartlett & Shawe-Taylor)**.** *Let $\mathcal{F}$ be a collection of real-valued functions on a set X and let D be a distribution over X. Let $\mathbf{X} = \{x^1, \ldots, x^n\}$ be a set of examples drawn i.i.d. from D with labels $y_i = h^*(x^i)$ for each i. With probability at least $1 - \delta$, if a classifier $h(x) = \operatorname{sign}(f(x))$ with $f \in \mathcal{F}$ satisfies $y_i f(x^i) \geq \gamma > 0$ for each $x^i \in \mathbf{X}$, then*

$$\mathsf{err}_D(h) \leq \frac{2}{n}\left(k \log \frac{8en}{k} \log(32n) + \log \frac{8n}{\delta}\right),$$

where $k = \text{fat}_{\mathcal{F}}(\gamma/16)$.

Now we can state and prove the following theorem which bounds the generalization performance of the L_qL_p SVM algorithm.

Theorem 8. *For any distribution D and target w^* with $\gamma_{q,p}(D, w^*) \geq \gamma_{q,p}$, if there is a constant $C = C(d, p)$ such that $\|\mathbf{X}\|_{2,p} \leq Cn^{\alpha} \|X\|_p$ for any set $\mathbf{X}$ of n examples from D then there is a polynomial time algorithm that outputs, with probability at least $1 - \delta$, a classifier h such that*

$$\mathsf{err}_D(h) = O\left(\frac{1}{n}\left(\left(\frac{CB_p}{\gamma_{q,p}}\right)^{\frac{1}{1-\alpha}} \log^2 n + \log\frac{n}{\delta}\right)\right).$$

Proof. By the definition of L_qL_p margin, there exists a w (namely, $w^*/\|w^*\|_q$) with $\|w\|_q = 1$ that achieves margin $\gamma_{q,p}$ with respect to D. This w has margin at least $\gamma_{q,p}$ with respect to any set $\mathbf{X}$ of n examples from D. A vector $\hat{w}$ satisfying these properties can be found in polynomial time by solving the convex program (1) and normalizing the solution. Notice that if the sample is normalized to have $\|x\|_p = 1$ for every $x \in \mathbf{X}$ then the L_qL_p margin of $\hat{w}$ does not change but becomes equal to the functional margin $y(\hat{w} \cdot x)$ appearing in the Theorem 7. Applying Theorem 6 with $\|W\|_q = 1$ and $\|X\|_p = 1$ yields $\text{fat}_{\mathcal{F}}(\gamma) \leq (\frac{CB_p}{\gamma})^{1/(1-\alpha)}$ and applying Theorem 7 to $\hat{w}$ gives us the desired result. □

Theorem 8 tells us that if the quantity

$$\left(\frac{CB_p}{\gamma_{q,p}}\right)^{\frac{1}{1-\alpha}} \tag{5}$$

is small for a certain choice of p and q then the L_qL_p SVM will have good generalization. This gives us a data-dependent bound, as (5) depends on data; specifically, C and α depend on the distribution D alone, while $\gamma_{q,p}$ depends on the relationship between D and the target w^*.

As mentioned, if $p \geq 2$ then we can use $C = 1$ and $\alpha = 1/2$ for any distribution, in which case the bound depends solely on the margin $\gamma_{q,p}$ (and to a lesser extent on B_p). If $p \leq 2$ then any set has $\|\mathbf{X}\|_{2,p} \leq n^{1/p} \|X\|_p$ (this follows by subadditivity of the function $z \mapsto z^{p/2}$ when $p \leq 2$) and we can obtain a similar dimension-independent bound with $C = 1$ and $\alpha = 1/p$. Achieving dimension independence for all distributions comes at the price of the bound becoming uninformative as $p \to 1$, as (5) simplifies to $(B_p/\gamma_{q,p})^q$ for these values. More interesting situations arise when we consider the quantity (5) for specific distributions, as we will show in Section 4.3.2.

4.3.1 Generalization Bounds in the Non-realizable Case

The results in Section 4.3 apply to the realizable case—that is, when the two classes are linearly separable by a positive "hard margin." When the data is not linearly separable, convex program (1) has no solution, but convex program (2) remains solvable and we may still achieve good generalization performance in the presence of a "soft margin" (some small margin violations exist in the data, but the majority of points will be far from the optimal separator). In this non-realizable case, we can still obtain generalization bounds analogous to Theorem 8, but they will include an additional dependence on how far the data is from being separable by a large margin (the hinge loss).

Bounds based on Rademacher complexity. The empirical Rademacher complexity of a class $\mathcal{F}$ of real-valued functions is

$$\mathcal{R}_n(\mathcal{F}) \;=\; \frac{1}{n}\,\mathbb{E}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{n}\sigma_i f(x^i)\right]$$

where $\sigma = (\sigma_1,\ldots,\sigma_n)$ is uniform over $\{-1,1\}^n$. In the case of linear functions $x \mapsto w\cdot x$ with $\|w\|_q \le \|W\|_q$, this is

$$\mathcal{R}_n(\mathcal{F}) \le \frac{B_p\,\|W\|_q\,\|\mathbf{X}\|_{2,p}}{n},$$

where we have applied Jensen's inequality and the Khintchine inequality as in Section 4.3. This result is a special case of Proposition 2 of [68]. If $\|\mathbf{X}\|_{2,p} \le Cn^{\alpha}\,\|X\|_p$, then this simplifies to

$$\mathcal{R}_n(\mathcal{F}) \le \frac{CB_p\,\|W\|_q\,\|X\|_p}{n^{1-\alpha}}$$

which can be used to bound the Rademacher complexity term in several standard generalization bounds such as those in terms of convex loss functions [17] or margins [70].

Bounds based on fat-shattering dimension. Theorem VII.14 of [94] gives a generalization error bound in terms of the fat-shattering dimension of the concept class $\mathcal{F}$ and the sum of the slack variables ξ in convex program (2). The bound is of the form

$$\mathsf{err}(h) \leq \tilde{O}\left(\frac{1}{n}\left(\text{fat}_{\mathcal{F}}(\gamma/16) + \frac{1}{\gamma}\sum_{i=1}^{n}\hat{\xi}_i\right)\right) \tag{6}$$

where h is the classifier corresponding to a solution $\hat{w}$ of (2) and where $\hat{\xi}_i = \max(0, \gamma - y^i(\hat{w} \cdot x^i))$. We can then use our bound from Theorem 6 to obtain a bound analogous to Theorem 8.

4.3.2 Examples

Here we will give some specific learning problems showing when large margins can be helpful and when they are not helpful. We focus on the $p \leq 2$ case, as large margins are always helpful when $p \geq 2$.

Example 1. Unhelpful margins. First, let D_1 be the uniform distribution over the standard basis vectors in $\mathbb{R}^d$ and let w^* be any weight vector in $\{-1, 1\}^d$. In this case $\gamma_{q,p} = d^{-1/q}$, which is a large margin for small p. If $n \leq d$, then $\|\mathbf{X}\|_{2,p}$ is roughly $n^{1/p}\|X\|_p$ (ignoring log factors), and (5) simplifies to $B_p^q d$. We could also choose to simplify (5) using $C = d^{1/p}$ and $\alpha = 0$, which gives us $B_p d$. Either way, the bound in Theorem 8 becomes $\tilde{O}(d/n)$ which is uninformative since $n \leq d$. If we take $n \geq d$, we can still obtain a bound of $\tilde{O}(d/n)$, but this is the same as the worst-case bound based on VC dimension, so the large margin has no advantage. In fact, this example provides a lower bound: even if an algorithm knows the distribution D_1 and is allowed a 1/2 probability of failure, an error of ϵ cannot be guaranteed with fewer than $(1 - 2\epsilon)d$ examples because any algorithm can hope for at best an error rate of 1/2 on the examples it has not yet seen.

Example 2. Helpful margins. As another example, divide the d coordinates into $k = o(d)$ disjoint blocks of equal size and let D_2 be the uniform distribution over

```
w*:    ++-+++-+--+--++---

D:     000100000000000000   +
       000000000001000000   -
       000000000100000000   -
                ⋮
```

Figure 6: A diagram showing a case with unhelpful margins as in Example 1.

Table 1: Summary of key parameters discussed in Example 1.

p	$\gamma_{q,p}(w^*)$	$\|\mathbf{X}\|_{2,p}$	sample complexity
1	1	$\sqrt{dn}$	$\tilde{O}(d/\epsilon)$
2	$1/\sqrt{d}$	$\sqrt{n}$	$\tilde{O}(d/\epsilon)$
∞	$1/d$	$\sqrt{n/d}$	$\tilde{O}(d/\epsilon)$

examples that have 1's for all coordinates within some block and 0's elsewhere. Taking w^* to be a vector in $\{-1, 1\}^d$ that has the same sign within each block, we have $\gamma_{q,p} = k^{-1/q}$. If $k < n < d$ then $\|\mathbf{X}\|_{2,p}$ is roughly $k^{1/p-1/2}\sqrt{n}\,\|X\|_p$, and (5) simplifies to $B_p^2 k$. When $k = o(d)$ this is a significant improvement over worst case bounds for any constant choice of p.

Example 3. An advantage for $p < 2$. Consider a distribution that is a combination of the previous two examples: with probability 1/2 it returns an example drawn from D_1 and otherwise returns an example from D_2. By including the basis vectors, we have made the margin $\gamma_{q,p} = d^{-1/q}$ but as long as $k = o(d)$ the bound on $\|\mathbf{X}\|_{2,p}$ does not change significantly from Example 2, and we can still use $C = k^{1/p-1/2}$ and $\alpha = 1/2$. Now (5) simplifies to $B_p^2 k$ for $p = 1$, but becomes $B_p^2 k^{2/p-1} d^{2/q}$ in general. When $k = \sqrt{d}$ this gives us an error bound of $\tilde{O}(\sqrt{d}/n)$ for $p = 1$ but $\tilde{O}(d/n)$ or worse for $p \geq 2$. While this upper bound does not imply that generalization error will be worse for $p \geq 2$ than it is for $p = 1$, we show in the next section that for a slightly modified version of this distribution we can obtain sample complexity lower bounds for large margin algorithms with $p \geq 2$ that are significantly greater than the upper

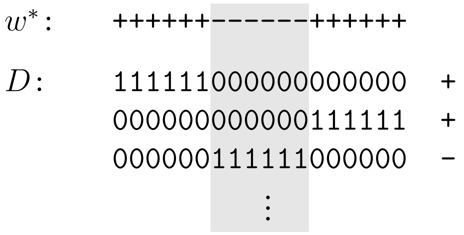


Figure 7: A diagram showing a case with helpful margins as in Example 2.

Table 2: Summary of key parameters discussed in Example 2.

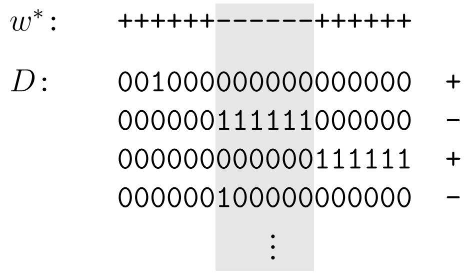
p	$\gamma_{q,p}(w^*)$	$\|\mathbf{X}\|_{2,p}$	sample complexity
1	1	$\sqrt{kn}$	$\tilde{O}(k/\epsilon)$
2	$1/\sqrt{k}$	$\sqrt{n}$	$\tilde{O}(k/\epsilon)$
∞	$1/k$	$\sqrt{n/k}$	$\tilde{O}(k/\epsilon)$

bound for $p = 1$.

4.4 The Case For $L_\infty L_1$ Margins

Here we give a family of learning problems to show the benefits of using $L_\infty L_1$ margins over other margins. We do this by defining a distribution D over unlabeled examples in $\mathbb{R}^d$ that can be consistently labeled by a variety of potential target functions w^*. We then consider a family of large $L_q L_p$ margin concept classes W_p and bound the sample complexity of learning a concept in W_p using covering number bounds. We show that learning W_1 can be much easier than learning W_p for $p > 1$; for example, with certain parameters for D having $O(\sqrt{d})$ examples is sufficient for learning W_1, while learning any other W_p requires $\Omega(d)$ examples.

Specifically, let $W_p = \{w \in \mathbb{R}^d : \|w\|_\infty = 1,\ \gamma_{q,p}(w) \geq \gamma_{q,p}(w^*)\}$, where w^* maximizes the $L_\infty L_1$ margin with respect to D. We restrict our discussion to weight vectors with unit L_∞ norm because normalization does not change the margin (nor does it affect the output of the corresponding classifier). Let the covering number $\mathcal{N}(\epsilon, W, D)$ be the size of the smallest set $V \subseteq W$ such that for every $w \in W$ there



```
w*:    ++++++------++++++

D:     001000000000000000   +
       000000111111000000   -
       000000000000111111   +
       000000100000000000   -
                 ⋮
```

Figure 8: A diagram showing a case with an advantage for $p < 2$ as in Example 3.

Table 3: Summary of key parameters discussed in Example 3.

p	$\gamma_{q,p}(w^*)$	$\|\mathbf{X}\|_{2,p}$	sample complexity
1	1	$\sqrt{kn}$	$\tilde{O}(k/\epsilon)$
2	$1/\sqrt{d}$	$\sqrt{n}$	$\tilde{O}(d/\epsilon)$
∞	$1/d$	$\sqrt{n/k}$	$\tilde{O}(d^2/(k\epsilon))$

exists a $v \in V$ with $d_D(w, v) := \Pr_{x \sim D}[\mathrm{sign}(w \cdot x) \neq \mathrm{sign}(v \cdot x)] \leq \epsilon$.

Define the distribution D over $\{0,1\}^d$ as follows. Divide the d coordinates into k disjoint blocks of size d/k (assume $d/(2k)$ is an odd integer). Flip a fair coin. If heads, pick a random block and return an example with exactly $d/(2k)$ randomly chosen coordinates set to 1 within the chosen block and all other coordinates set to 0. If tails, return a standard basis vector (exactly one coordinate set to 1) chosen uniformly at random. The target function will be determined by any weight vector w^* achieving the maximum $L_\infty L_1$ margin with respect to D. As we will see, w^* can be any vector in $\{-1,1\}^d$ with complete agreement within each block.

We first give an upper bound on the covering number of W_1.

Proposition 1. *For any $\epsilon > 0$, $\mathcal{N}(\epsilon, W_1, D) \leq 2^k$.*

Proof. Let V_k be the following set. Divide the d coordinates into k disjoint blocks of size d/k. A vector $v \in \{-1,1\}^d$ is a member of V_k if and only if each block in v is entirely $+1$'s or entirely -1's. We will show that $W_1 = V_k$, and since $|V_k| = 2^k$ we will have $\mathcal{N}(\epsilon, W_1, D) \leq 2^k$ for any ϵ.

Note that by Hölder's inequality, $\gamma_{q,p}(w) \leq 1$ for any $w \in \mathbb{R}^d$. For any $w \in V_k$ and any example x drawn from D, we have $|w \cdot x| = \|x\|_1$, so $\gamma_{\infty,1}(w) = 1$, the maximum margin. If $w \notin V_k$ then either $w \notin \{-1,1\}^d$ or w has sign disagreement within at least one of the k blocks. If $w \notin \{-1,1\}^d$ then $\gamma_{\infty,1}(w) = \min_x |w \cdot x| / \|x\|_1 \leq \min_i |w \cdot e_i| = \min_i |w_i| < 1$. If w has sign disagreement within a block then $|w \cdot x| < \|x\|_1$ for any x with 1's in disagreeing coordinates of w, and this results in a margin strictly less than 1. □

Next we will give lower bounds on the covering numbers for W_p with $p > 1$. In the following let $H(\alpha) = -\alpha \log(\alpha) - (1-\alpha)\log(1-\alpha)$, the binary entropy function.

Proposition 2. *If* $1 < p < \infty$ *then for any* $\epsilon > 0$, $\mathcal{N}(\epsilon, W_p, D) \geq 2^{(1/2 - H(2\epsilon))d - k^{1/q}(d/2)^{1/p} - k}$.

Proof. First we show that $|W_p| \geq 2^{d/2 - k^{1/q}(d/2)^{1/p} - k}$. Any $w^* \in W_1$ has margin $\gamma_{q,p}(w^*) = d^{-1/q}$, so $W_p = \{w \in \mathbb{R}^d : \gamma_{q,p}(w) \geq d^{-1/q}\}$. Note that $W_p \subseteq \{-1,1\}^d$ because if $w \notin \{-1,1\}^d$ then $\gamma_{q,p}(w) \leq \min_i |w \cdot e_i| / \|w\|_q = \min_i |w_i| / \|w\|_q < d^{-1/q}$. Let $w \in \{-1,1\}^d$ be a weight vector such that in each block there are at least $d/k - r$ positive values and at most r negative values or vice versa (there are at most r values with whichever sign is in the minority). Clearly w has large margin with respect to any of the basis vectors drawn from D. For the rest of D we have $\inf_x |w \cdot x| = \max(1, n/(2k) - 2r)$, so $w \in W_p$ if and only if $\max(1, d/(2k) - 2r) \geq (d/(2k))^{1/p}$. For $p < \infty$ and $d > 2k$, this happens if and only if $r \leq \frac{1}{2}(\frac{d}{2k} - (\frac{d}{2k})^{1/p})$. Letting $r^* = \lfloor \frac{1}{2}(\frac{d}{2k} - (\frac{d}{2k})^{1/p}) \rfloor$, we have $|W_p| = (2\sum_{i=0}^{r^*} \binom{d/k}{i})^k \geq 2^{d/2 - k^{1/q}(d/2)^{1/p} - k}$.

Now we can lower bound the covering number using a volume argument by noting that if m is the cardinality of the largest ϵ-ball around any $w \in W_p$ then $\mathcal{N}(\epsilon, W_p, D) \geq |W_p|/m$. For any pair $w, w' \in W_p$, $d_D(w, w') \geq h(w, w')/(2d)$ where $h(w, w')$ is the hamming distance (number of coordinates in which w and w' disagree). Therefore, in order for $d_D(w, w') \leq \epsilon$ we need $h(w, w') \leq 2\epsilon d$. For any $w \in W_p$ the number of w' such that $h(w, w') \leq 2\epsilon d$ is at most $\sum_{i=0}^{\lfloor 2\epsilon d \rfloor} \binom{d}{i} \leq 2^{H(2\epsilon)d}$. It follows that $\mathcal{N}(\epsilon, W_p, D) \geq$

$|W_p|/2^{H(2\epsilon)d} \geq 2^{d/2 - H(2\epsilon)d - k^{1/q}(d/2)^{1/p} - k}$. □

Proposition 3. *For any* $\epsilon > 0$, $\mathcal{N}(\epsilon, W_\infty, D) \geq 2^{(1-H(2\epsilon))d}$.

Proof. First we show that $W_\infty = \{-1,1\}^d$. Any $w^* \in W_1$ has margin $\gamma_{1,\infty}(w^*) = 1/d$, so $W_\infty = \{w \in \mathbb{R}^d : \gamma_{1,\infty}(w) \geq 1/d\}$. For any $w \in \{-1,1\}^d$ and example x drawn from D, we have $\|w\|_1 = d$, $\|x\|_\infty = 1$, and $|w \cdot x| \geq 1$ (since x has an odd number of coordinates set to 1) resulting in margin $\gamma_{1,\infty}(w) \geq 1/d$. If $w \notin \{-1,1\}^d$ then $\gamma_{1,\infty}(w) = \min_x |w \cdot x| / \|w\|_1 \leq \min_i |w \cdot e_i| / \|w\|_1 = \min_i |w_i| / \|w\|_1 < 1/d$.

To bound the covering number, we use the same volume argument as above. Again, the size of the largest ϵ-ball around any $w \in W_\infty$ is at most $2^{H(2\epsilon)d}$ (since this bound only requires that every pair $w, w' \in W_\infty$ has $d_D(w,w') \geq h(w,w')/(2d)$). It follows that $\mathcal{N}(\epsilon, W_\infty, D) \geq 2^d / 2^{H(2\epsilon)d}$. □

Using standard distribution-specific sample complexity bounds based on covering numbers [60], we have an upper bound of $O((1/\epsilon)\ln(\mathcal{N}(\epsilon, W, D)/\delta))$ and lower bound of $\ln((1-\delta)\mathcal{N}(2\epsilon, W, D))$ for learning, with probability at least $1-\delta$, a concept in W to within ϵ error. Thus, we have the following results for the sample complexity m of learning W_p with respect to D. If $p = 1$ then

$$m \leq O\left(\frac{1}{\epsilon}\left(k + \ln\frac{1}{\delta}\right)\right),$$

if $1 < p < \infty$ then

$$m \geq \left(\frac{1}{2} - H(4\epsilon)\right)d - k^{1/q}\left(\frac{d}{2}\right)^{1/p} - k + \ln(1-\delta),$$

and if $p = \infty$ then

$$m \geq (1 - H(4\epsilon))\,d + \ln(1-\delta).$$

For appropriate values of k and ϵ relative to d, the the sample complexity can be much smaller for the $p = 1$ case. For example, if $k = O(d^{1/4})$ and $\Omega(d^{-1/4}) \leq \epsilon \leq 1/40$, then (assuming δ is a small constant) having $O(\sqrt{d})$ examples is sufficient for learning W_1 while at least $\Omega(d)$ examples are required to learn W_p for any $p > 1$.

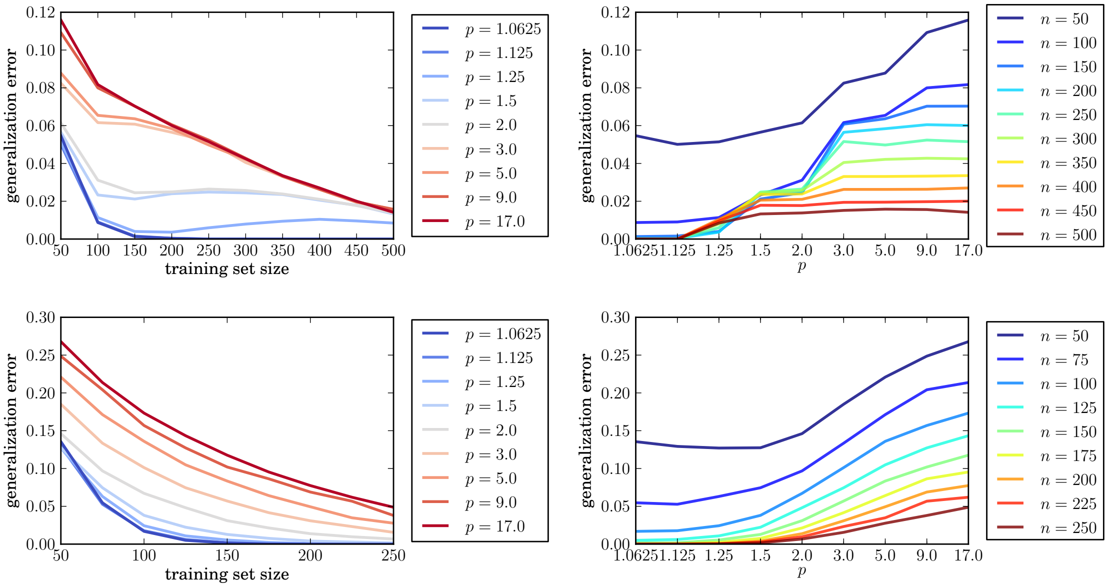


Figure 9: Synthetic data results for blocks distribution (top) and Gaussian with margin (bottom). The left column plots generalization error (averaged over 500 trials with different training sets) versus number of training examples n while the right column plots error versus p.

4.5 Experiments

We performed two empirical studies to support our theoretical results. First, to give further evidence that using $L_\infty L_1$ margins can lead to faster learning than other margins, we ran experiments on both synthetic and real-world data sets. Using the $L_q L_p$ SVM formulation defined in (1) for linearly separable data and the formulation defined in (2) for non-separable data, both implemented using standard convex optimization software, we ran our algorithms for a range of values of p and a range of training set sizes n on each data set. We report several cases in which maximizing the $L_\infty L_1$ margin results in faster learning (i.e., smaller sample complexity) than maximizing other margins.

Figure 9 shows results on two synthetic data sets. One is generated using the "Blocks" distribution family from Section 4.4 with $d = 90$ and $k = 9$. The other uses examples generated from a standard Gaussian distribution in $\mathbb{R}^{100}$ subject to having

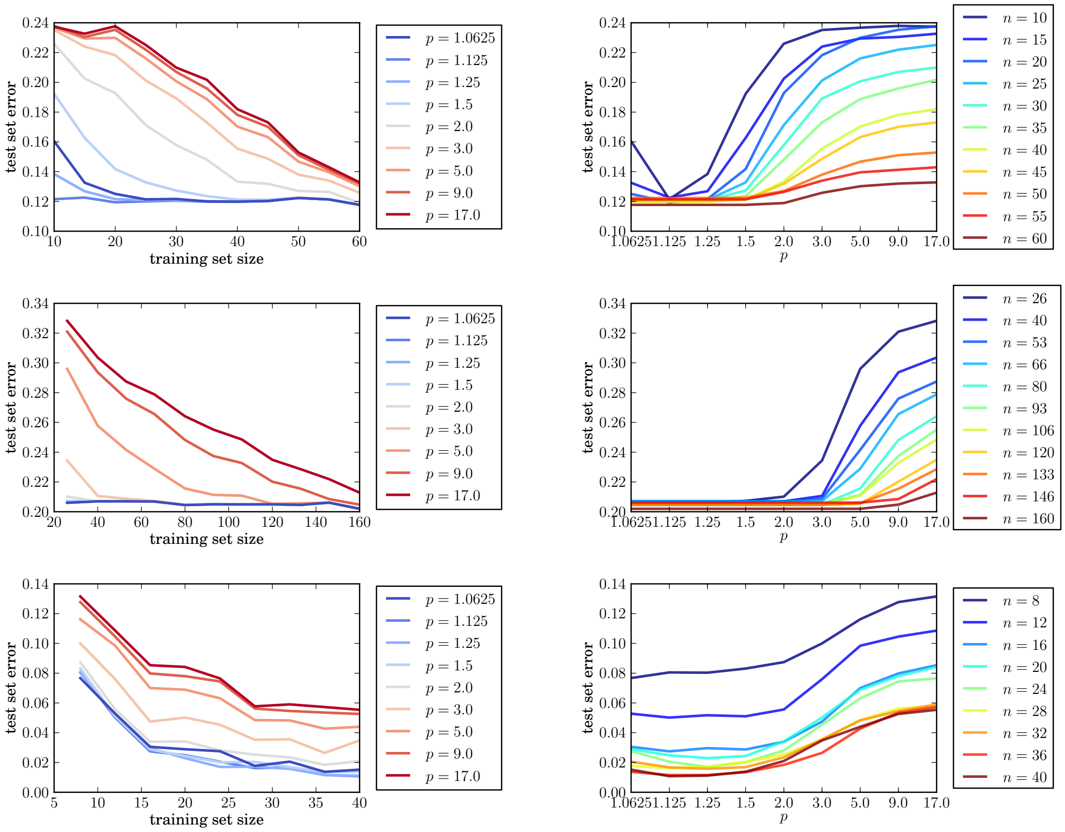


Figure 10: Results for Fertility data (top), SPECTF Heart data (middle), and CNAE-9 (bottom). The left column plots error (averaged over 100 trials with different training sets and tested on all non-training examples) versus number of training examples n while the right column plots error versus p.

$L_\infty L_1$ margin at least 0.075 with respect to a fixed random target vector in $\{-1,1\}^d$ (in other words, Gaussian samples with margin smaller than 0.075 are rejected). In both cases, the error decreases much faster for $p < 2$ than for large p.

Figure 10 shows results on three data sets from the UCI Machine Learning Repository [6]. The Fertility data set consists of 100 training examples in $\mathbb{R}^{10}$, the SPECTF Heart data set has 267 examples in $\mathbb{R}^{44}$, and we used a subset of the CNAE-9 data set with 240 examples in $\mathbb{R}^{857}$. In all three cases, better performance was achieved by algorithms with $p < 2$ than by those with $p > 2$.

The goal of our second experiment was to determine, for real-world data, what parameter α can be used in the bound on $\|\mathbf{X}\|_{2,p}$ in Theorem 8. Specifically, for each

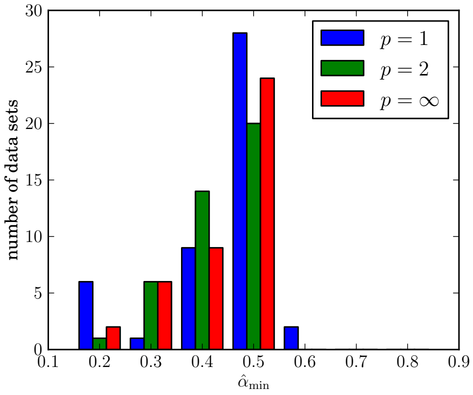


Figure 11: A histogram showing the values of $\hat{\alpha}_{\min}$ on 47 data sets from the UCI repository.

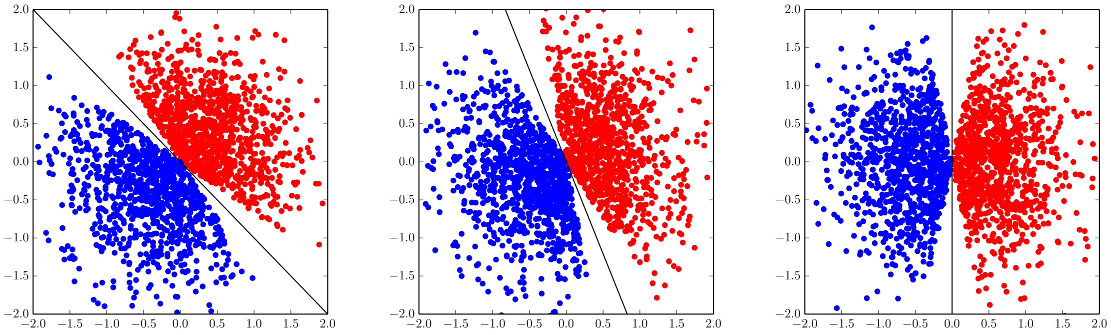
data set we want to find $\alpha_{\min} = \inf\{\alpha : \|\mathbf{X}\|_{2,p} \leq n^{\alpha} \|X\|_p\}$, the smallest value of α so the bound holds with $C = 1$. Recall that for $p = 1$, $\alpha_{\min}$ can theoretically be as great as 1, while for $p \geq 2$ it is at most 1/2. We would like to see whether $\alpha_{\min}$ is often small in real data sets or whether it is close to the theoretical upper bound.

We can estimate $\alpha_{\min}$ for a given set of data by creating a sequence $\{\mathbf{X}_m\}_{m=1}^{n}$ of data matrices by adding to the matrix one point from the data set at a time. For each point in the sequence we can compute

$$\alpha_m = \frac{\log(\|\mathbf{X}_m\|_{2,p} / \|X\|_p)}{\log m},$$

a value of α that realizes the bound with equality for this particular data matrix. We repeat this process T times, each with a different random ordering of the data, to find T sequences α_m^i, where $1 \leq i \leq T$ and $1 \leq m \leq n$. We can then compute $\hat{\alpha}_{\min} = \max_{i,m} \alpha_m^i$, a value of α which realizes the bound for every data matrix considered and which causes the bound to hold with equality in at least one instance.

Figure 11 shows a histogram of the resulting estimates on a variety of data sets and for three values of p. Notice that in the vast majority of cases, the estimate of $\alpha_{\min}$ is less than 1/2. As expected there are more values above 1/2 for $p = 1$ than for $p \geq 2$, but none of the estimates were above 0.7. This gives us evidence that many



(a) $p = 1$ (b) $p = 2$ (c) $p = \infty$

Figure 12: Margin-conditional Gaussian distributions in $\mathbb{R}^2$ with margin 0.2. In (a), the margin sizes are $\gamma_{\infty,1} = 0.2$, $\gamma_{2,2} = 0.196$, and $\gamma_{1,\infty} = 0.167$. In (b), the margin sizes are $\gamma_{\infty,1} = 0.156$, $\gamma_{2,2} = 0.2$, and $\gamma_{1,\infty} = 0.156$. In (c), the margin sizes are $\gamma_{\infty,1} = 0.167$, $\gamma_{2,2} = 0.196$, and $\gamma_{1,\infty} = 0.2$.

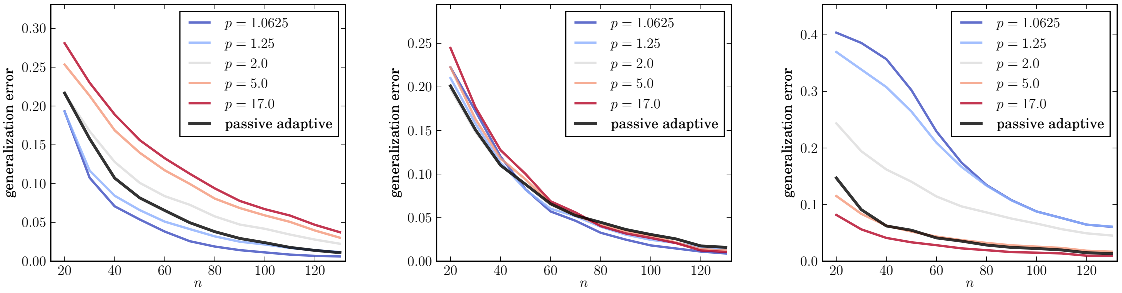
real data sets are much more favorable for learning with large $L_\infty L_1$ margins than the worst-case bounds may suggest.

4.6 Active Learning with $L_q L_p$ Margins

Here we show how active learning can be used to discover structure in data. In other words, if the underlying data distribution satisfies a large margin condition, active learning can be used to approximately determine the "best" p and q to use.

4.6.1 Passive Adaptation

Without active learning, we can passively determine the optimal margin parameters in the following manner. We first train one $L_q L_p$ SVM for each of several values of p and q spanning the entire margin spectrum. For each resulting linear separator, we then test how large its $L_q L_p$ margin is with respect to the training data (we assume linearly separable data for now as we will deal with the non-realizable case later). The separator that resulted in the largest margin is then used as the final hypothesis. This process of training several $L_q L_p$ SVMs and choosing the one resulting in the largest margin we refer to as the max-margin adaptation strategy.



(a) $p = 1$

(b) $p = 2$

(c) $p = \infty$

Figure 13: Passively adapting to the optimal margin parameters for three cases of margin-conditional Gaussian data. The caption labels the value of p used in the L_qL_p margin of the data distribution.

Figure 13 shows how on synthetic data this procedure consistently results in selecting hypotheses that perform nearly as well as if the correct p and q were known ahead of time. Each point in Figure 13 is an average over 50 independent trials, each with a different test set of 2000 examples. Examples are drawn from a standard Gaussian distribution in $\mathbb{R}^{25}$ conditioned on the L_qL_p margin with respect to target separator w^* being at least 0.05. We refer to this type of distribution as a *margin-conditional Gaussian* distribution. The examples are labeled with no noise according to $w^* \in \{0,1\}^{25}$. We give results for three cases: (1) $p = 1$, $q = \infty$, and $\|w^*\|_1 = 25$, (2) $p = 2$, $q = 2$, and $\|w^*\|_1 = 12$, and (3) $p = \infty$, $q = 1$, and $\|w^*\|_1 = 1$. See Figure 12 for samples of this type of distribution in $\mathbb{R}^2$ and with a margin of 0.2. Note that the distributions are designed so that the margin is larger for one set of margin parameters than for any other choice of margin parameters.

4.6.2 Active Adaptation in the Realizable Case

While the above passive procedure can make a small correction toward the optimal margin parameters, we hope to improve upon this by querying labels in a manner that allows us to more fully take advantage of the large margin. To activize the L_qL_p SVM, we apply the simple margin query strategy of Tong & Koller [103]. We note

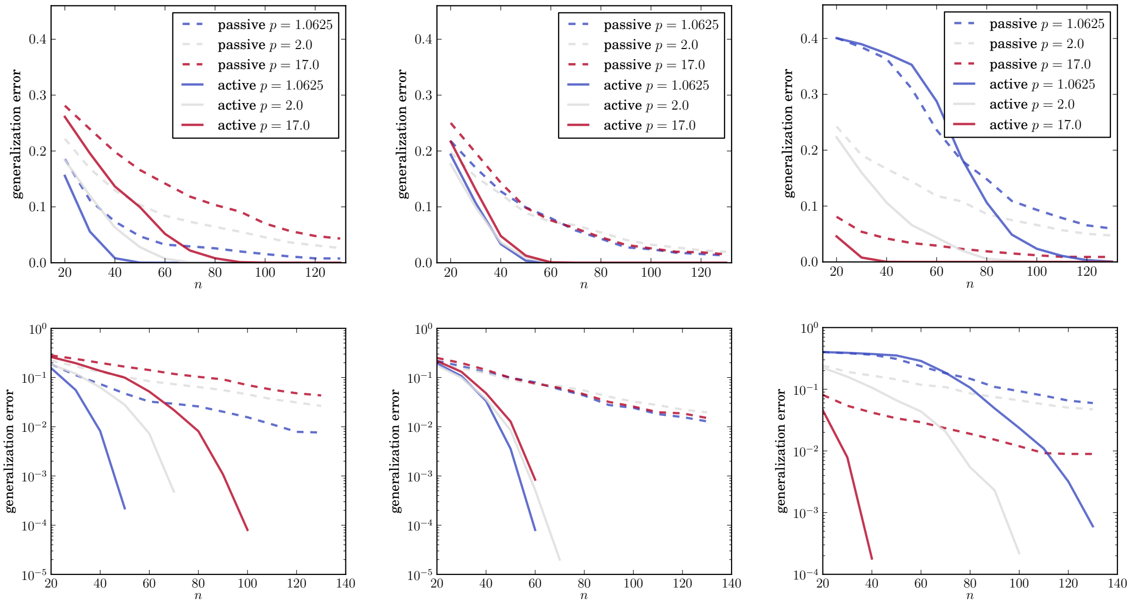
Algorithm 4 Active L_qL_p SVM

```
input Unlabeled data $U$, label budget $b$, margin parameters $p, q$
$L \leftarrow \emptyset$
$w^{(0)} \leftarrow \mathbf{1}$
for $i = 1$ to $b$ do
    $x^{(i)} \leftarrow \operatorname{argmin}_{x \in U \setminus L} \gamma_{q,p}(x, w^{(i-1)})$
    $y^{(i)} \leftarrow$ label of $x^{(i)}$
    $L \leftarrow L \cup \{(x^{(i)}, y^{(i)})\}$
    $w^{(i)} \leftarrow$ solution to (1) on data $L$ with margin parameters $p, q$
return $x \mapsto \operatorname{sign}(w^{(b)} \cdot x)$
```

that their ratio margin and max-min margin strategies may also work well for this task, as may the margin-based active learning algorithm of Balcan et al. [14]. While margin-based strategies such as these are not necessarily guaranteed to work well in general [37], there are several examples in the literature of cases in which they do work well [103, 61].

Specifically, we modify the simple margin strategy to use the L_qL_p margin instead of the typical L_2 margin. In each iteration, this algorithm trains an L_qL_p SVM on the current set of labeled examples and then queries the label of the example that has the least L_qL_p margin with respect to the separator found by the SVM. This process is repeated until the given label budget is reached. We refer to this algorithm, detailed in Algorithm 4, as the *active L_qL_p SVM* (with fixed margin parameters).

Figure 14 shows how this family of algorithms compares to passive learning on realizable data with various values of p and q. We again use a margin-conditional Gaussian with margin 0.05. Each point in Figure 14 is an average over 25 independent trials, each with a different test set of 2000 examples. When viewed on a logarithmic scale, it is clear that the learning rates for active learning are exponentially faster than those of passive learning, as we might expect given the data distribution. We can also see that knowing (or perhaps guessing) the correct margin parameters allows both the passive and active algorithms to perform better than using incorrect margin parameters, especially when the data-generating distribution uses parameters on the

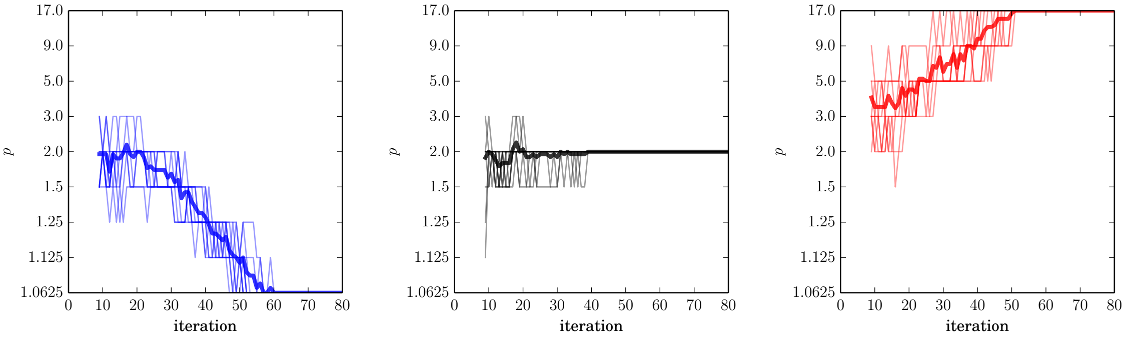


(a) $p = 1$ (b) $p = 2$ (c) $p = \infty$

Figure 14: Generalization error versus number of labeled examples used for passive and active $L_q L_p$ SVMs using fixed margin parameters on three types of realizable margin-conditional Gaussian data (caption indicates distribution parameters). Error is shown on both a linear scale (top) and a logarithmic scale (bottom).

Algorithm 5 Active adaptive $L_q L_p$ SVM

```
    **input** Unlabeled data $U$, label budget $b$, discretization $P$ of $[1, \infty]$
    $L \leftarrow \emptyset$
    $p^{(0)} \leftarrow 2$
    $w^{(0)} \leftarrow \mathbf{1}$
    **for** $i = 1$ **to** $b$ **do**
        $x^{(i)} \leftarrow \operatorname{argmin}_{x \in U \setminus L} \gamma_{q^{(i-1)}, p^{(i-1)}}(x, w^{(i-1)})$
        $y^{(i)} \leftarrow$ label of $x^{(i)}$
        $L \leftarrow L \cup \{(x^{(i)}, y^{(i)})\}$
        **for all** $p \in P$ **do**
            $w_p \leftarrow$ solution to (1) on data $L$ with margin parameters $p, q$
        $p^{(i)} \leftarrow \operatorname{argmax}_{p \in P} \gamma_{q,p}(L, w_p)$
        $w^{(i)} \leftarrow w_{p^{(i)}}$
    **return** $x \mapsto \operatorname{sign}(w^{(b)} \cdot x)$
```



(a) $p = 1$ (b) $p = 2$ (c) $p = \infty$

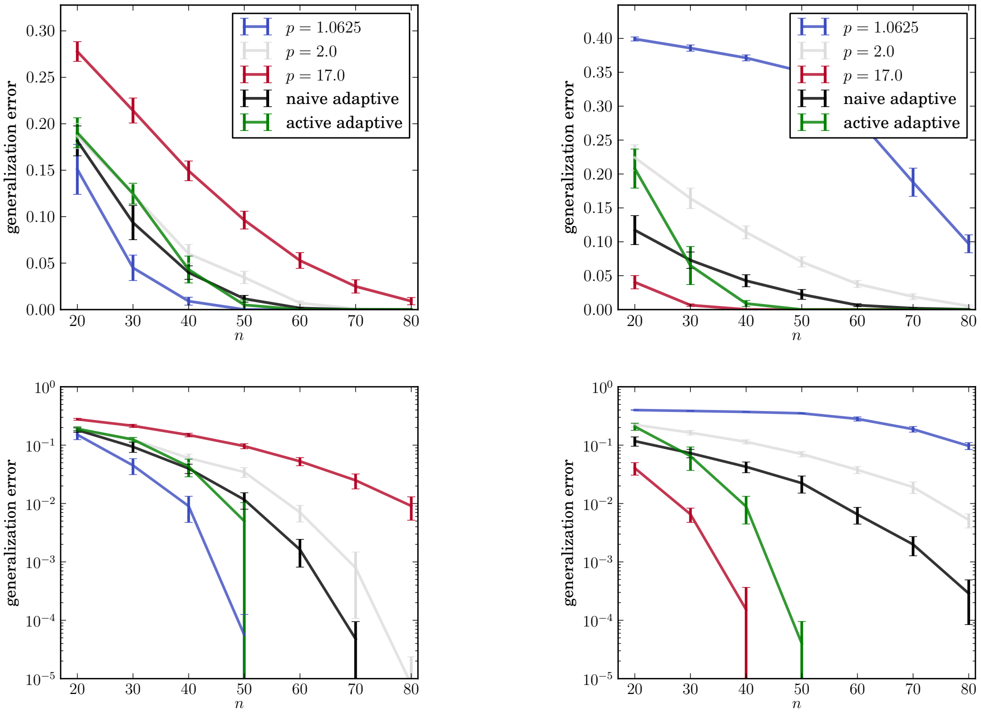
Figure 15: Margin parameters selected by the active adaptive algorithm on margin-conditional Gaussian data (caption indicates distribution parameters). Thin lines show the value of p selected in each iteration of the algorithm for 10 separate trials. Thick lines show the average p selected over all 10 trials.

extreme ends of the margin spectrum.

We can make our active L_qL_p SVM adapt to the optimal margin parameters by using the max-margin adaptation strategy after each query. Specifically, after each new label has been queried, we train on the current set of labeled data several L_qL_p SVMs with a wide range of margin parameters. We compute the margin for each of the resulting classifiers with respect to the labeled data, and choose the hypothesis with the largest margin to use for the next iteration. We refer to this type of algorithm, detailed in Algorithm 5, as *active adaptive.*

Figure 15 shows the values of p selected by the active adaptive algorithm on the same margin-conditional Gaussian data as above. Selections begin around $p = 2$ regardless of data, but they quickly approach the correct value in every trial before 60 examples are queried. We note that the algorithm appears to converge fastest in the $p = 2$ case and converges more quickly in the $p = \infty$ case than in the $p = 1$ case.

Note that the query strategy is somewhat self-correcting in the sense that selecting the incorrect margin parameters in one iteration makes it more likely to select the correct parameters in future iterations. This seems to be due to the dual nature of minimizing the margin for querying and maximizing the margin for model selection.



(a) $p = 1$

(b) $p = \infty$

Figure 16: Generalization error versus number of labeled examples used for active L_qL_p SVM, naïve adaptive, and active adaptive algorithms on realizable margin-conditional Gaussian data (caption indicates distribution parameters). Error is shown on both a linear scale (top) and a logarithmic scale (bottom). Error bars represent two standard errors, or approximately a 95% confidence interval.

That is, querying an example that minimizes $\gamma_{b,a}$ will shrink the margins of all the separators trained in the next iteration, but it will have the biggest shrinkage effect on the L_bL_a SVM, making the L_bL_a SVM less likely to be selected. However, the increase in sample size will eventually have a larger effect than this self-correction, so the bias introduced is not prohibitive. In fact, by encouraging some amount of exploration early on, it helps prevent the algorithm from compounding early mistakes.

Figure 16 shows how the active adaptive algorithm compares to fixed margin algorithms on realizable margin-conditional Gaussian data. Each point in Figure 16 is an average over 25 independent trials, each with a different test set of 5000 examples. The adaptive algorithm is competitive with the fixed-margin active SVMs

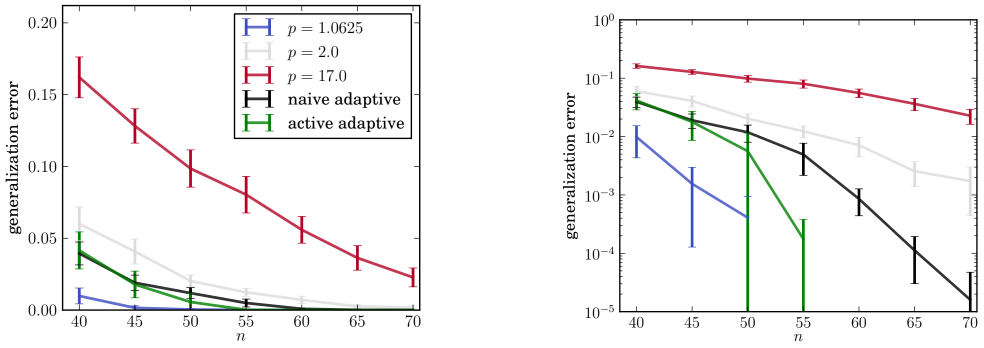


Figure 17: A more finely grained experiment showing the statistically significant advantage for the active adaptive algorithm in the $p = 1$ case. Error is shown on both a linear scale (left) and a logarithmic scale (right). Error bars represent two standard errors, or approximately a 95% confidence interval.

and significantly outperforms fixed-margin SVMs with a poor choice of margin parameters. In order to show that the adaptivity is due to the active query ability rather than simply the choice of margin used in training the final classifier, we also compare to a naïve adapter that makes active queries with respect to a fixed margin ($p = q = 2$) until the label budget is reached, and then attempts to find the optimal margin parameters (again by the max-margin strategy) for use with the final classifier. Figures 16b and 17 clearly show a statistically significant advantage for the active adapter over the naïve adapter. This is remarkable, as it shows that, without access to privileged information, the adaptive algorithm is able to learn how to shape its data distribution in a way that leads to better performance.

4.6.3 Active Adaptation in the Non-realizable Case

In the non-realizable case, we can no longer rely on the max-margin adaptation strategy because noisy examples make the hard margin meaningless. Instead, we use Algorithm 6 which chooses the separator in each iteration that minimizes

$$\min_{\gamma} \frac{1}{\gamma} \left(1 + \sum_{i=1}^{n} \hat{\xi}_i \right)$$

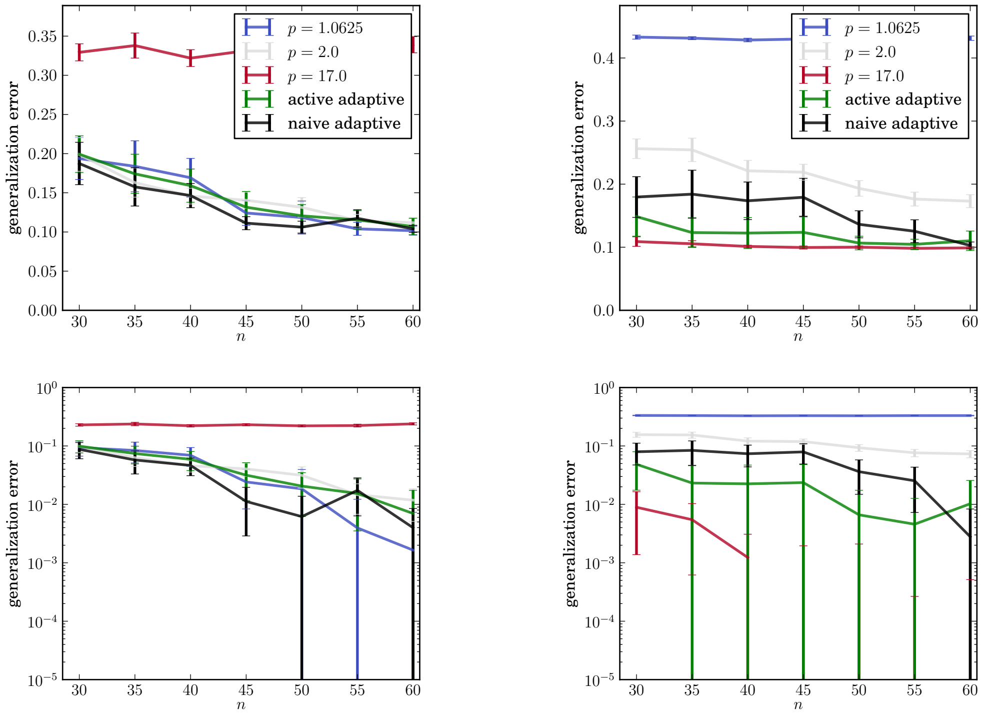
Algorithm 6 Active adaptive L_qL_p soft SVM

```
input Unlabeled data $U$, label budget $b$, discretization $P$ of $[1,\infty]$, tradeoff $C$
$L \leftarrow \emptyset$
$p^{(0)} \leftarrow 2$
$w^{(0)} \leftarrow \mathbf{1}$
for $i = 1$ to $b$ do
    $x^{(i)} \leftarrow \text{argmin}_{x \in U \setminus L}\, \gamma_{q^{(i-1)},p^{(i-1)}}(x, w^{(i-1)})$
    $y^{(i)} \leftarrow$ label of $x^{(i)}$
    $L \leftarrow L \cup \{(x^{(i)}, y^{(i)})\}$
    for all $p \in P$ do
        $w_p \leftarrow$ solution to (2) on data $L$ with margin parameters $p, q$ and tradeoff $C$
    $p^{(i)} \leftarrow \text{argmin}_{p \in P} \min_{\gamma} \frac{1}{\gamma}\left(1 + \sum_{i=1}^{|L|} \max(0, \gamma - y^{(i)}(w_p \cdot x^{(i)}))\right)$
    $w^{(i)} \leftarrow w_{p^{(i)}}$
return $x \mapsto \text{sign}(w^{(b)} \cdot x)$
```

where $\hat{\xi}_i = \max(0, \gamma - y^i(\hat{w} \cdot x^i))$ for a solution $\hat{w}$ of (2). Minimizing this quantity is motivated by the upper bound (6) on generalization error in the non-realizable case. Intuitively, a large margin γ that results in a small hinge loss (total distance of margin violations) indicates good generalization, and the margin parameters used to generate this separator are more likely to be the correct choice. While we could minimize (6) exactly by estimating α for each p, q based on the unlabeled data, this turns out to introduce unnecessary additional error into the selection process.

Figure 18 shows how this algorithm can actively adapt to the optimal margin parameters in the non-realizable case. The unlabeled data is again drawn from a margin-conditional Gaussian distribution, this time with a slightly larger margin of 0.1. The examples are labeled according to a target separator as before, but we corrupt the labels by adding random classification noise with noise rate 0.1. In other words, each example disagrees with the target separator independently with probability 0.1. Each point in Figure 18 is an average over 25 independent trials, each with a different test set of 5000 examples.

While it is not clear from Figure 18 whether the active adaptive algorithm consistently outperforms our naïve adapter, we can demonstrate the benefit of label queries

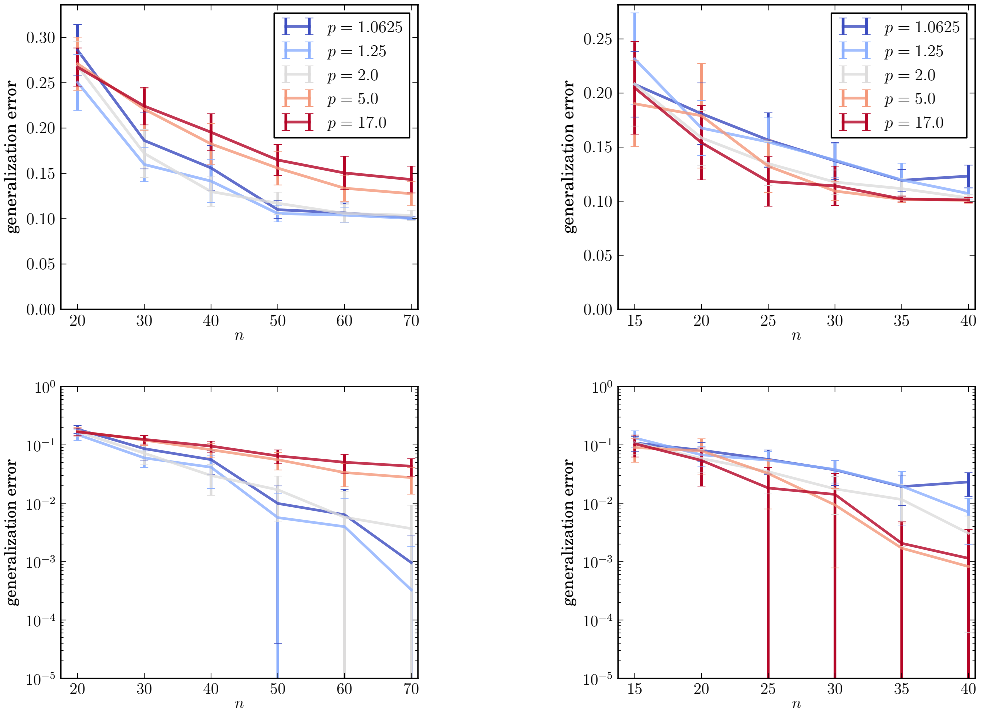


(a) $p = 1$ (b) $p = \infty$

Figure 18: Generalization error versus number of labeled examples used for active L_qL_p SVM, active adaptive, and naïve adaptive algorithms on margin-conditional Gaussian data with random classification noise (caption indicates distribution parameters). Error is shown on both a linear scale (top) and a logarithmic scale (bottom). Error bars represent two standard errors, or approximately a 95% confidence interval.

based on the correct margin parameters through a different experiment. The idea is to separate the margin parameters used for querying from those used for training the final classifier. We compare several active L_qL_p SVMs with different combinations of margin parameters. Each one has a different set of margin parameters used for querying, and the parameter choice is fixed for the duration of the algorithm (they are not adaptive). In all of them, we set the margin parameters used for training the final classifier to match the parameters of the distribution. This allows us to attribute any difference in performance to the query strategy alone.

Figure 19 shows the results of this experiment, where each point is an average over 50 independent trials with separate test set of 5000 examples. When $p = 1$, we see that



(a) $p = 1$

(b) $p = \infty$

Figure 19: Generalization error versus number of labeled examples used for active L_qL_p SVM with fixed margin parameters. The legend gives the margin parameters used for querying while the final classifier is trained using margin parameters matching the distribution. The data is drawn from a margin-conditional Gaussian distribution with random classification noise (caption indicates distribution parameters). Error is shown on both a linear scale (top) and a logarithmic scale (bottom). Error bars represent two standard errors, or approximately a 95% confidence interval.

querying based on L_2L_2 margin is still approximately as effective as using the $L_\infty L_1$ margin, but performance significantly degrades as the type of margin moves further from the correct choice. When $p = \infty$, we see a statistically significant separation between $L_\infty L_1$ margin and $L_1 L_\infty$ margin, with the L_2L_2 somewhere in the middle. These results give very strong evidence that in addition to approximately determining the optimal margin parameters to use, the power of active learning can also be used to shape the distribution of examples in order to more fully take advantage of large L_qL_p margins.

CHAPTER V

ACTIVE LEARNING FOR DOMAIN ADAPTATION

Most machine learning paradigms operate under the assumption that the data generating process remains stable. Training and test data are assumed to be from the same task. However, this is often not an adequate model of reality. For example, it is often desirable to train engineered systems on simulated examples before deployment in the real world, where the instances encountered will inevitably be different. Speech and face recognition systems may be trained on only a small subset of users but intended to work well for everyone. E-commerce recommendation systems that are trained on customers in one country may be used to make predictions and recommendations in a different country. These and numerous other examples signify the importance of developing learning algorithms that adapt to and perform well in changing environments. This is usually referred to as transfer learning or domain adaptation.

In a common model for domain adaptation, the learner receives large amounts of labeled data from a *source* distribution and unlabeled data from the actual *target* distribution (and possibly a small amount of labeled data from the target task as well). The goal of the learner is to output a good model for the target task. Designing methods for this scenario that are statistically consistent with respect to the target task is important, yet challenging. This difficulty occurs even in the so-called *covariate shift* setting, where the change in the environments is restricted to the marginal over the covariates, while the regression functions (the labeling rules) of the involved distributions are identical.

In this chapter, we give the first formal analysis showing that using active learning for domain adaptation yields a way to address these challenges. This is our third example of how the power to make adaptive label queries has benefits beyond reducing labeling effort over passive learning.

In this active domain adaptation model, the learner can make a small number of queries for labels of target examples. Now the goal is to accurately learn a classifier for the target task while making as few label requests as possible. We design and analyze an algorithm showing that being *active adaptive* can yield a consistent learner that uses target labels only where needed.

We propose a simple nonparametric algorithm, ANDA, that combines an active nearest neighbor querying strategy with nearest neighbor prediction. ANDA receives a labeled sample from the source distribution and an unlabeled sample from the target task. It first actively selects a subset of the target data to be labeled based on the amount of source data among the k' nearest neighbors of each target example. Then it outputs a k-nearest neighbor classifier on the combined source and target labeled data.

We prove that ANDA enjoys strong performance guarantees. We first provide a finite sample bound on the expected loss of the resulting classifier in the covariate shift setting. Remarkably, the bound does not depend on source-target relatedness; it only depends on the size of the given unlabeled target sample and properties of the target distribution. This is in stark contrast to most theoretical results for domain adaptation, where additive error terms describing the difference between the source and target frequently appear.

On the other hand, the number of target label queries ANDA makes does depend on the closeness of the involved tasks. ANDA will automatically adjust the number of queries it makes based on local differences between the source and target. We quantify this by giving sample sizes sufficient to guarantee that ANDA makes no queries at all in regions with large enough relative source support. Simply put, ANDA is guaranteed to make enough queries to be consistent but will not make unnecessary ones.

ANDA's intelligent querying behavior and its advantages are further demonstrated by our visualizations and experiments. We visually illustrate ANDA's query strategy and show empirically that ANDA successfully corrects for dataset bias in a challenging image classification task.

The idea of incorporating active learning in to the design of algorithms for domain adaptation has recently received some attention in the machine learning research community [30, 29, 89]. However, to the best of our knowledge, there has not been any formal analysis of using active learning to adapt to distribution changes. We believe that active learning is a particularly promising tool for obtaining domain adaptive learners and that this work provides an important piece of the theoretical foundation this area deserves.

5.1 Related Work

For domain adaptation, even under covariate shift, performance guarantees usually involve an extra additive term that measures the difference between source and target tasks (that is the loss does not converge to the target optimal opt_T but to $opt_T + \Delta$, where Δ is some measure of distance between distributions) [19, 75], or they rely on strong assumptions, such as the target support being a subset of the source support and the density ratio between source and target being bounded from below [99, 20]. Generally, the case where the target is partly supported in regions that are not covered by the source, is considered to be particularly challenging [32]. There are heuristics, that aim to find a suitable mapping of source and target into some common space [95], but the success of any such method again relies on very strong prior knowledge about source and target relatedness. We show that our method guarantees small loss independently of source target relatedness.

Nearest neighbor methods have been studied for decades [34, 97, 72]. Due to their flexibility, nearest neighbor methods can suffer in high dimensions, both computationally and statistically. However, recently, there has been renewed interest in these methods and ways to overcome the curse of dimensionality. It has been proven that the generalization performance actually scales with notions of intrinsic dimension, which can be lower than the dimension of the feature space [71]. Several recent studies have shown how to perform nearest neighbor search more efficiently [42, 83, 82] Selective sampling for nearest neighbor classification has been shown to be consistent under certain conditions on the querying rule [38]; however, this work considers a data stream that comes from a fixed distribution (as opposed to our covariate shift setting). A 1-Nearest Neighbor algorithm has been analyzed under covariate shift [20]; however, that study assumes a fixed lower bound on a weight ratio between source and target, and therefore does not apply to settings where the target is supported in areas where the source is not. In this work, we argue that the flexibility of nearest neighbor methods can be exploited for adapting to changing environments; particularly so for choosing where to query for labels by detecting areas of the target task that are not well covered by the source.

5.2 Preliminaries

Let $(\mathcal{X}, \rho)$ be a separable metric space. We let $B_r(x)$ denote the closed ball of radius r around x. We let $\mathbb{N}_\epsilon(\mathcal{X}, \rho)$ denote the *ϵ-cover-number* of the metric space, that is, the minimum number of subsets $C \subseteq \mathcal{X}$ of diameter at most ϵ that cover the space $\mathcal{X}$ (a set $C \subseteq \mathcal{X}$ has diameter at most ϵ if, for all $x, x' \in C$, we have $\rho(x, x') \leq \epsilon$).

We consider a binary classification task, where P_S and P_T denote *source* and *target distributions* over $\mathcal{X} \times \{0, 1\}$. We let D_S and D_T denote the source and target marginal distributions over $\mathcal{X}$, respectively. Further, we let $\mathcal{X}_S$ and $\mathcal{X}_T$ denote the *support* of D_S and D_T respectively. That is, for $I \in \{S, T\}$, we have

$$\mathcal{X}_I := \{x \in \mathcal{X} \ : \ \forall r > 0, \ D_I(B_r(x)) > 0\}.$$

We use the notation S and T for i.i.d. samples from P_S and D_T, respectively, and let $|S| = m_S$, $|T| = m_T$, and $m = m_S + m_T$. We let $\hat{S}, \hat{T}$ denote the empirical distributions according to S and T.

Our analysis is in the *covariate shift* setting, in which the regression function $\eta(x) = \mathbb{P}[y = 1|x]$ is the same for both source and target. In other words, the only difference between the distributions P_S and P_T is the difference between the marginal distributions D_S and D_T.

For any finite $A \subseteq \mathcal{X}$ and $x \in \mathcal{X}$, the notation $x_1(x, A), \ldots, x_{|A|}(x, A)$ gives an ordering of the elements of A such that $\rho(x_1(x, A), x) \leq \rho(x_2(x, A), x) \leq \cdots \leq \rho(x_{|A|}(x, A), x)$. If A is a labeled sequence of domain points, $A = ((x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m))$, then we use the same notation for the labels (that is $y_i(x, A)$ denotes the label of the i-th nearest point to x in A). We use the notation $k(x, A) = \{x_1(x, A), \ldots, x_k(x, A)\}$ to denote the set of the k nearest neighbors of x in A.

We are interested in bounding the target loss of a k-nearest neighbor classifier. For a sequence A of labeled points $A = ((x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m))$ we let h^k_A denote the *k-NN classifier* on A:

$$h^k_A(x) := \mathbb{1}\left[\frac{1}{k}\Sigma^k_{i=1} y_i(x, A) \geq \frac{1}{2}\right],$$

where $\mathbb{1}[\cdot]$ denotes the indicator function.

We denote the *Bayes classifier* by $h^*(x) = \mathbb{1}[\eta(x) \geq 1/2]$ and the *target loss* of a classifier $h : \mathcal{X} \to \{0, 1\}$ by $\mathcal{L}_T(h) = \mathbb{P}_{(x,y)\sim P_T}[y \neq h(x)]$. For a subset $A \subseteq \mathcal{X}$ of the domain that is measurable both with respect to D_S and D_T and satisfies $D_T(A) > 0$, we define the *weight ratio* of A as $\beta(A) := D_S(A)/D_T(A)$. For a collection of subsets $\mathcal{B} \subseteq 2^{\mathcal{X}}$ (for example all balls in $(\mathcal{X}, \rho)$), we let $d_{\mathsf{VC}}(\mathcal{B})$ denote its VC-dimension. For example, the VC-dimension of the class of all balls in $\mathbb{R}^d$ is $d + 1$.

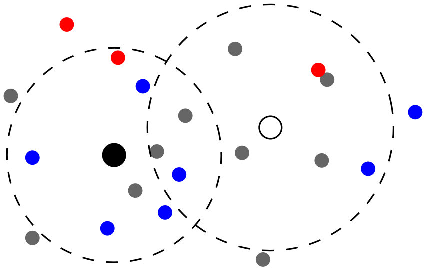


Figure 20: An illustration of (k, k')-NN-cover with $k = 4$ and $k' = 9$. Unlabeled examples are gray and labeled examples are blue and red. The large filled circle (left) is covered by labeled examples while the large open circle (right) is not.

5.3 Active Nearest Neighbors Algorithm

In brief, our algorithm receives a labeled sample S (from the source distribution), an unlabeled sample T (from the target distribution), and two parameters k and k'. It then chooses a subset $T^l \subset T$ to be labeled, queries the labels of points in T^l, and outputs a k-NN predictor on $S \cup T^l$ (see Algorithm 7). The subset T^l is chosen so that the resulting labeled set $S \cup T^l$ is a (k, k')-NN-*cover* for the target (unlabeled) sample T.

Definition ((k, k')-NN-cover)**.** Let $T \subseteq \mathcal{X}$ be a set of elements in a metric space $(\mathcal{X}, \rho)$ and let $k, k' \in \mathbb{N}$ with $k \leq k'$. A set $R \subseteq \mathcal{X}$ is a (k, k')-NN-cover for T if, for every $x \in T$, either $x \in R$ or there are k elements from R among the k' nearest neighbors of x in $T \cup R$, that is $|k'(x, T \cup R) \cap R| \geq k$ (or both).

Figure 20 illustrates the concept of (k, k')-NN-covers. Our loss bound in Section 5.4 (Theorem 10) holds whenever $T^l \cup S$ is some (k, k')-NN-cover of T. Algorithm 8 provides a simple strategy to find such a cover: add to T^l all points whose k' nearest neighbors among $S \cup T$ include fewer than k source examples. It is easy to see that this will always result in a (k, k')-NN-cover of T. Furthermore, this approach has a query safety property: the set T^l produced by Algorithm 8 satisfies $T^l \cap Q = \emptyset$ where $Q = \{x \in T : |k'(x, S \cup T) \cap S| \geq k\}$ is the set of target examples that have k source

Algorithm 7 ANDA: Active Nearest Neighbors for Domain Adaptation

```
    input Labeled set $S$, unlabeled set $T$, parameters $k$, $k'$
    Find $T^l \subseteq T$ s.t. $S \cup T^l$ is a $(k, k')$-NN-cover of $T$
    Query the labels of points in $T^l$
    return $h^k_{S \cup T^l}$, the $k$-NN classifier on $S \cup T^l$
```

Algorithm 8 Safe: Finding a (k, k')-NN-cover

```
    input Labeled set $S$, unlabeled set $T$, parameters $k$, $k'$
    return $\{x \in T : |k'(x, S \cup T) \cap S| < k\}$
```

neighbors among their k' nearest neighbors in $S \cup T$. In other words, Algorithm 8 will not query the label of any target example in regions with sufficiently many labeled source examples nearby, a property used in the query bound of Theorem 11.

5.3.1 Finding a Small (k, k')-NN-cover

In order to make as few label queries as possible, we would like to find the smallest subset T^l of T to be labeled such that $T^l \cup S$ is a (k, k')-NN-cover of T. As we show below, this problem is NP-hard and is a special case of Minimum Multiset Multicover, a generalization of the well-known NP-hard Minimum Set Cover problem (see [81] and Chapter 13.2 in [106]).

Definition (Minimum Multiset Multicover)**.** Given a universe U of n elements, a collection of multisets $\mathcal{S}$, and a coverage requirement r_e for each element $e \in U$, we say that a multiset $S \in \mathcal{S}$ covers element e once for each copy of e appearing in S. The goal is to find the minimum cardinality set $\mathcal{C} \subseteq \mathcal{S}$ such that every element $e \in U$ is covered at least r_e times by the multisets in $\mathcal{C}$.

We show that finding a minimum (k, k')-NN-cover is NP-hard via reduction from Minimum Dominating Set in 3-regular graphs (Min-Dom-3Reg). Given a graph $G = (V, E)$, a set $D \subseteq V$ is a *dominating set* of G if for every $v \in V$, either $v \in D$ or v is adjacent to some vertex in D. The optimization problem Min-Dom-3Reg is that of finding, given a 3-regular graph G, a dominating set of G of minimum cardinality.

Min-Dom-3Reg is known to be NP-hard [1].

Theorem 9. *Given a metric space (X, ρ), a set $T \subseteq X$ of targets, and integers $k \leq k'$, the optimization problem of finding a minimum cardinality set R such that R is a (k, k')-*NN*-cover of T is NP-hard.*

Proof. Given an instance of Min-Dom-3Reg consisting of a 3-regular graph G, we construct an instance of minimum (k, k')-NN-cover as follows. Let $X = T = V$ and ρ be the shortest path metric on G. Let $k' = 3$ and $k = 1$. Notice that for any element $x \in T$, its $k' = 3$ nearest neighbors are precisely its adjacent neighbors in G because G is 3-regular. Since we require only $k = 1$ of them to be in D in order for x to be covered, the notions of cover for dominating set and for (k, k')-NN-cover are equivalent. Therefore, a set D is a dominating set of G if and only if D is a (k, k')-NN-cover of T, and from this it follows that the minimization problems are also equivalent. □

We can phrase the problem of finding the smallest T^l such that $T^l \cup S$ is a (k, k')-NN-cover of T as a Minimum Multiset Multicover problem as follows. Let $U = T$ and set the coverage requirements as $r_x = \max(0, k - |k'(x, S \cup T) \cap S|)$ for each $x \in T$. The collection $\mathcal{S}$ contains a multiset S_x for each $x \in T$, where S_x contains k copies of x and one copy of each element in $\{x' \in T : x \in k'(x', S \cup T)\}$. By construction, a minimum multiset multicover of this instance is also a minimum (k, k')-NN-cover and vice versa.

While (k, k')-NN-cover is NP-hard to solve exactly, by phrasing it as a special case of Minimum Multiset Multicover we know that a greedy algorithm efficiently provides an approximate solution (see the end of this section). The greedy algorithm iteratively picks the "most helpful" multiset until every element e is covered at least r_e times, where "most helpful" means the multiset that provides the most total coverings of elements up to, but not above, their coverage requirements.

Algorithm 9 EMMA: Efficient multiset multicover approximation for finding a small (k, k')-NN-cover

```
input Labeled set $S$, unlabeled set $T$, parameters $k$, $k'$
$T^l \leftarrow \emptyset$
for all $x \in T$ do
  $r_x \leftarrow \max(0, k - k'(x, S \cup T) \cap S)$
  $n_x \leftarrow |\{x' \in T : r_{x'} > 0 \wedge x \in k'(x', S \cup T)\}|$
while $\{x \in T : r_x > 0\} \neq \emptyset$ do
  $T^l \leftarrow T^l \cup \{\text{argmax}_{x \in T \setminus T^l} r_x + n_x\}$
  for all $x \in T$ do
    $r_x \leftarrow \max(0, k - k'(x, S \cup T) \cap (S \cup T^l))$
    $n_x \leftarrow |\{x' \in T \setminus T^l : r_{x'} > 0 \wedge x \in k'(x', S \cup T)\}|$
return $T^l$
```

Algorithm 9 formalizes this as an ANDA subroutine called EMMA for finding a small (k, k')-NN-cover. In the language of (k, k')-NN-covers, in each round EMMA computes the helpfulness of each $x \in T$ in two parts. The remaining coverage requirement r_x is the number of times x would cover itself if added to T^l (that is, the savings from not having to use r_x additional neighbors of x), and the total neighbor coverage n_x is the number of times x would cover its neighbors if added to T^l. EMMA then selects the point x with the largest sum $r_x + n_x$ among all points in T that have not yet been added to T^l.

In its most basic form, EMMA does not have the same query safety property enjoyed by Safe because the greedy strategy may elect to query labels of target examples that were already fully covered by source examples. We can ensure that an intelligent query strategy like EMMA still has the desired query safety property by first running Safe and then passing the resulting set T_{safe} to EMMA as its unlabeled sample. We call the resulting strategy for finding a (k, k')-NN-cover Safe-EMMA.

Computational considerations. Apart from the computational complexity of performing the k- and k'-nearest neighbor search steps[1], computational considerations

[1] Since our primary concern is that of analyzing the statistical properties, we assume access to a method for performing nearest neighbor search. We treat the issue of how to efficiently perform

also arise when determining whether to use Safe or Safe-EMMA. The computational complexity of Safe is relatively small. For each target example, it only requires performing a k'-NN search and counting the labels among the resulting neighbor set, so the runtime is $O(m_T(k' + N_{k'}))$, where we use $N_{k'}$ to denote the runtime of a k'-NN search on $S \cup T$.

On the other hand, Safe-EMMA requires first performing k'-NN searches and initializing coverage counts for each target example and then maximizing the coverage improvement and updating coverage counts for each query made. Assuming the appropriate data structures are used, the first part takes $O(N_{k'})$ time for k'-NN search and $O(k')$ time for initializing coverage counts for each target example. The main loop requires an $O(m_T)$-time maximization step and $O(1)$-time coverage count updates per target example, so the overall complexity is at most $O(m_T(k' + N_{k'}) + m_T^2) = O(m_T(k' + N_{k'} + m_T))$.

Asymptotically, Safe is always at least as fast as Safe-EMMA. If $N_{k'} = \Omega(m_T)$ (for example, if a linear search method is used or if m_S is much larger than m_T) then the two methods will have the same computational complexity. Otherwise, Safe will be slightly better asymptotically (this will occur, for example, if a space partitioning search is used for data in $\mathbb{R}^d$ with $m_S = O(m_T)$ and $d \ll m_T$). While the two methods may behave the same asymptotically, constant factor slowdowns for Safe-EMMA may be a significant issue in practice.

Approximation guarantees. Minimum Multiset Multicover is known to remain NP-hard even when the multisets in $\mathcal{S}$ are small. However, a small upper bound b on the maximum size of any multiset in $\mathcal{S}$ can make the problem much easier to approximate. Specifically, the greedy algorithm has an approximation factor of H_b, the b-th harmonic number [81]. This is known to be essentially optimal under

nearest neighbor search as orthogonal to this primary concern. For some results in this area, see [45, 39, 107, 82, 42] and references therein.

standard hardness assumptions.

In our setting, the size of the largest multiset is determined by the point $x \in T$ with the largest number of points in $S \cup T$ having x as one of their k' nearest neighbors. In general metric spaces this can be up to $m = m_S + m_T$, resulting in a multiset of size $m + k$ and an approximation factor of $H_{m+k} = O(\log m)$. However, in spaces with doubling-dimension γ, it is known that $b \leq k' 4^\gamma \log_{3/2}(2L/S)$ where L and S are respectively the longest and shortest distances between any two points in T [111].

5.4 Performance Guarantees

In this section, we analyze the expected loss of the output classifier of ANDA as well as its querying behavior. The bound in Section 5.4.1 on the loss holds for ANDA with any of the sub-procedures presented in Section 5.3. To simplify the presentation we use ANDA as a placeholder for any of ANDA-Safe, ANDA-EMMA and ANDA-Safe-EMMA. The bounds on the number of queries in Section 5.4.3 hold for ANDA-Safe and ANDA-Safe-EMMA, which we group under the placeholder ANDA-S.

5.4.1 Bounding the Loss

We start with a finite sample bound under the assumption that the regression function η satisfies a λ-Lipschitz condition. That is, we have $|\eta(x) - \eta(x')| \leq \lambda \rho(x, x')$ for all $x, x' \in \mathcal{X}_S \cup \mathcal{X}_T$.

Our bound on the expected loss in Theorem 10 is proven using standard techniques for nearest neighbor analysis. However, since our algorithm does not predict with a fully labeled sample from the target distribution (possibly very few or even none of the target generated examples get actually labeled and the prediction is mainly based on source generated examples), we need to ensure that the set of labeled examples still sufficiently covers the target task. The following lemma serves this purpose. It bounds the distance of an arbitrary domain point x to its k-th nearest *labeled point* in terms of its distance to its k'-th nearest *target sample point*. Note that the bound in the lemma is easy to see for points in T. However, we need it for arbitrary (test-) points in the domain.

Lemma 4. *Let T be a finite set of points in a metric space $(\mathcal{X}, \rho)$ and let R be a (k, k')-*NN-*cover for T. Then, for all $x \in \mathcal{X}$ we have*

$$\rho(x, x_k(x, R)) \leq 3\rho(x, x_{k'}(x, T))$$

Proof. Let $x \in \mathcal{X}$. If the set $k'(x, T)$ of the k' nearest neighbors of x in T contains k points from R, we are done (in this case we actually have $\rho(x, x_k(x, R)) \leq \rho(x, x_{k'}(x, T))$). Otherwise, let $x' \in k'(x, T) \setminus R$ be one of these points that is not in R. Since R is a (k, k')-NN-cover for T, and $x' \in T$, the set of the k' nearest neighbors of x' in $R \cup T$ contains k elements from R.

Let x'' be any of these k elements, that is $x'' \in R \cap k'(x', R \cup T)$. Note that $\rho(x', x'') \leq 2\rho(x, x_{k'}(x, T))$ since x' is among the k' nearest neighbors of x and x'' is among the k' nearest neighbors of x' in $R \cup T$. Thus, we have

$$\begin{aligned}\rho(x, x'') &\leq \rho(x, x') + \rho(x', x'') \\ &\leq \rho(x, x_{k'}(x, T)) + 2\rho(x, x_{k'}(x, T)) \\ &= 3\rho(x, x_{k'}(x, T)).\end{aligned}$$

□

This lemma allows us to establish the finite sample guarantee on the expected loss of the classifier output by ANDA. Note that the guarantee in the theorem below is independent of the size and the generating process of S (except for the labels being generated according to η), while possibly (if S covers the target sufficiently) only few target points are queried for labels. Recall that $\mathbb{N}_\epsilon(\mathcal{X}_T, \rho)$ denotes the ϵ-covering number of the target support.

Theorem 10. *Let $(\mathcal{X}, \rho)$ be a metric space and let P_T be a (target) distribution over $\mathcal{X} \times \{0,1\}$ with λ-Lipschitz regression function η. Then for all $k' \geq k \geq 10$, all $\epsilon > 0$, and any unlabeled sample size m_T and labeled sequence $S = ((x_1, y_1), \ldots, (x_{m_S}, y_{m_S}))$ with labels y_i generated by η,*

$$\mathop{\mathbb{E}}_{T \sim P_T^{m_T}} [\mathcal{L}_T(\mathrm{ANDA}(S, T, k, k'))] \;\leq\; \left(1 + \sqrt{\frac{8}{k}}\right) \mathcal{L}_T(h^*) + 9\lambda\epsilon + \frac{2\, \mathbb{N}_\epsilon(\mathcal{X}_T, \rho)\, k'}{m_T}.$$

The proof (see Appendix A.1) incorporates our bound on the distance to the k nearest labeled points of Lemma 4 into a standard technique for nearest neighbor analysis (as in [93]). The key to the guarantee being the bound in Lemma 4, one could obtain analogous generalization bounds under relaxed assumptions for which nearest neighbor classification can be shown to succeed (see, e.g. [31] for a discussion on such). Similarly, one could obtain bounds for other settings, such as multi-class classification and regression.

Generalized covariate shift. While the covariate shift assumption seems rather restrictive (source and target regression functions need to take identical values at every domain point), it is not hard to see that our guarantee in Theorem 10 holds under more general conditions. Intuitively, ANDA only requires the source to be a good estimate of the target's Bayes predictor. That is, the source regression function could be a "less noisy" version of the target regression function, and does not need to be Lipschitz continuous (or even continuous at all) itself. Below, we formally state the relaxed conditions under which our guarantees hold.

Definition (Generalized covariate shift)**.** For two values $a, b \in [0,1]$ we say that $a >_{noise} b$ if

$$a >_{noise} b \;\Leftrightarrow\; \begin{cases} a \geq b \;\wedge\; \min\{b, 1-b\} \leq 1/2 \\ \text{or} \\ a < b \;\wedge\; \min\{b, 1-b\} > 1/2 \end{cases}$$

Now we say that source and target regression function satisfy the *generalized covariate shift with target Lipschitz constant* λ, if the target regression function η_T is λ-Lipschitz and for all $x, x' \in \mathcal{X}$ we have

$$\eta_S(x') >_{noise} \eta_T(x) \;\Rightarrow\; |\eta_S(x') - \eta_T(x)| \leq \lambda|x - x'|$$

5.4.2 Consistency

We show that ANDA is consistent in a slightly more general setting, namely if the regression function is *uniformly continuous* and the $\mathbb{N}_\epsilon(\mathcal{X}_T, \rho)$ are finite. Note that this is the case, for example, if $(\mathcal{X}, \rho)$ is compact and η is continuous. Recall that a function $\eta : \mathcal{X} \to \mathbb{R}$ is *uniformly continuous* if for every $\gamma > 0$ there exists a δ such that for all $x, x' \in \mathcal{X}$, $\rho(x, x') \leq \delta \Rightarrow |\eta(x) - \eta(x')| \leq \gamma$.

Corollary 1. *Let $(\mathcal{X}, \rho)$ be a metric space, and let $\mathcal{P}(\mathcal{X}, \rho)$ denote the class of distributions over $\mathcal{X} \times \{0, 1\}$ with uniformly continuous regression functions. Let $(k_i)_{i\in\mathbb{N}}$, $(k'_i)_{i\in\mathbb{N}}$ and $(m_i)_{i\in\mathbb{N}}$ be non-decreasing sequences of natural numbers with $k'_i \geq k_i$ for all i, and $k_i \to \infty, k'_i \to \infty, m_i \to \infty$ and $(k'_i/m_i) \to 0$ as $i \to \infty$. For each $i \in \mathbb{N}$, let $S_i \in (\mathcal{X} \times \{0, 1\})^{n_i}$ be a sequence of labeled domain points. Then for any distribution $P_T \in \mathcal{P}(\mathcal{X}, \rho)$ with finite covering numbers $\mathbb{N}_\epsilon(\mathcal{X}_T, \rho)$, we have*

$$\lim_{i\to\infty} \mathop{\mathbb{E}}_{T\sim P_T^{m_i}} [\mathcal{L}_T(\mathrm{ANDA}(S_i, T, k_i, k'_i))] \;=\; \mathcal{L}_T(h^*).$$

Proof. We need to show that for every $\alpha > 0$, there exists an index i_0, such that

$$\mathop{\mathbb{E}}_{T\sim P_T^{m_i}} [\mathcal{L}_T(\mathrm{ANDA}(S_i, T, k_i, k'_i))] \;=\; \mathcal{L}_T(h^*) + \alpha$$

for all $i \geq i_0$. Let $P_T \in \mathcal{P}(\mathcal{X}, \rho)$ and α be given.

Let γ be so that $9\gamma \leq \alpha/3$. Since η is uniformly continuous, there is a δ, such that for all $x, x' \in \mathcal{X}$,

$$\rho(x, x') \leq \delta \Rightarrow |\eta(x) - \eta(x')| \leq \gamma.$$

Note that the only way we used the λ-Lipschitzness in the proof of Theorem 10 is by using that for any two points x, x' that lie in a common element C of an ϵ-cover of the space, we have $|\eta(x) - \eta(x')| \leq \lambda\epsilon$. Thus, we could now repeat the proof of Theorem 10, using a δ-cover of the space and obtain that

$$\mathop{\mathbb{E}}_{T\sim D_T^{m_T}}[\mathcal{L}_T(\mathrm{ANDA}(S,T,k,k'))] \;\leq\; (1+\sqrt{\frac{8}{k}})\mathcal{L}_T(h^*) + 9\gamma + \frac{2\,\mathbb{N}_\delta(\mathcal{X}_T,\rho)\,k'}{m_T}.$$

for all $k \geq 10$ and $k' \geq k$. Now let i_1 be so that $\sqrt{\frac{8}{k_i}} \leq \frac{\alpha}{3}$ for all $i \geq i_1$. Note that this implies

$$\sqrt{\frac{8}{k_i}}\mathcal{L}_T(h^*) \leq \frac{\alpha}{3}$$

for all $i \geq i_1$. Since $(k'_i/m_i) \to 0$ as $i \to \infty$, we can choose i_2 be so that

$$\frac{2\,\mathbb{N}_\delta(\mathcal{X}_T,\rho)\,k'_i}{m_i} \leq \frac{\alpha}{3}$$

for all $i \geq i_2$. Together these imply that for all $i \geq i_0 := \max\{i_1, i_2\}$, we have

$$\mathop{\mathbb{E}}_{T\sim P_T^{m_i}}[\mathcal{L}_T(\mathrm{ANDA}(S_i,T,k_i,k'_i))] \;=\; \mathcal{L}_T(h^*) + \alpha$$

as desired. □

5.4.3 Bounding the Number of Queries

In this section, we show that our algorithm automatically adapts the number of label queries to the similarity of source and target task. First, we now provide a finite sample bound that implies that with a sufficiently large source sample, with high probability, ANDA-S does not query at all in areas where the weight ratio of balls is bounded from below; i.e. it only queries where it is "needed." In our analysis, we employ a lemma by [71], which follows from VC-theory [105].

Lemma 5 (Lemma 1 in [71]). *Let $\mathcal{B}$ denote the class of balls in $(\mathcal{X}, \rho)$, and let D be a distribution over $\mathcal{X}$. Let $0 < \delta < 1$, and define $\alpha_n = (d_{\mathsf{VC}}(\mathcal{B})\ln(2n) + \ln(6/\delta))/n$. The following holds with probability at least $1 - \delta$ (over a sample T of size n drawn* *i.i.d. from D) for all balls $B \in \mathcal{B}$: if $a \geq \alpha_n$, then $\hat{T}(B) \geq 3a$ implies $D(B) \geq a$ and $D(B) \geq 3a$ implies $\hat{T}(B) \geq a$.*

With this, we now prove our query bound. We let $B_{k,T}(x)$ denote the smallest ball around x that contains the k nearest neighbors of x in T, and $\mathcal{B}$ the class of all balls in $(\mathcal{X}, \rho)$. Recall that $\beta(B) = D_S(B)/D_T(B)$ is the weight ratio.

Theorem 11. *Let $\delta > 0$, $w > 0$ and $C > 1$. Let m_T be some target sample size with $m_T > k' = (C+1)k$ for some k that satisfies $k \geq 9\left(d_{\mathsf{VC}}(\mathcal{B})\ln(2m_T) + \ln(6/\delta)\right)$. Let the source sample size satisfy*

$$m_S \geq \frac{72\,\ln(6/\delta) m_T}{C\,w} \ln\left(\frac{9\,m_T}{C\,w}\right)$$

Then, with probability at least $1 - 2\delta$ over samples S of size m_S (i.i.d. from P_S) and T of size m_T (i.i.d. from D_T), ANDA-S on input S, T, k, k' will not query any points $x \in T$ with $\beta(B_{Ck,T}(x)) > w$.

Proof. Since $k \geq 9\left(d_{\mathsf{VC}}(\mathcal{B})\ln(2m_T) + \ln(6/\delta)\right)$, we have $d_{\mathsf{VC}}(\mathcal{B})/k < 1$. Thus, we have

$$m_S \geq \max\left\{8\left(\frac{9\,d_{\mathsf{VC}}(\mathcal{B})\,m_T}{C\,k\,w}\right)\ln\left(\frac{9\,d_{\mathsf{VC}}(\mathcal{B})\,m_T}{C\,k\,w}\right),\ \frac{18\,\ln(6/\delta)\,m_T}{C\,k\,w},\ \frac{9\,m_T}{C\,w}\right\},$$

Note that

$$m_S \geq 8\left(\frac{9\,d_{\mathsf{VC}}(\mathcal{B})\,m_T}{C\,k\,w}\right)\ln\left(\frac{9\,d_{\mathsf{VC}}(\mathcal{B})\,m_T}{C\,k\,w}\right)$$

implies that

$$m_S \geq 2\left(\frac{9\,d_{\mathsf{VC}}(\mathcal{B})\,m_T}{C\,k\,w}\right)\ln(2m_S),$$

and together with the second lower bound (in the max) on m_S, this yields

$$m_S\frac{C\,k\,w}{3\,m_T} \ \geq\ 3(d_{\mathsf{VC}}(\mathcal{B})\ln(2m_S) + \ln(6/\delta)). \tag{7}$$

We now assume that S and T are so that the implications in Lemma 5 are valid (this holds with probability at least $1 - 2\delta$ over the samples S and T). Let $x \in T$ be such that $\beta(B_{Ck,T}(x)) > w$. By definition of the ball $B_{Ck,T}(x)$, we have $\hat{T}(B_{Ck,T}(x)) = \frac{Ck}{m_T}$, and by our choice of k, we have

$$\hat{T}(B_{Ck,T}(x)) \;=\; \frac{C\,k}{m_T} \;\geq\; \frac{C\,9\,(d_{\mathsf{VC}}(\mathcal{B})\ln 2m_T + \ln 6/\delta)}{m_T}.$$

Now Lemma 5 implies that $D_T(B_{Ck,T}(x)) \;\geq\; \frac{C\,k}{3\,m_T}$, so the condition on the weight ratio of this ball now yields

$$D_S(B_{Ck,T}(x)) \;\geq\; \frac{C\,k\,w}{3\,m_T} \;=\; m_S\frac{C\,k\,w}{3\,m_T\,m_S} \;\geq\; 3\left(\frac{d_{\mathsf{VC}}(\mathcal{B})\ln(2m_S) + \ln(6/\delta)}{m_S}\right),$$

where the last inequality follows from Equation (7). Now, Lemma 5, together with $m_S \geq \frac{9\,m_T}{C\,w}$ (the third term in the max), implies that

$$\hat{S}(B_{Ck,T}(x)) \;\geq\; \frac{C\,k\,w}{9\,m_T} \;\geq\; \frac{k}{m_S}.$$

This means that $B_{Ck,T}(x)$ contains k examples from the source, which implies that among the $k' = Ck + k$ nearest sample points (in $S \cup T$) there are k source examples, and therefore x will not be queried by ANDA-S. □

Theorem 11 provides a desirable guarantee for the “lucky” case: It implies that if the source and target distributions happen to be identical or very similar, then, given that ANDA-S is provided with a sufficiently large source sample, it will not make any label queries at all. More importantly, the theorem shows that, independent of an *overall* source/target relatedness measure, the querying of ANDA-S adapts automatically to a *local* relatedness measure in the form of weight ratios of balls around target sample points. ANDA-S queries only where it is necessary to compensate for insufficient source coverage.

5.4.4 Query Consistency

Extending the proof technique of Theorem 11, we get a “query-consistency” result under the assumption that D_S and D_T have continuous density functions. In the limit of large source samples, ANDA-S will, with high probability, not make any queries in the source support.

Theorem 12. *Let D_S and D_T have continuous density functions. Let $\delta > 0$, $C > 1$, and let m_T, k and k' satisfy the conditions of Theorem 11. Then, there exists a (sufficiently large) source sample size M_S such that with probability at least $(1 - 3\delta)$ over source samples of size $m_S \geq M_S$ and target samples of size m_T, ANDA-S will not make any label queries in the source support.*

Proof. Recall that, according to the requirements of Theoren 11, we have $m_T > k' = (C+1)k$ for some k that satisfies

$$k \geq 9\left(d_{\mathsf{VC}}(\mathcal{B})\ln(2m_T) + \ln(6/\delta)\right).$$

Since D_T has a continuous density function, for every point x in $\mathcal{X}_T$ and $0 < \epsilon \leq 1$, there is a ball $B^\epsilon(x)$ of target weight exactly ϵ around x (i.e. $D_T(B^\epsilon(x)) = \epsilon$). For some $w > 0$, let $\mathcal{X}_T(\epsilon, w) \subseteq \mathcal{X}_T$ denote the set of points x whose ϵ-ball has weight ratio smaller than w, that is

$$\mathcal{X}_T(\epsilon, w) = \{x \in \mathcal{X}_T \mid \beta(B^\epsilon(x)) < w\}.$$

Claim 1.

$$\lim_{w \to 0} D_T(\mathcal{X}_T(\epsilon, w) \cap \mathcal{X}_S) = 0$$

Let $\epsilon = Ck/3m_T$. Given the claim (which we prove below), we can choose w small enough such that (with probability at least $1 - \delta$), a target sample of size m_T will not hit $\mathcal{X}_T(\epsilon, w) \cap \mathcal{X}_S$. Now we can choose a size M_S for the source sample S large enough such that (with probability $1 - 2\delta$) ANDA-S will not query any points in $\mathcal{X}_S \setminus \mathcal{X}_T(\epsilon, w)$. This is shown similarly to the proof of Theorem 11 as follows.

First, assume that the sample T is so that the implications of Lemma 5 are satisfied (this also happens with probability at least $(1 - \delta)$). Then, by invoking the contrapositive of the first implication in Lemma 5,

$$D_T(B^\epsilon(x)) = \epsilon = \frac{Ck}{3m_T}$$

and

$$\frac{Ck}{m_T} \geq \frac{C\,9\,(d_{\mathsf{VC}}(\mathcal{B})\ln(2m_T) + \ln(6/\delta))}{m_T}$$

implies that

$$\widehat{T}(B^\epsilon(x)) \leq \frac{Ck}{m_T}.$$

Thus, for all x, the ball $B^\epsilon(x)$ contains at most Ck points from the target sample T.

Now we choose a sufficiently large size for the source sample S, namely

$$m_S \geq M_S = \frac{72\ \ln(6/\delta)m_T}{C\,w} \ln\left(\frac{9\,m_T}{C\,w}\right)$$

for the value of w chosen above. We assume that the sample S is so that the implications of Lemma 5 are satisfied (this, again, holds with probability at least $(1-\delta)$).

Exactly as in the proof of Theorem 11, we can show that, for all x with $\beta(B^\epsilon(x)) \geq w$,

$$D_T(B^\epsilon(x)) = \frac{Ck}{3m_T}$$

implies

$$\widehat{S}(B^\epsilon(x)) \geq \frac{k}{m_S},$$

Thus, for all x with $\beta(B^\epsilon(x)) \geq w$, the ball $B^\epsilon(x)$ contains at least k points from the source sample S.

In summary, we have shown that with probability $(1-3\delta)$ over the samples S and T, for all target sample points x, that fall into the source support, we have $\beta(B^\epsilon(x)) \geq w$, and for those the ball $B^\epsilon(x)$ contains at most Ck target and at least k source samples points. This implies that for all target sample points, that fall into the source support, the $k' = (C+1)k$ Nearest Neighbor ball (in $S \cup T$) around x contains at least k points from the source sample and will therefore not be queried.

Proof of Claim 1. Let $(w_i)_{i\in\mathbb{N}}$ be a decreasing sequence that converges to 0. Then the sets $\mathcal{X}_T(\epsilon, w_i)$ are linearly ordered by inclusion (getting smaller as w_i gets smaller).

Thus, the limit of the sequence of sets $\mathcal{X}_T(\epsilon, w_i)$ exists and we have

$$\lim_{i\to\infty} \mathcal{X}_T(\epsilon, w_i) = \bigcap_{i=1}^{\infty} \mathcal{X}_T(\epsilon, w_i) \subseteq \mathcal{X}_T \setminus \mathcal{X}_S$$

To see the last inclusion, recall that, by definition, a point x is in the source support $\mathcal{X}_S$ if and only if every ball B around x has positive source mass $D_S(B) > 0$. Hence, in particular $D_S(B^\epsilon(x)) > 0$, which implies that these balls also have strictly positive weight ratio $\beta(B^\epsilon(x)) > 0$. Thus, for every point x in the source support, there exists an i such that $x \notin \mathcal{X}_T(\epsilon, w_i)$, since the w_i converge to 0.

The above set convergence implies

$$\lim_{i\to\infty} D_T(\mathcal{X}_T(\epsilon, w_i)) = D_T(\bigcap_{i=1}^{\infty} \mathcal{X}_T(\epsilon, w_i)) \leq D_T(\mathcal{X}_T \setminus \mathcal{X}_S).$$

This, in turn, implies

$$\lim_{i\to\infty} D_T(\mathcal{X}_T(\epsilon, w_i) \cap \mathcal{X}_S) \leq D_T((\mathcal{X}_T \setminus \mathcal{X}_S) \cap \mathcal{X}_S) = 0,$$

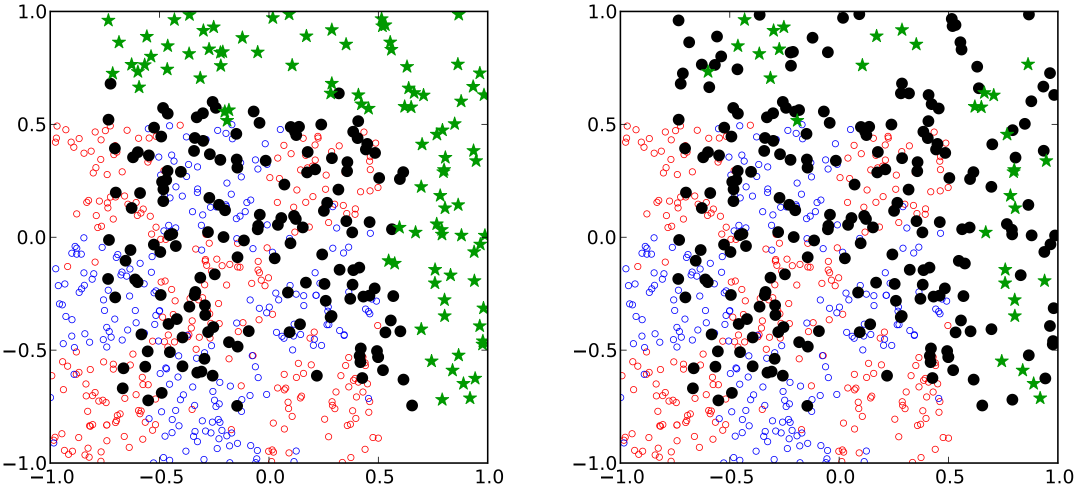
yielding the claim. □

□

Together with Corollary 1 this shows that for increasing target sample sizes, the expected loss of the output of ANDA-S converges to the Bayes optimal and, with high probability over increasing source samples, ANDA-S will not query target sample points in the source support.

5.5 Experiments

Our experiments on synthetic data illustrate ANDA's adaptation ability and show that its classification performance compares favorably with baseline passive nearest neighbors. Experiments on challenging image classification tasks show that ANDA is a good candidate for correcting dataset bias. We discuss the results in relation to our theory.



(a) ANDA-Safe

(b) ANDA-Safe-EMMA

Figure 21: Visualization of synthetic data and query strategies for two versions of ANDA. Red and blue circles represent labeled source examples, black circles represent unqueried target examples, and green stars represent queried target examples.

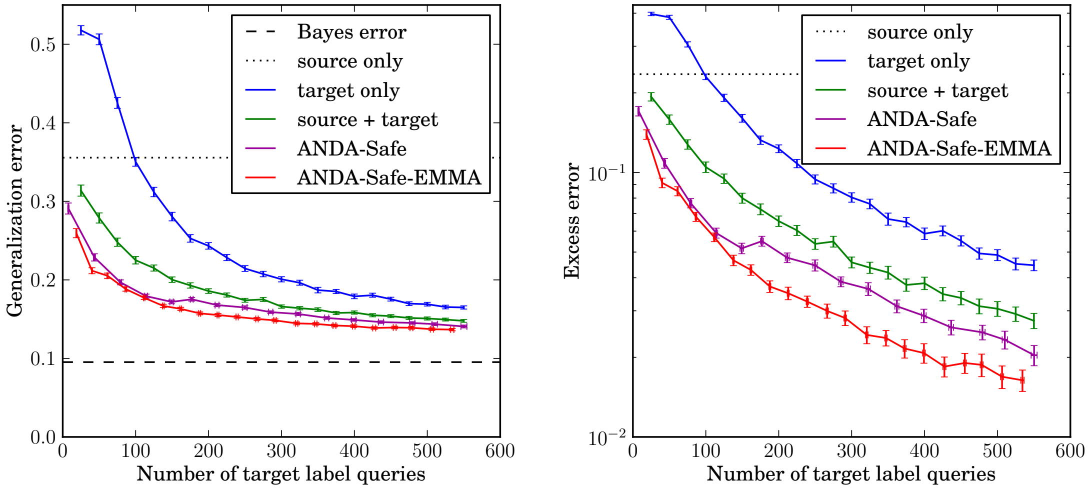
5.5.1 Synthetic Data

The source marginal D_S was taken to be the uniform distribution over $[-1, 0.5]^2$ and the target marginal D_T was set to uniform over $[-0.75, 1]^2$. This ensures enough source/target overlap so the source data is helpful in learning the target task but not sufficient to learn well. The regression function chosen for both tasks was

$$\eta(x_1, x_2) = \frac{1}{2} - \frac{(\sin(2\pi x_1)\sin(2\pi x_2))^{1/6}}{2}$$

for $(x_1, x_2) \in \mathbb{R}^2$. This creates a 4×4 checkerboard of mostly-positively and mostly-negatively labeled regions with noise on the boundaries where η crosses $1/2$. Training samples from this setting are pictured in Figure 21 along with query locations. Notice that queries are almost never made inside the source support, as our theory would suggest.

The baseline algorithms we compare against are the following. The "source only" algorithm predicts according to a k-NN classifier built on a source sample alone. The



(a) Linear scale

(b) Log scale

Figure 22: Experimental results on synthetic data. Error bars represent two standard errors, or roughly a 95% confidence interval.

"target only" algorithm creates a k-NN classifier on a random sample from the target, and "source + target" does the same but includes labeled data from a source sample as well.

We compare the generalization error of ANDA-Safe-EMMA and ANDA-Safe against these baselines across a range of unlabeled target sample sizes. Since the number of queries made by both ANDA-Safe-EMMA and ANDA-Safe increases with target sample size, this generates a range of query counts for the active algorithms. The baseline algorithms were given labeled target samples of sizes in the same range as these query counts. For all algorithms and target sample sizes we fixed $m_S = 3200$, $k = 7$, and $k' = 21$. Figure 22 shows the resulting generalization error (averaged over 100 independent trials) for each algorithm as a function of the number of target labels used.

Both active algorithms perform significantly better than the passive baselines in terms of the error they achieve per target label query. ANDA-Safe-EMMA also outperforms ANDA-Safe, since (as shown in Figure 21) achieves full coverage of the target region with many fewer queries.

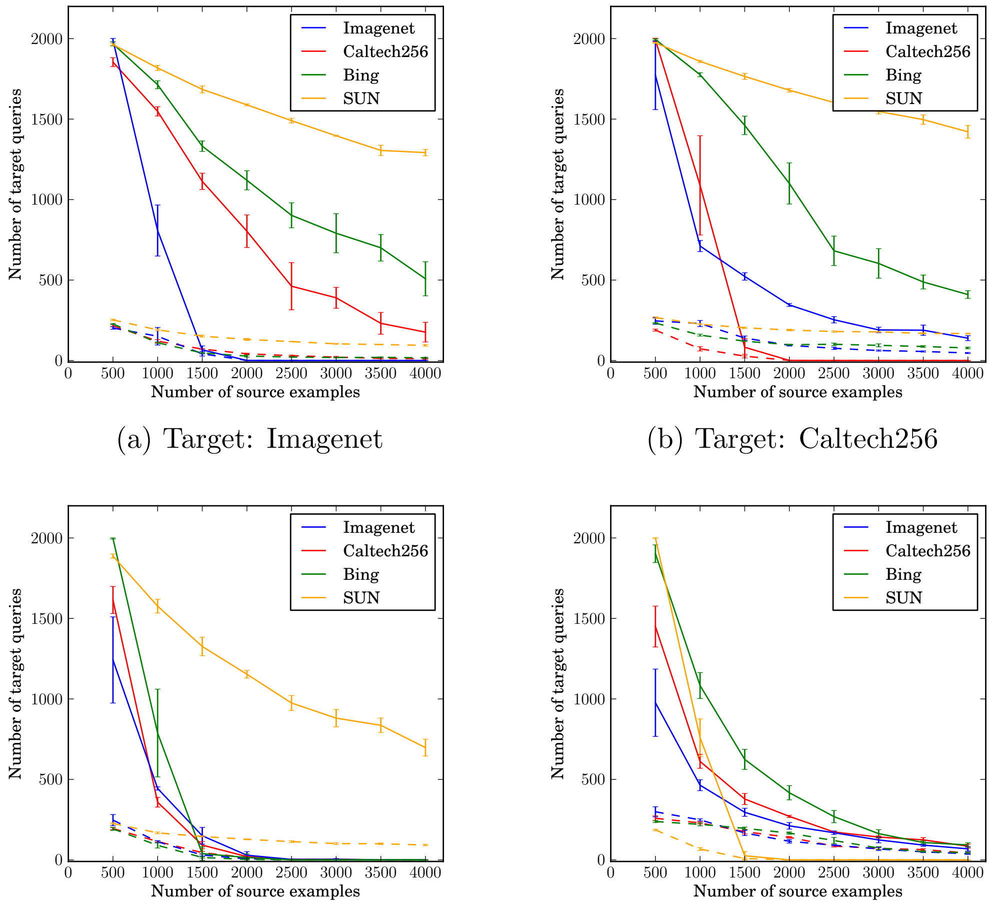
5.5.2 Image Classification

A major problem in building robust image classifiers is that the source of training images is often not the same as the source of images on which the classifier is expected to perform. This leads to the problem of *dataset bias*, which requires some form of domain adaptation to correct. [101] aligned and preprocessed several image datasets that provide a way of comparing domain adaptation algorithms on this problem. Even though these datasets are unlikely to satisfy the covariate shift setting exactly, we compare ANDA with baseline nearest neighbor classifiers to show that ANDA provides a partial solution to the dataset bias problem.

The task is to classify images according to the object in the image. We use the dense setup which contains four datasets (representing different domains) and 40 object classes. SIFT features for each image were precomputed and grouped into a bag-of-words representation with a 1000-word vocabulary. Despite the high dimensionality, we find that nearest neighbor methods work well on these datasets without further dimensionality reduction.

Three of the four datasets (Imagenet, Caltech256, and Bing) are *object-centric*, meaning the object corresponding to an image's class is usually centered and relatively large, while the fourth (SUN) is *scene-centric*, meaning the objects of interest are much more varied in position and scale [101]. Furthermore, the Bing dataset is known to contain some incorrectly labeled images as they were obtained via web search. These differences between datasets may explain several of our findings.

Number of target queries. Before testing classification performance, we run experiments to check how many target queries are made by both ANDA-Safe and ANDA-Safe-EMMA for each combination of source and target (including self pairs). This allows us to see similarities and differences in the datasets that are based on the



(a) Target: Imagenet

(b) Target: Caltech256

(c) Target: Bing

(d) Target: SUN

Figure 23: Number of target queries made by ANDA-Safe (solid lines) and ANDA-Safe-EMMA (dashed lines) for each source-target combination. Each plot fixes the target data and target sample size ($m_T = 2000$) while varying the source data, source sample size, and algorithm. Error bars represent two standard errors.

covariates alone rather than the effects of labeled information. To test this, we fixed $m_T = 2000$, $k = 25$, and $k' = 75$ and ran ANDA-Safe and ANDA-Safe-EMMA for a series of values of m_S. The resulting query counts (averaged over 5 independent trials) are shown in Figure 23.

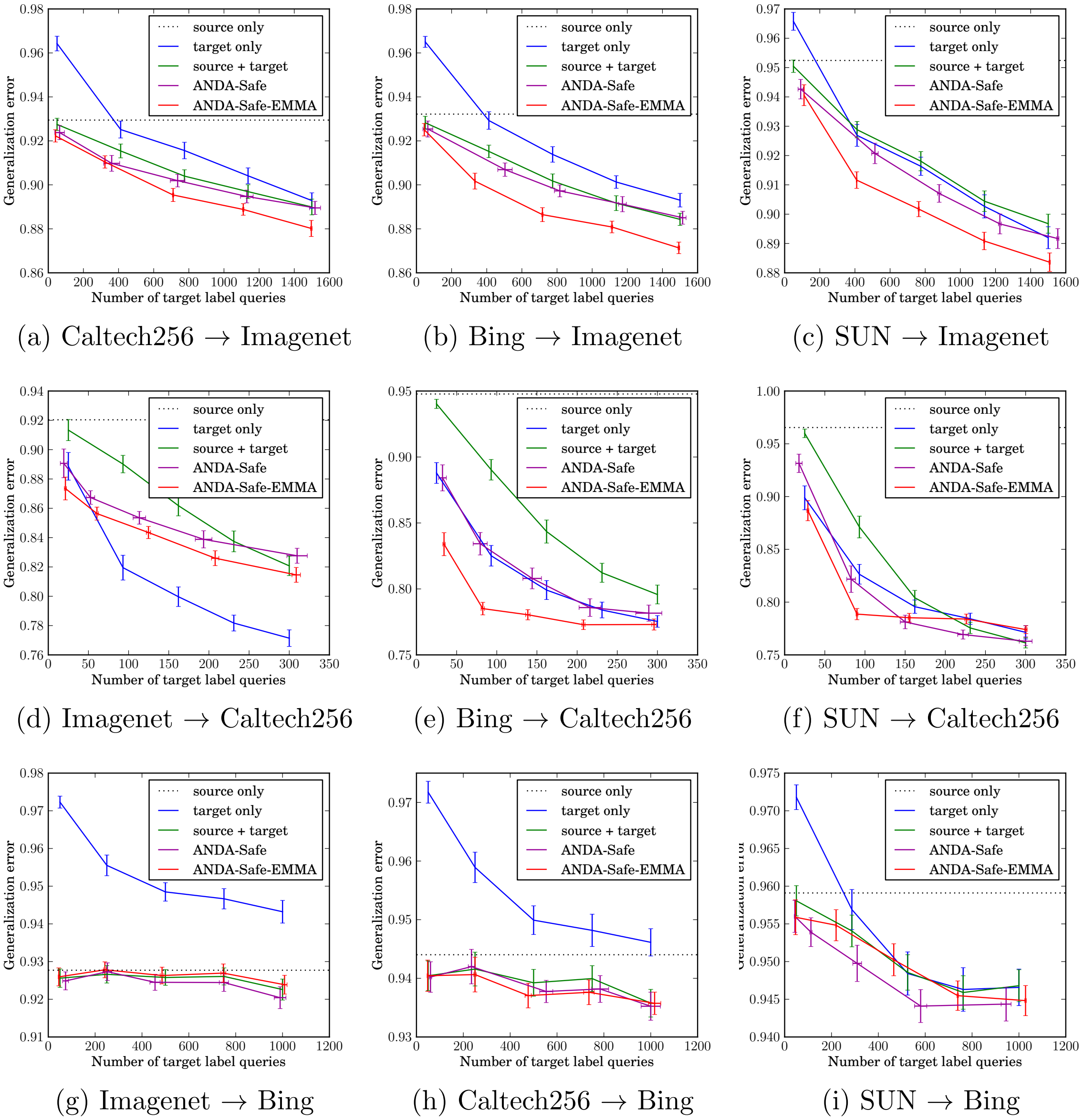
This is valuable in a few different ways. First, when the source and target examples are sampled from the same dataset, we would expect no target queries to be made when the source sample is large enough. Indeed, for all four datasets, neither

ANDA-Safe nor ANDA-Safe-EMMA makes any queries when $m_S \geq m_T$. This confirms a desirable property predicted by our theory: ANDA will automatically detect when to rely on source data alone and not waste label queries.

When the source and target are not the same, Figure 23 provides insight into how much overlap exists between each of the four datasets in the current representation. For instance, if the source support completely contains the target support, we would expect the number of target queries made by ANDA-Safe to rapidly approach zero as m_S increases. In contrast, if there are portions of the target support that do not overlap at all with the source support, the number of target queries made by ANDA-Safe will decrease more slowly, possibly approaching a positive constant rather than zero.

Based on this reasoning, we can draw several conclusions from Figure 23. Figure 23a tells us that Imagenet appears to have regions that are not covered by any of the other datasets, since the query counts for the other datasets decrease slowly. We can also order the other three datasets by how much mass Imagenet has outside of their support. We can make a similar conclusion for Caltech256 based on Figure 23b: Caltech256 has a very small mass uncovered by Imagenet (Imagenet's curve decreases quickly, but never quite reaches zero) while it has significant portions uncovered by Bing and SUN. Figure 23c shows that Bing is an interesting target because the query counts for sources Imagenet and Caltech256 decrease to zero as fast as when Bing itself is the source. This seems to indicate that Bing is completely contained within the support of Imagenet and Caltech256 but has significant mass uncovered by SUN. Figure 23d shows that SUN appears to be the smallest dataset since the query counts decrease quickly for every source dataset.

Classification performance. Our next experiments demonstrate ANDA's ability to correct for dataset bias. For each of the twelve source-target permutations, we



(a) Caltech256 → Imagenet

(b) Bing → Imagenet

(c) SUN → Imagenet

(d) Imagenet → Caltech256

(e) Bing → Caltech256

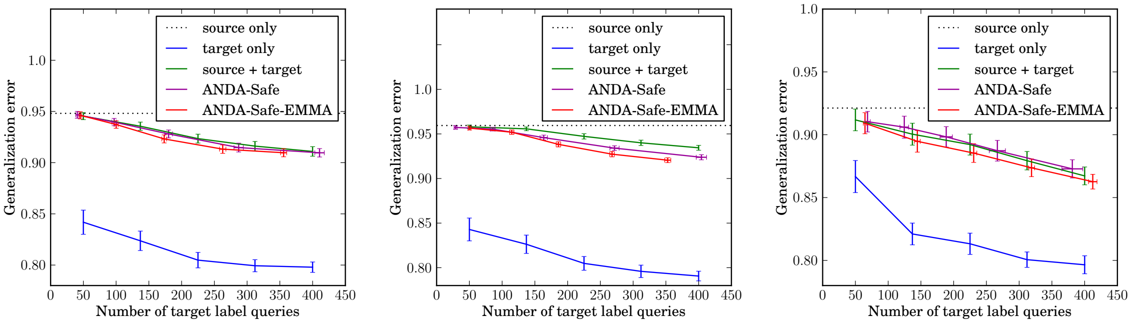
(f) SUN → Caltech256

(g) Imagenet → Bing

(h) Caltech256 → Bing

(i) SUN → Bing

Figure 24: Results on image classification task for object-centric target datasets. Each plot caption is of the form *source* → *target*. Error bars represent two standard errors.



(a) Imagenet → SUN

(b) Caltech256 → SUN

(c) Bing → SUN

Figure 25: Results for cases in which the scene-based SUN dataset is the target. Each plot caption is of the form *source* → *target*. Error bars represent two standard errors.

compared the same algorithms described in Section 5.5.1 and used the same method for generating a range of query counts. For all algorithms and target sample sizes we fixed $m_S = 2000$, $k = 25$, and $k' = 75$. Figures 24 and 25 show the resulting generalization error (estimated from test sets of 1000 examples and averaged over 50 independent trials) for each algorithm as a function of the number of target labels used. The error values reported here[2] are on the same order as those in the results of [101], but since different sample sizes were used, they cannot be directly compared.

Overall we find that our methods (especially ANDA-Safe-EMMA) successfully correct for dataset bias in image classification, also showing that ANDA is robust to small violations of our theory's assumptions. For all 12 pairs of datasets, ANDA-Safe-EMMA performs better than using source data alone (adding in target examples always helps). Even more encouraging, on 6 of the 12 pairs (6 of 9 with object-centric targets), it performs better than the target-only baseline (indicating that having source examples allows us to make more efficient use of target labels) and on 7 of the 12 (6 of 9 with object-centric targets) it outperforms the passive source + target baseline (and never performs worse). The results are particularly promising when Imagenet is the target

[2] Note that since there are 40 classes, guessing labels uniformly at random results in a generalization error of 97.5%.

(Figures 24a, 24b, and 24c). In all three of these cases and nearly all query counts, ANDA-Safe-EMMA outperforms all other methods.

When Caltech256 is the target (Figures 24d, 24e, and 24f), the target-only baseline outperforms the other methods for high enough query counts. This is likely because Caltech256 is less noisy than the other datasets, so the noisy source data is helpful in the absence of target data but harmful when enough target data is available. Notice that for all three of these cases, ANDA-Safe-EMMA has the best accuracy at small query counts, exemplifying its efficiency at making use of labels when it only makes a few queries.

When Bing is the target (Figures 24g, 24h, and 24i), neither active algorithm performs better (or worse) than the passive baseline. Bing was previously known to be noisier than the other two datasets [101], and further evidence of this can be found in the observation that the source-only baselines (for both Caltech256 and Imagenet) perform better than Bing's target-only baseline. This means the target queries from Bing are generally less informative than source examples, regardless of where the queries are made, resulting in all the source-target combination methods performing equally well.

Figure 25 shows the three cases in which SUN is the target. These cases exhibit particularly poor performance for all methods incorporating source examples. This is possibly due to the fact that SUN appears to be a relatively clean and localized dataset (in this representation), so oppositely-labeled source examples from outside this region may be misleading the k-NN predictor. These cases also point to what may be a more general issue with this feature representation: while scene-centric data can be a helpful source for an object-centric target task, the reverse appears not to be true.

These experiments also give evidence for the types of situations in which ANDA-Safe-EMMA has an advantage over ANDA-Safe. ANDA-Safe-EMMA has the greatest improvement over ANDA-Safe when Imagenet and Caltech256 are the target tasks. From our analysis in the previous section, these are the two datasets that are the most spread out and are the least covered by the other datasets. Intuitively, this is exactly what we would expect, since the approximation algorithm of ANDA-Safe-EMMA will have a bigger effect when there are large, contiguous regions containing only target examples rather than cases where source and target examples are uniformly interspersed.

5.6 Discussion

Domain adaptation is crucial to nearly every application of machine learning. In many of these applications it makes sense to allow learners to request labels of selected examples from the target task. We give the first formal analysis of this setting, proving that ANDA successfully adapts from a source task to a target task by automatically determining where it needs labeled target examples. We prove not only that ANDA will have small classification error, but also that it will not make queries where it does not need to and that the queries it does make are necessary. We also give experimental evidence that ANDA can correct for dataset bias in image classification.

One feature that ANDA exhibits in our experimental analysis, but is not captured by our theory, is that approximately finding a minimum (k, k')-NN-cover as done by ANDA-Safe-EMMA can lead to improved performance over the use of larger covers. Fully capturing the sample complexity of ANDA-Safe and ANDA-Safe-EMMA will likely require and lead to a more general understanding of selective sampling or sample compressions for nearest neighbor methods. This is an exciting research avenue on its own and left to future work. Our notion of nearest neighbor covers may prove beneficial for this more general theory.

Another interesting direction would be to explore how different data representations effect ANDA's ability to transfer knowledge. For the image classification problem we study here, there is some recent evidence that decaf7 features may be more amenable to domain adaptation than the SIFT bag-of-words features we use here [100].

Here we use active learning for handling covariate shift. However, we believe active learning should be a powerful tool for detecting and correcting label shift as well and that algorithms for performing this task are greatly needed. More generally, we have shown that the query ability of active learning can provide great advantages to domain adaptation algorithms based on k-NN classifiers. However, the benefits active learning has for domain adaptation are by no means limited to nearest-neighbor-based algorithms. We believe that two aspects of ANDA will inspire future work on developing learning methods that perform well under changing tasks: (1) being active and (2) automatically adapting to the relatedness of source and target, that is, not needing any parameter tuning to account for the relatedness of specific source and target tasks at hand. We hope that future research will further develop these aspects and the intriguing relationship between them.

CHAPTER VI

SENSOR CONSENSUS GAME FOR HIGH-NOISE ACTIVE LEARNING

As discussed in Chapter 2, most prior work on active learning has focused only on the single-agent low-noise setting, with a learning algorithm obtaining labels from a single, nearly-perfect labeling entity. In large part this is because the effectiveness of active learning is known to quickly degrade as noise rates become high [22]. In this chapter, we introduce and analyze a novel setting where label information is held by highly-noisy low-power agents (such as sensors or micro-robots). We show how by first using simple game-theoretic dynamics among the agents we can quickly approximately denoise the system. This allows us to exploit the power of active learning (especially recent advances in agnostic active learning), leading to efficient learning from only a small number of expensive queries. This ability to drastically improve label complexity over passive learning even in the presence of very noisy data is our final example of a new capability of active learning.

We specifically examine an important setting relevant to many engineered systems where we have a large number of low-power agents (e.g., sensors). These agents are each measuring some quantity, such as whether there is a high or low concentration of a dangerous chemical at their location, but they are assumed to be highly noisy. We also have a center, far away from the region being monitored, which has the ability to query these agents to determine their state. Viewing the agents as examples, and their states as noisy labels, the goal of the center is to learn a good approximation to the true target function (e.g., the true boundary of the high-concentration region for the chemical being monitored) from a small number of label queries. However, because of the high noise rate, learning this function directly would require a very large number of queries to be made (for noise rate η, one would necessarily require $\Omega(\frac{1}{(1/2-\eta)^2})$ queries [9]). The question we address in this chapter is to what extent this difficulty can be alleviated by providing the agents the ability to engage in a small amount of local communication among themselves.

What we show is that by using local communication and applying simple robust state-changing rules such as following natural game-theoretic dynamics, randomly distributed agents can modify their state in a way that greatly de-noises the system without destroying the true target boundary. This then nicely meshes with recent advances in agnostic active learning [5], allowing for the center to learn a good approximation to the target function from a small number of queries to the agents. In particular, in addition to proving theoretical guarantees on the denoising power of game-theoretic agent dynamics, we also show experimentally that a version of the agnostic active learning algorithm of [5], when combined with these dynamics, indeed is able to achieve low error from a small number of queries, outperforming active and passive learning algorithms without the best-response denoising step, as well as outperforming passive learning algorithms with denoising. More broadly, engineered systems such as sensor networks are especially well-suited to active learning because components may be able to communicate among themselves to reduce noise, and the designer has some control over how they are distributed and so assumptions such as a uniform or other "nice" distribution on data are reasonable. We focus in this chapter primarily on the natural case of linear separator decision boundaries but many of our results extend directly to more general decision boundaries as well.

6.1 Related Work

There has been extensive work analyzing the performance of simple dynamics in consensus games [26, 43, 78, 67, 8, 7]. However, this line of work has focused on getting to *some* equilibria or states of low social cost, while we are primarily interested in getting near a *specific configuration*, which as we show below is an approximate equilibrium.

6.2 Preliminaries

First we describe the basic setup underlying the remainder of this chapter. We assume we have a large number N of agents (e.g., sensors) distributed uniformly at random in a geometric region, which for concreteness we consider to be the unit ball in R^d. There is an unknown linear separator such that in the initial state, each sensor on the positive side of this separator is positive independently with probability $\geq 1-\eta$, and each on the negative side is negative independently with probability $\geq 1-\eta$. The quantity $\eta < 1/2$ is the *noise rate*.

6.2.1 The Sensor Consensus Game

The sensors will denoise themselves by viewing themselves as players in a certain consensus game, and performing a simple dynamics in this game leading towards a specific ϵ-equilibrium.

Specifically, the game is defined as follows, and is parameterized by a communication radius r, which should be thought of as small. Consider a graph where the sensors are vertices, and any two sensors within distance r are connected by an edge. Each sensor is in one of two states, positive or negative. The *payoff* a sensor receives is its correlation with its neighbors: the fraction of neighbors in the same state as it minus the fraction in the opposite state. So, if a sensor is in the same state as all its neighbors then its payoff is 1, if it is in the opposite state of all its neighbors then its payoff is -1, and if sensors are in uniformly random states then the expected payoff is 0. Note that the states of highest social welfare (highest sum of utilities) are the all-positive and all-negative states, which are *not* what we are looking for. Instead, we want sensors to approach a different near-equilibrium state in which (most of) those on the positive side of the target separator are positive and (most of) those on the negative side of the target separator are negative. For this reason, we need to be particularly careful with the specific dynamics followed by the sensors.

We begin with a simple lemma that for sufficiently large N, the target function (i.e., all sensors on the positive side of the target separator in the positive state and the rest in the negative state) is an ϵ-equilibrium, in that no sensor has more than ϵ incentive to deviate.

Lemma 6. *For any $\epsilon, \delta > 0$, for sufficiently large N, with probability $1 - \delta$ the target function is an ϵ-equilibrium.*

Proof Sketch: The target function fails to be an ϵ-equilibrium iff there exists a sensor for which more than an $\epsilon/2$ fraction of its neighbors lie on the opposite side of the separator. Fix one sensor x and consider the probability this occurs to x, over the random placement of the $N-1$ other sensors. Since the probability mass of the r-ball around x is at least $(r/2)^d$ (see discussion in proof of Theorem 13), so long as $N - 1 \geq (2/r)^d \cdot \max[8, \frac{4}{\epsilon^2}] \ln(\frac{2N}{\delta})$, with probability $1 - \frac{\delta}{2N}$, point x will have $m_x \geq \frac{2}{\epsilon^2} \ln(\frac{2N}{\delta})$ neighbors (by Chernoff bounds), each of which is at least as likely to be on x's side of the target as on the other side. Thus, by Hoeffding bounds, the probability that more than a $\frac{1}{2} + \frac{\epsilon}{2}$ fraction lie on the wrong side is at most $\frac{\delta}{2N} + \frac{\delta}{2N} = \frac{\delta}{N}$. The result then follows by union bound over all N sensors. For a bit tighter argument and a concrete bound on N, see the proof of Theorem 13 which essentially has this as a special case. □

Lemma 6 motivates the use of best-response dynamics for denoising. Specifically, we consider a dynamics in which each sensor switches to the majority vote of all the other sensors in its neighborhood. We analyze below the denoising power of this dynamics under both synchronous and asynchronous update models.

6.2.2 Margin-based Active Learning

Recently, Awasthi et al. [5] gave the first polynomial-time active learning algorithm able to learn linear separators to error ϵ over the uniform distribution in the presence of agnostic noise of rate $O(\epsilon)$. Moreover, the algorithm does so with optimal query complexity of $O(d \log 1/\epsilon)$. This algorithm is ideally suited to our setting because (a) the sensors are uniformly distributed, and (b) the result of best response dynamics is noise that is low but potentially highly coupled (hence, fitting the low-noise agnostic model). In our experiments (Section 6.5) we show that indeed this algorithm when combined with best-response dynamics achieves low error from a small number of queries, outperforming active and passive learning algorithms without the best-response denoising step, as well as outperforming passive learning algorithms with denoising.

Here, we briefly describe the algorithm of [5] and the intuition behind it. At high level, the algorithm proceeds through several rounds, in each performing the following operations (see also Figure 26):

Instance space localization: Request labels for a random sample of points within a band of width $b_k = O(2^{-k})$ around the boundary of the previous hypothesis w_k.

Concept space localization: Solve for hypothesis vector w_{k+1} by minimizing hinge loss subject to the constraint that w_{k+1} lie within a radius r_k from w_k; that is, $||w_{k+1} - w_k|| \leq r_k$.

[5, 52, 108] show that by setting the parameters appropriately (in particular, $b_k = \Theta(1/2^k)$ and $r_k = \Theta(1/2^k)$), the algorithm will achieve error ϵ using only $k = O(\log 1/\epsilon)$ rounds, with $O(d)$ label requests per round. In particular, a key idea of their analysis is to decompose, in round k, the error of a candidate classifier w as its error outside margin b_k of the current separator plus its error inside margin b_k, and to

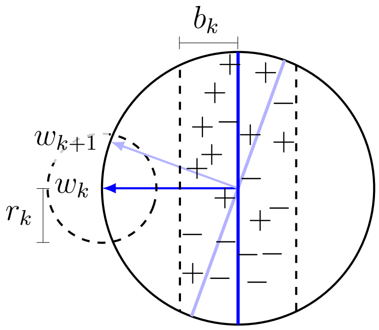


Figure 26: The margin-based active learning algorithm after iteration k. The algorithm samples points within margin b_k of the current weight vector w_k and then minimizes the hinge loss over this sample subject to the constraint that the new weight vector w_{k+1} is within distance r_k from w_k.

prove that for these parameters, a small constant error inside the margin suffices to reduce overall error by a constant factor. A second key part is that by constraining the search for w_{k+1} to vectors within a ball of radius r_k about w_k, they show that hinge-loss acts as a sufficiently faithful proxy for 0-1 loss.

6.3 Simultaneous-move Dynamics

We start by providing a positive theoretical guarantee for one-round simultaneous move dynamics.

Theorem 13. *If*

$$N \geq \frac{2}{(r/2)^d(1/2-\eta)^2} \ln\left(\frac{1}{(r/2)^d(1/2-\eta)^2\delta}\right) + 1$$

then, with probability at least $1-\delta$, after one synchronous consensus update every sensor at distance at most r from the separator has the correct label.

The result follows from a union bound and an application of Bernstein's inequality on the difference between the number of positively and negatively labeled sensors within radius-r balls around sensors.

Note that since a band of width $2r$ about a linear separator has probability mass $O(r\sqrt{d})$, Theorem 13 implies that with high probability one synchronous update denoises all but an $O(r\sqrt{d})$ fraction of the sensors. In fact, Theorem 13 does not require the separator to be linear, and so this conclusion applies to any decision boundary with similar surface area, such as an intersection of a constant number of halfspaces or a decision surface of bounded curvature.

Proof (Theorem 13): Fix a point x in the sample at distance $\geq r$ from the separator and consider the ball of radius r centered at x. Let n_+ be the number of correctly labeled points within the ball and n_- be the number of incorrectly labeled points within the ball. Now consider the random variable $\Delta = n_- - n_+$. Denoising x can give it the incorrect label only if $\Delta \geq 0$, so we would like to bound the probability that this happens. We can express Δ as the sum of $N-1$ independent random variables Δ_i taking on value 0 for points outside the ball around x, 1 for incorrectly labeled points inside the ball, or -1 for correct labels inside the ball. Let V be the measure of the ball centered at x (which may be less than r^d if x is near the boundary of the unit ball). Then since the ball lies entirely on one side of the separator we have

$$\mathbb{E}[\Delta_i] = (1-V)\cdot 0 + V\eta - V(1-\eta) = -V(1-2\eta).$$

Since $|\Delta_i| \leq 1$ we can take $M = 2$ in Bernstein's theorem. We can also calculate that $\mathrm{Var}[\Delta_i] \leq \mathbb{E}[\Delta_i^2] = V$. Thus the probability that the point x is updated incorrectly is

$$\begin{aligned}
\Pr\left[\sum_{i=1}^{N-1} \Delta_i \geq 0\right] &= \Pr\left[\sum_{i=1}^{N-1} \Delta_i - \mathbb{E}\Big[\sum_{i=1}^{N-1} \Delta_i\Big] \geq (N-1)V(1-2\eta)\right] \\
&\leq \exp\left(\frac{-(N-1)^2V^2(1-2\eta)^2}{2\big((N-1)V + 2(N-1)V(1-2\eta)/3\big)}\right) \\
&\leq \exp\left(\frac{-(N-1)V(1-2\eta)^2}{2+4(1-2\eta)/3}\right) \\
&\leq \exp\left(-(N-1)V(1/2-\eta)^2\right) \\
&\leq \exp\left(-(N-1)(r/2)^d(1/2-\eta)^2\right),
\end{aligned}$$

where in the last step we lower bound the measure V of the ball around r by the measure of the sphere of radius $r/2$ inscribed in its intersection with the unit ball. Taking a union bound over all N points, it suffices to have $e^{-(N-1)(r/2)^d(1/2-\eta)^2} \leq \delta/N$, or equivalently

$$N - 1 \geq \frac{1}{(r/2)^d(1/2-\eta)^2}\left(\ln N + \ln\frac{1}{\delta}\right).$$

Using the fact that $\ln x \leq \alpha x - \ln \alpha - 1$ for all $x, \alpha > 0$ yields the claimed bound on N. □

We can now combine this result with the efficient agnostic active learning algorithm of [5]. In particular, applying the most recent analysis of [52, 108] of the algorithm of [5], we get the following bound on the number of queries needed to efficiently learn to accuracy $1-\epsilon$ with probability $1-\delta$.

Corollary 2. *There exists constant $c_1 > 0$ such that for $r \leq \epsilon/(c_1\sqrt{d})$, and N satisfying the bound of Theorem 13, if sensors are each initially in agreement with the target linear separator independently with probability at least $1-\eta$, then one round of best-response dynamics is sufficient such that the agnostic active learning algorithm of [5] will efficiently learn to error ϵ using only $O(d \log 1/\epsilon)$ queries to sensors.*

In Section 6.5 we implement this algorithm and show that experimentally it learns a low-error decision rule even in cases where the initial value of η is quite high.

6.4 Asynchronous Dynamics

In this section we discuss the asynchronous update setting in which the sensors update their label one at a time. We show that the results can differ drastically depending on the order in which the sensors perform their updates.

6.4.1 Arbitrary-order Asynchronous Dynamics

We contrast the above positive result with a negative result for arbitrary-order asynchronous moves. In particular, we show that for any $d \geq 1$, for sufficiently large

N, with high probability there exists an update order that will cause all sensors to become negative.

Theorem 14. *For some absolute constant $c > 0$, if $r \leq 1/2$ and sensors begin with noise rate η, and*

$$N \geq \frac{16}{(cr)^d \phi^2}\left(\ln \frac{8}{(cr)^d \phi^2} + \ln \frac{1}{\delta} \right),$$

where $\phi = \phi(\eta) = \min(\eta, 1/2 - \eta)$, then with probability at least $1 - \delta$ there exists an ordering of the agents so that asynchronous updates in this order cause all points to have the same label.

The basic idea is to have the sensors update from the far negative side of the separator toward the positive side, in order of distance from the separator. The first half of them will all flip negative as long as N is large relative to $1/(1/2 - \eta)$ (similar to the argument of Theorem 13). The rest will all flip negative because half of the neighboring sensors are already negative and the other half will not be all positive because N is large relative to $1/\eta$.

Proof. We first give a sketch for the $d = 1$ case before moving on to the more general setting. Consider the case $d = 1$ and a target function $x > 0$. Each subinterval of $[-1, 1]$ of width r has probability mass $r/2$, and let $m = rN/2$ be the expected number of points within such an interval. The given value of N is sufficiently large that with high probability, all such intervals in the initial state have both a positive count and a negative count that are within $\pm\frac{\phi}{4}m$ of their expectations. This implies that if sensors update left-to-right, initially all sensors will (correctly) flip to negative, because their neighborhoods have more negative points than positive points. But then when the "wave" of sensors reaches the positive region, they will continue (incorrectly) flipping to negative because the at least $m(1 - \frac{\phi}{2})$ negative points in the left-half of their neighborhood will outweigh the at most $(1 - \eta + \frac{\phi}{4})m$ positive points in the right-half of their neighborhood.

Now we give the proof in full generality for $d \geq 1$. Suppose the labeling is given by $\text{sign}(w \cdot x)$. We show that if sensors are updated in increasing order of $w \cdot x$ (from most negative to most positive) then with high probability all sensors will update to negative labels.

Consider what we see when we come to update the sensor at x. Assuming we have not yet failed (given a positive label), all of the points x' with $w \cdot x' < w \cdot x$ are labeled negative, while those with $w \cdot x' > w \cdot x$ are unchanged from their original states, and so are still labeled with independent uniform noise. As in the proof of Theorem 13, we apply Bernstein's theorem to the difference Δ between the number of negative and positive points in the neighborhood of x, which we write as a sum of $(N-1)$ independent variables Δ_i. The expected labels of the nearby points depend on the location of x, so we consider three regions: $w \cdot x \leq -r$, $w \cdot x \geq 0$, and $-r < w \cdot x < 0$.

Let V denote the probability mass of the ball of radius r around x. In all cases the variance is bounded by $\text{Var}[\Delta_i] \leq \mathbb{E}[\Delta_i^2] = V \leq r^d$.

In the first region ($w \cdot x \leq -r$) we can use the same analysis from Theorem 13 to find that $\mathbb{E}[\Delta_i] \leq -V(1-2\eta) \leq -(r/2)^d(1-2\eta)$, since the ball around x never crosses the separator and any sensors previously updated to negative labels cannot hurt.

In the second region ($w \cdot x \leq 0$) we can use a similar analysis, bounding

$$\mathbb{E}[\Delta_i] \leq -V/2 + (1-\eta)V/2 = -\eta V/2 \leq -\frac{1}{2}(r/2)^d,$$

since the measure of the (positive biased) half of the ball further from the separator than x is never larger than the measure of the remaining (all negative) half of the ball.

In the final region ($0 < w \cdot x < r$), we must take a little more care, as the measure of the all-negative half of the ball may be less than the measure of the unexamined side, which may be positive-biased due to crossing the separator. To analyze this

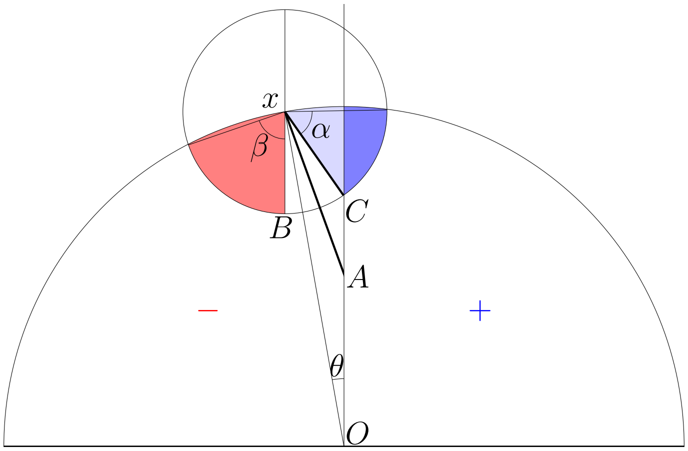


Figure 27: A ball around x intersecting the decision boundary and the boundary of the unit ball.

case, we project onto the 2-dimensional space spanned by x and w. The worst case is clearly when x is on the surface of the ball, as shown in Figure 27.

Any point in the red region is known to have a negative label, while points in the dark blue region are biased towards positive labels. We first show that the red region is bigger by showing that the angle α subtended by the dark blue region is smaller than the angle β of the red region. Construct the segment $\overline{xA}$ by reflecting the segment $\overline{xB}$ about the line $\overline{xO}$ and extending it to the separator. Note that the angle $\angle OxA$ is the same as the angle θ between x and the separator. We find that $\alpha \leq \beta$ precisely when $xA \geq xC = r$. Indeed, by considering the isosceles triangle $\triangle AxO$ we see that $xA = 1/(2\cos\theta) \geq 1/2$. So as long as $r \leq 1/2$ we have $\beta \geq \alpha$. Thus, since the projection of the uniform distribution over the unit ball onto this plane is radially symmetric, the red region has more probability mass than the blue region.

We can now calculate for this case

$$\begin{aligned}\mathbb{E}[\Delta_i] &\leq (-1)[\text{measure of red}] + (1-2\eta)[\text{measure of blue}] + (2\eta-1)[\text{measure of white}] \\ &\leq -2\eta[\text{measure of red}].\end{aligned}$$

Note that although the projection does not make sense for $d = 1$ the result obviously still holds (as there are no points near both the separator and the boundary of the unit ball). We can lower bound the measure of the red region by the measure of the sphere inscribed in the sector, which has radius at least cr for some $0 < c < 1/2$ as long as $r \leq 1/2$ (since β is bounded away from 0 in this range of r).

Now we see that for any x the expected label satisfies

$$\mathbb{E}[\Delta_i] \leq -\frac{1}{2}(cr)^d \min(\eta, 1/2 - \eta).$$

Letting $\phi = \min(\eta, \frac{1}{2} - \eta)$, we find that the probability of giving a positive label on any given update is

$$\begin{aligned}\Pr[\Delta \geq 0] &\leq \exp\left(\frac{-\frac{1}{4}(N-1)^2(cr)^{2d}\phi^2/2}{(N-1)r^d + (N-1)(cr)^d\phi/3}\right)\\ &= \exp\left(\frac{-\frac{1}{4}(N-1)(cr)^d\phi^2}{1+\phi/3}\right)\\ &= \exp\big(-(N-1)(cr)^d\phi^2/8\big)\end{aligned}$$

By the union bound, we find that

$$N \geq \frac{16}{(cr)^d\phi^2}\left(\ln\frac{8}{(cr)^d\phi^2} + \ln\frac{1}{\delta}\right)$$

suffices to ensure that with probability at least $1-\delta$ all sensors are updated to negative labels. □

Note that if $r = O(1/\sqrt{d})$ then we can lower bound all of the relevant measures in the preceding proof by $\Theta(r^d)$ rather than $(\Theta(r))^d$, to see that

$$N \geq \Omega\left(\frac{1}{r^d\phi^2}\left(\ln\frac{1}{r\phi} + \ln\frac{1}{\delta}\right)\right)$$

suffices.

6.4.2 Random Order Dynamics

While Theorem 14 shows that there *exist* bad orderings for asynchronous dynamics, we now show that we can get positive theoretical guarantees for *random order* best-response dynamics (that is, sensors each update only once, but they do so in a random order).

The high level idea of the analysis is to partition the sensors into three sets: those that are within distance r of the target separator, those at distance between r and $2r$ from the target separator, and then all the rest. For those at distance $< r$ from the separator we will make no guarantees: they might update incorrectly when it is their turn to move due to their neighbors on the other side of the target. Those at distance between r and $2r$ from the separator might also update incorrectly (due to "corruption" from neighbors at distance $< r$ from the separator that had earlier updated incorrectly) but we will show that with high probability this only happens in the last $1/4$ of the ordering. I.e., within the first $3N/4$ updates, with high probability there are no incorrect updates by sensors at distance between r and $2r$ from the target. Finally, we show that with high probability, those at distance greater than $2r$ *never* update incorrectly. This last part of the argument follows from two facts: (1) with high probability all such points begin with more correctly-labeled neighbors than incorrectly-labeled neighbors (so they will update correctly so long as no neighbors have previously updated incorrectly), and (2) after $3N/4$ total updates have been made, with high probability more than half of the neighbors of each such point have already (correctly) updated, and so those points will now update correctly no matter what their remaining neighbors do. Our argument for the sensors at distance in $[r, 2r]$ requires r to be small compared to $(\frac{1}{2} - \eta)/\sqrt{d}$, and the final error is $O(r\sqrt{d})$, so the conclusion is we have a total error less than ϵ for $r < c \min[\frac{1}{2} - \eta, \epsilon]/\sqrt{d}$ for some absolute constant c.

We begin with a key lemma. For any given sensor, define its inside-neighbors to be its neighbors in the direction of the target separator and its outside-neighbors to be its neighbors away from the target separator. Also, let $\gamma = 1/2 - \eta$.

Lemma 7. *For any $c_1, c_2 > 0$ there exist $c_3, c_4 > 0$ such that for $r \leq \frac{\gamma}{c_3\sqrt{d}}$ and*

$$N \geq \frac{c_4}{(r/2)^d \gamma^2} \ln \frac{1}{r^d \gamma \delta},$$

with probability $1 - \delta$, each sensor x at distance between r and $2r$ from the target separator has $m_x \geq \frac{c_1}{\gamma^2} \ln(4N/\delta)$ neighbors, and furthermore the number of inside-neighbors of x that move before x is within $\pm \frac{\gamma}{c_2} m_x$ of the number of outside neighbors of x that move before x.

Proof. First, the guarantee on m_x follows immediately from the fact that the probability mass of the ball around each sensor x is at least $(r/2)^d$, so for appropriate c_4 the expected value of m_x is at least $\max[8, \frac{2c_1}{\gamma^2}] \ln(4N/\delta)$, and then applying Hoeffding bounds [58, 27] and the union bound. Now, fix some sensor x and let us first assume the ball of radius r about x does not cross the unit sphere. Because this is random-order dynamics, if x is the kth sensor to move within its neighborhood, the $k-1$ sensors that move earlier are each equally likely to be an inside-neighbor or an outside-neighbor. So the question reduces to: if we flip $k - 1 \leq m_x$ fair coins, what is the probability that the number of heads differs from the number of tails by more than $\frac{\gamma}{c_2} m_x$. For $m_x \geq 2(\frac{c_2}{\gamma})^2 \ln(4N/\delta)$, this is at most $\delta/(2N)$ by Hoeffding bounds.

Now, if the ball of radius r about x does cross the unit sphere, then a random neighbor is slightly more likely to be an inside-neighbor than an outside-neighbor. We can analyze this difference in probabilities as follows. First, in the worst case, x is at distance exactly $2r$ from the separator, and is right on the edge of the unit ball. So we can define our coordinate system to view x as being at location $(2r, \sqrt{1-4r^2}, 0, \ldots, 0)$. Now, consider adding to x a random offset y in the r-ball. We want to look at the probability that $x + y$ has Euclidean length less than 1 conditioned on the first coordinate of y being negative compared to this probability conditioned on the first coordinate of y being positive. Notice that because the second coordinate of x is nearly 1, if $y_2 \leq -cr^2$ for appropriate c then $x + y$ has length less than 1 no matter what the other coordinates of y are (worst-case is if $y_1 = r$ but even that adds at most $O(r^2)$ to the squared-length). On the other hand, if $y_2 \geq cr^2$ then $x + y$ has length greater than 1 also no matter what the other coordinates of y are. So, it is only in between that the value of y_1 matters. But notice that the distribution over y_2 has maximum density $O(\sqrt{d}/r)$. So, with probability nearly $1/2$, the point is inside the unit ball for sure, with probability nearly $1/2$ the point is outside the unit ball for sure, and only with probability $O(r^2\sqrt{d}/r) = O(r\sqrt{d})$ does the y_1 coordinate make any difference at all.

Therefore, the difference in probabilities is only $O(r\sqrt{d})$, which is at most $\frac{\gamma}{2c_2}$ for appropriate choice of constant c_3. The result follows by applying Hoeffding bounds to the $\frac{\gamma}{2c_2}$ gap that remains. □

Theorem 15. *For some absolute constants c_3, c_4, for $r \leq \frac{\gamma}{c_3\sqrt{d}}$ and*

$$N \geq \frac{c_4}{(r/2)^d\gamma^2} \ln \frac{1}{r^d\gamma\delta},$$

in random order dynamics, with probability $1-\delta$ all sensors at distance greater than $2r$ from the target separator update correctly.

Proof. We begin by using Lemma 7 to argue that with high probability, no points at distance between r and $2r$ from the separator update incorrectly within the first $3N/4$ updates (which immediately implies that all points at distance greater than $2r$ update correctly as well, since by Theorem 13, with high probability they begin with more correctly-labeled neighbors than incorrectly-labeled neighbors and their neighborhood only becomes more favorable). In particular, for any given such point, the concern is that some of its inside-neighbors may have previously updated incorrectly. However, we use two facts: (1) by Lemma 7, we can set c_4 so that with high probability the total contribution of neighbors that have already updated is at most $\frac{\gamma}{8}m_x$ in the incorrect direction (since the outside-neighbors will have updated correctly, by induction), and (2) by standard concentration inequalities [58, 27], with high probability at least $\frac{1}{8}m_x$ neighbors of x have *not* yet updated. These $\frac{1}{8}m_x$ un-updated neighbors together have in expectation a $\frac{\gamma}{4}m_x$ bias in the correct direction, and so with high probability have greater than a $\frac{\gamma}{8}m_x$ correct bias for sufficiently large m_x (sufficiently large c_1 in

Lemma 7). So, with high probability this overcomes the at most $\frac{7}{8}m_x$ incorrect bias of neighbors that have already updated, and so the points will indeed update correctly as desired. Finally, we consider the points of distance $\geq 2r$. Within the first $\frac{3}{4}N$ updates, with high probability they will all update correctly as argued above. Now consider time $\frac{3}{4}N$. For each such point, in expectation $\frac{3}{4}$ of its neighbors have already updated, and with high probability, for all such points the fraction of neighbors that have updated is more than half. Since all neighbors have updated correctly so far, this means these points will have more correct neighbors than incorrect neighbors no matter what the remaining neighbors do, and so they will update correctly themselves.

□

6.5 Experiments

In our experiments we seek to determine whether our overall algorithm of best-response dynamics combined with active learning is effective at denoising the sensors and learning the target boundary. The experiments were run on synthetic data, and compared active and passive learning (with Support Vector Machines) both pre- and post-denoising.

Synthetic data. The N sensor locations were generated from a uniform distribution over the unit ball in $\mathbb{R}^2$, and the target boundary was fixed as a randomly chosen linear separator through the origin. To simulate noisy scenarios, we corrupted the true sensor labels using two different methods: 1) flipping the sensor labels with probability η and 2) flipping randomly chosen sensor labels and all their neighbors, to create pockets of noise, with η fraction of total sensors corrupted.

Denoising via best-response dynamics. In the denoising phase of the experiments, the sensors applied the basic majority consensus dynamic. That is, each sensor was made to update its label to the majority label of its neighbors within distance

r from its location[1]. We used radius values $r \in \{0.025, 0.05, 0.1, 0.2\}$. Updates of sensor labels were carried out both through simultaneous updates to all the sensors in each iteration (synchronous updates) and updating one randomly chosen sensor in each iteration (asynchronous updates).

Learning the target boundary. After denoising the dataset, we employ the agnostic active learning algorithm of Awasthi et al. [5] described in Section 6.2.2 to decide which sensors to query and obtain a linear separator. We can also extend the algorithm to the case of non-linear boundaries by implementing a kernelized version. Here we compare the resulting error (as measured against the "true" labels given by the target separator) against that obtained by training a SVM on a randomly selected labeled sample of the sensors of the same size as the number of queries used by the active algorithm. We also compare these post-denoising errors with those of the active algorithm and SVM trained on the sensors before denoising. For the active algorithm, we used parameters asymptotically matching those given in Awasthi et al [5] for a uniform distribution. For SVM, we chose for each experiment the regularization parameter that resulted in the best performance.

6.5.1 Results

Here we report the results for $N = 10000$ and $r = 0.1$. Every value reported is an average over 50 independent trials.

Denoising effectiveness. Figure 28 (left side) shows, for various initial noise rates, the fraction of sensors with incorrect labels after applying 100 rounds of synchronous denoising updates. In the random noise case, the final noise rate remains very small even for relatively high levels of initial noise. Pockets of noise appear to be more

[1] We also tested distance-weighted majority and randomized majority dynamics and experimentally observed similar results to those of the basic majority dynamic.

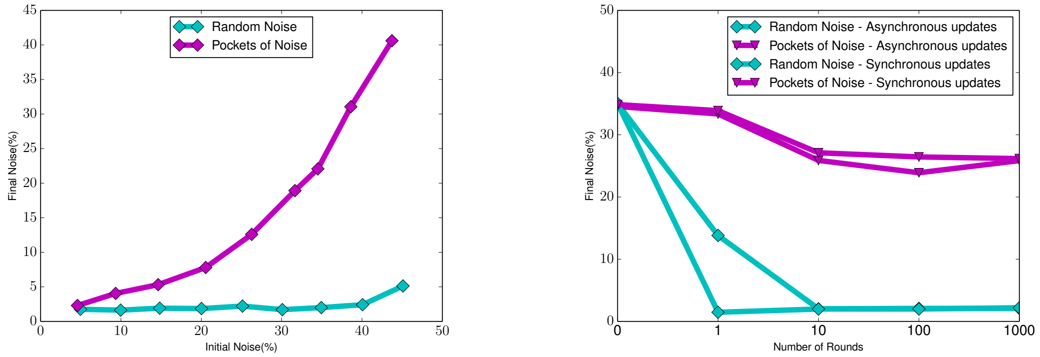


Figure 28: Initial vs. final noise rates for synchronous updates (left) and comparison of synchronous and asynchronous dynamics (right). One synchronous round updates every sensor once simultaneously, while one asynchronous round consists of N random updates.

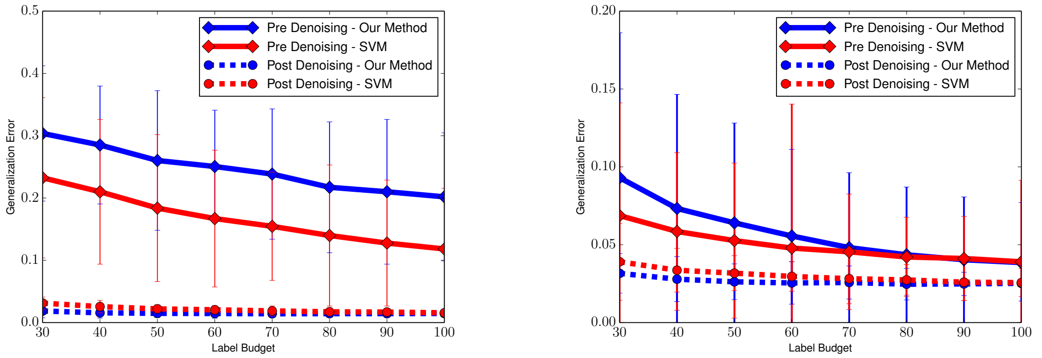


Figure 29: Generalization error of the two learning methods with random noise at rate $\eta = 0.35$ (left) and pockets of noise at rate $\eta = 0.15$ (right).

difficult to denoise. In this case, the final noise rate increases with initial noise rate, but is still nearly always smaller than the initial level of noise.

Synchronous vs. asynchronous updates. To compare synchronous and asynchronous updates we plot the noise rate as a function of the number of rounds of updates in Figure 28 (right side). As our theory suggests, both simultaneous updates and asynchronous updates can quickly converge to a low level of noise in the random noise setting (in fact, convergence happens quickly nearly every time). Neither update strategy achieves the same level of performance in the case of pockets of noise.

Generalization error: pre- vs. post-denoising and active vs. passive. We trained both active and passive learning algorithms on both pre- and post-denoised sensors at various label budgets, and measured the resulting generalization error (determined by the angle between the target and the learned separator). The results of these experiments are shown in Figure 29. Notice that, as expected, denoising helps significantly and on the denoised dataset the active algorithm achieves better generalization error than support vector machines at low label budgets. For example, at a label budget of 30, active learning achieves generalization error approximately 33% lower than the generalization error of SVMs.

CHAPTER VII

CONCLUSIONS

We have given theoretical and empirical evidence that active learning has several uses beyond the traditional one of label complexity improvement over passive learning under limited noise. Our four examples show that active learning can have computational benefits over semi-supervised learning, can be used to discover and exploit margin structure in data, can be used to adapt to a shifting distribution, and can achieve label complexity improvements over passive learning in some scenarios with very high noise. In this chapter, we summarize these findings in more detail and discuss some future directions that have opened up as a result of this work.

In Chapter 3 we proved that the problem of finding a consistent and compatible two-sided disjunction in the semi-supervised setting is NP-hard but that allowing the learner to make active queries allows it to solve this problem efficiently. We also give efficient semi-supervised algorithms for learning two-sided disjunctions, but these are at full strength in a somewhat less general setting than our active algorithm. In addition to this being the first example of provable computational advantages for active learning over semi-supervised learning, our work is also one of the first two discuss active learning in the context of compatibility notions typically associated with semi-supervised learning. This leads to questions of whether computational advantages for active learning can be viewed as a more general phenomenon, rather than specific to our setting of two-sided disjunctions. Are there other settings typically associated with semi-supervised learning where active learning has a computational advantage? Are there more general classes of concepts and compatibility notions where these or similar results will hold? These new directions would be very interesting to explore.

In Chapter 4 we initiated a discussion of the L_qL_p margin spectrum. We proved a generalization guarantee for passive learning that applies to the entire spectrum, generalizing previous results. We showed both theoretically and empirically that situations exist where taking advantage of $L_\infty L_1$ margins leads to better performance than using other margin parameters, complementing prior work showing similar results for other regions of the margin spectrum. We then showed how label queries allow learning algorithms to identify the appropriate margin parameters for the data at hand and then exploit this by shaping the distribution of labeled data to enhance classification accuracy. The idea of using label queries to discover structure in data and then use that knowledge to improve learning is both natural and intriguing and may be possible in many other settings as well.

Chapter 5 introduced ANDA, a novel active nearest neighbors algorithm for domain adaptation. We proved that ANDA has nearly the same generalization guarantee as a passive nearest neighbors classifier but will use drastically fewer labeled examples when information is shared between the source and target distributions. The use of active label queries allows ANDA to automatically adapt its label usage to a local measure of relatedness between the source and target tasks and allows it to maintain statistical consistency even when the source and target tasks are not at all related. These features are not typical of traditional domain adaptation techniques. We gave further evidence of ANDA's adaptation ability by showing that it can correct for data set bias in multi-class image categorization. We hope that this first formal analysis of active domain adaptation will spur the machine learning and learning theory communities to explore this setting more deeply and broadly. One possible direction is to more precisely characterize the convergence rates of different active nearest neighbor methods for domain adaption. On the other hand, it would be very interesting to find other (non-nearest-neighbor) active domain adaptation algorithms that also enjoy strong theoretical guarantees. The power active learning provides for domain adaptation is clear, but our understanding of the benefits and limitations of this setting is far from complete.

In Chapter 6 we gave our final example of a novel use for active learning by introducing a setting in which high-noise active learning is feasible. In this setting, the data is held by many distributed low-power power sensors which can communicate locally with their neighbors. We showed that by using a simple consensus update strategy, the sensors can nearly completely denoise the system so that an active learner running on the resulting data will require exponentially fewer queries than a passive learner would need. The sensor dynamics lead to an approximate equilibrium rather than a true equilibrium, making the analysis different from much of the game theory literature. Avenues for future research include theoretically analyzing alternative denoising schemes and active learning algorithms as well as empirically testing these in real-world sensor networks.

In conclusion, we have shown that the power of active learning extends far beyond its traditional use. In addition to deepening our understanding of active learning itself, this work connects active learning to several other areas of machine learning and game theory. We hope these connections will open new doors for future research and insights into the nature of machine learning.

APPENDIX A

ACTIVE LEARNING FOR DOMAIN ADAPTATION

A.1 Proof of Theorem 10

We adapt the proof (guided exercise) of Theorem 19.5 in [93] to our setting. As is done there, we use the notation $y \sim p$ to denote drawing from a Bernoulli random variable with mean p. We will employ the following lemmas:

Lemma 8 (Lemma 19.6 in [93])**.** *Let $C_1, \ldots, C_r$ be a collection of subsets of some domain set $\mathcal{X}$. Let S be a sequence of m points sampled i.i.d. according to some probability distribution D over $\mathcal{X}$. Then, for every $k \geq 2$,*

$$\mathop{\mathbb{E}}_{S \sim D^m}\left[\sum_{i:|C_i \cap S|<k} \mathbb{P}[C_i]\right] \;\leq\; \frac{2rk}{m}\ .$$

Lemma 9 (Lemma 19.7 in [93])**.** *Let $k \geq 10$ and let $Z_1, \ldots, Z_k$ be independent Bernoulli random variables with $\mathbb{P}[Z_i = 1] = p_i$. Denote $p = \frac{1}{k}\sum_i p_i$ and $p' = \frac{1}{k}\sum_{i=1}^k Z_i$. Then*

$$\mathop{\mathbb{E}}_{Z_1,\ldots,Z_k} \mathop{\mathbb{P}}_{y\sim p}[y \neq \mathbb{1}[p' > 1/2]] \;\leq\; \left(1+\sqrt{\frac{8}{k}}\right) \mathop{\mathbb{P}}_{y\sim p}[y \neq \mathbb{1}[p > 1/2]]\ .$$

Before we prove the theorem, we show the following:

Claim 2 (Ex. 3 of Chapter 19 in [93])**.** *Fix some $p, p' \in [0,1]$ and $y' \in \{0,1\}$. Then*

$$\mathop{\mathbb{P}}_{y\sim p}[y \neq y'] \;\leq\; \mathop{\mathbb{P}}_{y\sim p'}[y \neq y'] + |p - p'|\ .$$

Proof. If $y' = 0$, we have

$$\begin{aligned}
\underset{y\sim p}{\mathbb{P}}[y \neq y'] &= p \\
&= p - p' + p' \\
&= \underset{y\sim p'}{\mathbb{P}}[y \neq y'] + p - p' \\
&\leq \underset{y\sim p'}{\mathbb{P}}[y \neq y'] + |p - p'|.
\end{aligned}$$

If $y' = 1$, we have

$$\begin{aligned}
\underset{y\sim p}{\mathbb{P}}[y \neq y'] &= 1 - p \\
&= 1 - p - p' + p' \\
&= \underset{y\sim p'}{\mathbb{P}}[y \neq y'] - p + p' \\
&\leq \underset{y\sim p'}{\mathbb{P}}[y \neq y'] + |p - p'|.
\end{aligned}$$

□

Proof of Theorem 10. Let h_{ST} denote the output classifier of Algorithm 7, and let $\mathcal{C} = \{C_1, \ldots, C_r\}$ denote an ϵ-cover of the target support $(\mathcal{X}_T, \rho)$. That is, $\bigcup_i C_i = \mathcal{X}_T$ and each C_i has diameter at most ϵ. Without loss of generality, we assume that the C_i are disjoint and for a domain point $x \in \mathcal{X}$ we let $C(x)$ denote the element of $\mathcal{C}$ that contains x. Let $L = T^l \cup S$ denote the (k, k')-NN-cover of T that ANDA uses (that is, the set of labeled points that h_{ST} uses for prediction). We bound its expected loss as follows:

$$\begin{aligned}
\underset{T\sim {D_T}^{m_T}}{\mathbb{E}}[\mathcal{L}_{P_T}(h_{ST})] &= \underset{T\sim {D_T}^{m_T}}{\mathbb{E}}\left[\underset{(x,y)\sim P_T}{\mathbb{P}}[h_{ST}(x) \neq y]\right] \\
&\leq \underset{T\sim {D_T}^{m_T}}{\mathbb{E}}\left[\underset{(x,y)\sim P_T}{\mathbb{P}}[h_{ST}(x) \neq y \wedge \rho(x, x_{k'}(x,T)) > \epsilon] + \underset{(x,y)\sim P_T}{\mathbb{P}}[h_{ST}(x) \neq y \wedge \rho(x, x_{k'}(x,T)) \leq \epsilon]\right] \\
&= \underset{T\sim {D_T}^{m_T}}{\mathbb{E}}\left[\underset{(x,y)\sim P_T}{\mathbb{P}}[\rho(x, x_{k'}(x,T)) > \epsilon]\right] + \underset{T\sim {D_T}^{m_T}}{\mathbb{E}}\left[\underset{(x,y)\sim P_T}{\mathbb{P}}[h_{ST}(x) \neq y \;\wedge\; \rho(x, x_{k'}(x,T)) \leq \epsilon]\right] \\
&= \underset{T\sim {D_T}^{m_T}}{\mathbb{E}}\left[\underset{(x,y)\sim P_T}{\mathbb{P}}[\rho(x, x_{k'}(x,T)) > \epsilon]\right] + \underset{x\sim D_T}{\mathbb{E}}\left[\underset{\substack{y\sim\eta(x)\\ T\sim {D_T}^{m_T}}}{\mathbb{P}}[h_{ST}(x) \neq y \;\wedge\; \rho(x, x_{k'}(x,T)) \leq \epsilon]\right],
\end{aligned}$$

where the last equality holds by Fubini's theorem. Continuing the chain above, we have

$$
\begin{aligned}
&\leq \mathop{\mathbb{E}}_{T\sim D_T{}^{m_T}}\left[\mathop{\mathbb{P}}_{(x,y)\sim P_T}[\rho(x,x_{k'}(x,T))>\epsilon]\right]+\mathop{\mathbb{E}}_{x\sim D_T}\left[\mathop{\mathbb{P}}_{\substack{y\sim\eta(x)\\ T\sim D_T{}^{m_T}}}[h_{ST}(x)\neq y \ \mid \ \rho(x,x_{k'}(x,T))\leq\epsilon]\right]\\
&\leq \mathop{\mathbb{E}}_{T\sim D_T{}^{m_T}}\left[\mathop{\mathbb{P}}_{(x,y)\sim P_T}[|T\cap C(x)|<k']\right]+\mathop{\mathbb{E}}_{x\sim D_T}\left[\mathop{\mathbb{P}}_{\substack{y\sim\eta(x)\\ T\sim D_T{}^{m_T}}}[h_{ST}(x)\neq y \ \mid \ \rho(x,x_{k'}(x,T))\leq\epsilon]\right], \qquad (8)
\end{aligned}
$$

where for the first summand of the last inequality, we used that a point x can only have distance more than ϵ to its k'-th nearest neighbor in T if $C(x)$ is hit less than k' times by T. Lemma 8 implies that this first summand can be upper bounded as

$$
\mathop{\mathbb{E}}_{T\sim D_T{}^{m_T}}\left[\mathop{\mathbb{P}}_{(x,y)\sim P_T}[|T\cap C(x)|<k']\right]\leq\frac{2\,\mathbb{N}_\epsilon(\mathcal{X}_T,\rho)\,k'}{m_T}. \qquad (9)
$$

To bound the second summand, we now first fix a sample T and a point x such that $\rho(x,x_{k'}(x,T))\leq\epsilon$ (and condition on these). Since the set of labeled points $L=T^l\cup S$ used for prediction is an (k,k')-NN-cover of T, Lemma 4 implies that there are at least k labeled points in L at distance at most 3ϵ from x. Let $k(x,L)=\{x_1,\ldots,x_k\}$ be the k nearest neighbors of x in L, let $p_i=\eta(x_i)$ and set $p=\frac{1}{k}\sum_i p_i$. Now we get

$$
\begin{aligned}
\mathop{\mathbb{P}}_{y_1\sim p_1,\ldots y_k\sim p_k,y\sim\eta(x)}[h_{ST}(x)\neq y] &= \mathop{\mathbb{E}}_{y_1\sim p_1,\ldots y_k\sim p_k}\left[\mathop{\mathbb{P}}_{y\sim\eta(x)}[h_{ST}(x)\neq y]\right]\\
&\leq \mathop{\mathbb{E}}_{y_1\sim p_1,\ldots y_k\sim p_k}\left[\mathop{\mathbb{P}}_{y\sim p}[h_{ST}(x)\neq y]\right]+|p-\eta(x)|\\
&\leq \left(1+\sqrt{\frac{8}{k}}\right)\mathop{\mathbb{P}}_{y\sim p}[y\neq\mathbb{1}[p>1/2]]+|p-\eta(x)|,
\end{aligned}
$$

where the first inequality follows from Claim 2 and the second from Lemma 9. We have

$$
\begin{aligned}
\mathop{\mathbb{P}}_{y\sim p}[\mathbb{1}[p>1/2]\neq y] &= \min\{p,1-p\}\\
&\leq \min\{\eta(x),1-\eta(x)\}+|p-\eta(x)|\ .
\end{aligned}
$$

Further, since the regression function η is λ-Lipschitz and $\rho(x_i,x)\leq 3\epsilon$ for all i, we have

$$\begin{aligned}
|p - \eta(x)| &= \left| \left(\frac{1}{k} \sum_i \eta(x_i) \right) - \eta(x) \right| \\
&= \left| \left(\frac{1}{k} \sum_i \eta(x_i) - \eta(x) + \eta(x) \right) - \eta(x) \right| \\
&\leq \left| \left(\frac{1}{k} \sum_i 3\lambda\epsilon + \eta(x) \right) - \eta(x) \right| \\
&= \left| 3\lambda\epsilon + \left(\frac{1}{k} \sum_i \eta(x) \right) - \eta(x) \right| \; = \; 3\lambda\epsilon.
\end{aligned}$$

Thus, we get

$$\begin{aligned}
\underset{y_1 \sim p_1, \ldots y_k \sim p_k, y \sim \eta(x)}{\mathbb{P}} [h_{ST}(x) \neq y] &= \underset{y_1 \sim p_1, \ldots y_k \sim p_k}{\mathbb{E}} \left[\underset{y \sim \eta(x)}{\mathbb{P}} [h_{ST}(x) \neq y] \right] \\
&\leq \left(1 + \sqrt{\frac{8}{k}} \right) \underset{y \sim p}{\mathbb{P}} [y \neq \mathbb{1}[p > 1/2]] + |p - \eta(x)| \\
&\leq \left(1 + \sqrt{\frac{8}{k}} \right) (\min\{\eta(x), 1 - \eta(x)\} + |p - \eta(x)|) + |p - \eta(x)| \\
&\leq \left(1 + \sqrt{\frac{8}{k}} \right) \min\{\eta(x), 1 - \eta(x)\} + 3|p - \eta(x)| \\
&\leq \left(1 + \sqrt{\frac{8}{k}} \right) \min\{\eta(x), 1 - \eta(x)\} + 9\lambda\epsilon.
\end{aligned}$$

Since this holds for all samples T and points x with $\rho(x, x_{k'}(x, T)) \leq \epsilon$, we obtain

$$\begin{aligned}
&\underset{x \sim D_T}{\mathbb{E}} \left[\underset{\substack{y \sim \eta(x) \\ T \sim {D_T}^{m_T}}}{\mathbb{P}} [h_{ST}(x) \neq y \;\mid\; \rho(x, x_{k'}(x, T)) \leq \epsilon] \right] \\
&\qquad \leq \underset{x \sim D_T}{\mathbb{E}} \left[\left(1 + \sqrt{\frac{8}{k}} \right) \min\{\eta(x), 1 - \eta(x)\} + 9\lambda\epsilon \right] \\
&\qquad = \left(1 + \sqrt{\frac{8}{k}} \right) \underset{x \sim D_T}{\mathbb{E}} [\min\{\eta(x), 1 - \eta(x)\}] + 9\lambda\epsilon \\
&\qquad = \left(1 + \sqrt{\frac{8}{k}} \right) \mathcal{L}_T(h_T^*) + 9\lambda\epsilon. \qquad (10)
\end{aligned}$$

Combining Equations (8), (9), and (10) completes the proof. □

REFERENCES

[1] Alimonti, P. and Kann, V., "Some APX-completeness results for cubic graphs," *Theoretical Computer Science*, vol. 237, no. 1–2, pp. 123–134, 2000.

[2] Angluin, D., "Queries and concept learning," *Machine learning*, vol. 2, no. 4, pp. 319–342, 1988.

[3] Arora, S., Ge, R., and Moitra, A., "Learning topic models – going beyond SVD," *CoRR*, 2012.

[4] Atlas, L. E., Cohn, D. A., Ladner, R. E., El-Sharkawi, M. A., II, R. J. M., Aggoune, M. E., and Park, D. C., "Training connectionist networks with queries and selective sampling," in *Advances in Neural Information Processing Systems 2* (Touretzky, D., ed.), pp. 566–573, Morgan-Kaufmann, 1990.

[5] Awasthi, P., Balcan, M. F., and Long, P. M., "The power of localization for efficiently learning linear separators with noise," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pp. 449–458, ACM, 2014.

[6] Bache, K. and Lichman, M., "UCI machine learning repository," 2013.

[7] Balcan, M.-F., Blum, A., and Mansour, Y., "The price of uncertainty," in *EC*, 2009.

[8] Balcan, M.-F., Blum, A., and Mansour, Y., "Circumventing the price of anarchy: Leading dynamics to good behavior.," *SICOMP*, 2014.

[9] Balcan, M. F. and Feldman, V., "Statistical active learning algorithms," in *NIPS*, 2013.

[10] Balcan, M.-F. and Berlind, C., "A new perspective on learning linear separators with large L_qL_p margins," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 68–76, 2014.

[11] Balcan, M.-F., Berlind, C., Ehrlich, S., and Liang, Y., "Efficient Semi-supervised and Active Learning of Disjunctions," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

[12] Balcan, M.-F., Beygelzimer, A., and Langford, J., "Agnostic active learning," in *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 2006.

[13] Balcan, M.-F. and Blum, A., "A discriminative model for semi-supervised learning," *Journal of the ACM*, vol. 57, no. 3, 2010.

[14] Balcan, M.-F., Broder, A., and Zhang, T., "Margin-based active learning," in *COLT*, 2007.

[15] Balcan, M.-F. and Long, P., "Active and passive learning of linear separators under log-concave distributions," in *Conference on Learning Theory*, pp. 288–316, 2013.

[16] Balcan, M.-F. F., Berlind, C., Blum, A., Cohen, E., Patnaik, K., and Song, L., "Active learning and best-response dynamics," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 2222–2230, 2014.

[17] Bartlett, P. L. and Mendelson, S., "Rademacher and gaussian complexities: Risk bounds and structural results," *The Journal of Machine Learning Research*, vol. 3, pp. 463–482, 2003.

[18] Bartlett, P. L. and Shawe-Taylor, J., "Generalization performance of support vector machines and other pattern classifiers," in *Advances in Kernel Methods* (Schölkopf, B., Burges, C. J. C., and Smola, A. J., eds.), pp. 43–54, MIT Press, 1999.

[19] Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F., "Analysis of representations for domain adaptation," in *NIPS*, 2006.

[20] Ben-David, S. and Urner, R., "Domain adaptation-can quantity compensate for quality?," *Ann. Math. Artif. Intell.*, vol. 70, no. 3, pp. 185–202, 2014.

[21] Berlind, C. and Urner, R., "Active nearest neighbors in changing environments," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[22] Beygelzimer, A., Dasgupta, S., and Langford, J., "Importance weighted active learning," in *ICML*, 2009.

[23] Beygelzimer, A., Langford, J., Tong, Z., and Hsu, D. J., "Agnostic active learning without constraints," in *Advances in Neural Information Processing Systems*, pp. 199–207, 2010.

[24] Blum, A. and Balcan, M.-F., "Open problems in efficient semi-supervised PAC learning," in *Proceedings of the 20th Annual Conference on Computational Learning Theory (COLT)*, 2007.

[25] Blum, A. and Mitchell, T., "Combining labeled and unlabeled data with co-training," in *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*, 1998.

[26] Blume, L., "The statistical mechanics of strategic interaction," *Games and Economic Behavior*, vol. 5, pp. 387–424, 1993.

[27] Boucheron, S., Lugosi, G., and Massart, P., *Concentration Inequalities: A Nonasymptotic Theory of Independence.* OUP Oxford, 2013.

[28] Chapelle, O., Schlkopf, B., and Zien, A., *Semi-Supervised Learning.* MIT press, 2006.

[29] Chattopadhyay, R., Fan, W., Davidson, I., Panchanathan, S., and Ye, J., "Joint transfer and batch-mode active learning," in *ICML*, 2013.

[30] Chattopadhyay, R., Wang, Z., Fan, W., Davidson, I., Panchanathan, S., and Ye, J., "Batch mode active sampling based on marginal probability distribution matching," *TKDD*, vol. 7, no. 3, p. 13, 2013.

[31] Chaudhuri, K. and Dasgupta, S., "Rates of convergence for nearest neighbor classification," in *Advances in Neural Information Processing Systems*, pp. 3437–3445, 2014.

[32] Cortes, C., Mansour, Y., and Mohri, M., "Learning bounds for importance weighting," in *NIPS*, 2010.

[33] Cortes, C. and Vapnik, V., "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[34] Cover, T. M. and Hart, P. E., "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[35] Dasgupta, S., "Analysis of a greedy active learning strategy," *Advances in neural information processing systems*, vol. 17, pp. 337–344, 2004.

[36] Dasgupta, S., "Coarse sample complexity bounds for active learning," in *Advances in neural information processing systems*, pp. 235–242, 2005.

[37] Dasgupta, S., "Two faces of active learning," *Theoretical computer science*, vol. 412, no. 19, pp. 1767–1781, 2011.

[38] Dasgupta, S., "Consistency of nearest neighbor classification under selective sampling," in *COLT*, 2012.

[39] Dasgupta, S. and Freund, Y., "Random projection trees and low dimensional manifolds," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 537–546, ACM, 2008.

[40] Dasgupta, S., Littman, M. L., and McAllester, D., "PAC generalization bounds for co-training," in *Advances in Neural Information Processing Systems (NIPS)*, 2001.

[41] Dasgupta, S., Monteleoni, C., and Hsu, D. J., "A general agnostic active learning algorithm," in *Advances in neural information processing systems*, pp. 353–360, 2007.

[42] Dasgupta, S. and Sinha, K., "Randomized partition trees for exact nearest neighbor search," in *COLT*, 2013.

[43] Ellison, G., "Learning, local interaction, and coordination," *Econometrica*, vol. 61, pp. 1047–1071, 1993.

[44] Freund, Y., Seung, H. S., Shamir, E., and Tishby, N., "Selective sampling using the query by committee algorithm," *Machine learning*, vol. 28, no. 2-3, pp. 133–168, 1997.

[45] Friedman, J. H., Bentley, J. L., and Finkel, R. A., "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 3, pp. 209–226, 1977.

[46] Gentile, C., "The Robustness of the p-Norm Algorithms," *Machine Learning*, vol. 53, pp. 265–299, 2003.

[47] Gentile, C., "Personal communication," 2013.

[48] Golovin, D., Krause, A., and Ray, D., "Near-optimal bayesian active learning with noisy observations," in *NIPS*, 2010.

[49] Gonen, A., Sabato, S., and Shalev-Shwartz, S., "Efficient active learning of halfspaces: an aggressive approach," in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, vol. 28 of *JMLR Workshop and Conference Proceedings*, pp. 480–488, 2013.

[50] Grove, A. J., Littlestone, N., and Schuurmans, D., "General Convergence Results for Linear Discriminant Updates," *Machine Learning*, vol. 43, pp. 173–210, 2001.

[51] Haagerup, U., "The best constants in the Khintchine inequality," *Studia Mathematica*, 1982.

[52] Hanneke, S., "Personal communication." 2013.

[53] Hanneke, S., "A bound on the label complexity of agnostic active learning," in *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007.

[54] Hanneke, S., "Teaching dimension and the complexity of active learning," in *Proceedings of the 20th Annual Conference on Computational Learning Theory (COLT)*, 2007.

[55] Hanneke, S., "Rates of convergence in active learning," *The Annals of Statistics*, vol. 39, no. 1, pp. 333–361, 2011.

[56] Hanneke, S., "Activized learning: Transforming passive to active with improved label complexity," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1469–1587, 2012.

[57] Hanneke, S., "Theory of disagreement-based active learning," *Foundations and Trends in Machine Learning*, vol. 7, no. 2-3, pp. 131–309, 2014.

[58] Hoeffding, W., "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, pp. 13–30, March 1963.

[59] Huang, T., Agarwal, A., Hsu, D. J., Langford, J., and Schapire, R. E., "Efficient and parsimonious agnostic active learning," *CoRR*, vol. abs/1506.08669, 2015.

[60] Itai, A. and Benedek, G. M., "Learnability with respect to fixed distributions," *Theoretical Computer Science*, vol. 86, no. 2, pp. 377–389, 1991.

[61] Jain, P., Vijayanarasimhan, S., and Grauman, K., "Hashing hyperplane queries to near points with applications to large-scale active learning," in *Advances in Neural Information Processing Systems*, pp. 928–936, 2010.

[62] Joachims, T., "Transductive inference for text classification using support vector machines," in *Proceedings of the 16th International Conference on Machine Learning (ICML)*, 1999.

[63] Kääriäinen, M., "Generalization error bounds using unlabeled data," in *Proceedings of the 18th Annual Conference on Computational Learning Theory (COLT)*, 2005.

[64] Kääriäinen, M., "Active learning in the non-realizable case," in *Algorithmic Learning Theory*, pp. 63–77, Springer, 2006.

[65] Kakade, S. M., Sridharan, K., and Tewari, A., "On the Complexity of Linear Prediction: Risk Bounds, Margin Bounds, and Regularization," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2009.

[66] Kearns, M. J., Schapire, R. E., and Sellie, L. M., "Toward efficient agnostic learning," *Machine Learning*, vol. 17, no. 2-3, pp. 115–141, 1994.

[67] Kempe, D., Kleinberg, J., and Tardos, E., "Maximizing the spread of influence through a social network," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pp. 137–146, ACM, 2003.

[68] Kloft, M. and Blanchard, G., "On the Convergence Rate of ℓ_p-Norm Multiple Kernel Learning," *Journal of Machine Learning Research*, vol. 13, pp. 2465–2501, 2012.

[69] Koltchinskii, V., "Rademacher complexities and bounding the excess risk in active learning," *The Journal of Machine Learning Research*, vol. 11, pp. 2457–2485, 2010.

[70] Koltchinskii, V. and Panchenko, D., “Empirical margin distributions and bounding the generalization error of combined classifiers,” *Annals of Statistics*, pp. 1–50, 2002.

[71] Kpotufe, S., “k-NN regression adapts to local intrinsic dimension,” in *NIPS*, 2011.

[72] Kulkarni, S. R. and Posner, S. E., “Rates of convergence of nearest neighbor estimation under arbitrary sampling,” *IEEE Transactions on Information Theory*, vol. 41, no. 4, pp. 1028–1039, 1995.

[73] Littlestone, N., “Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm,” *Machine Learning*, vol. 2, no. 4, pp. 285–318, 1988.

[74] MacKay, D., “Information-based objective functions for active data selection,” *Neural computation*, vol. 4, no. 4, pp. 590–604, 1992.

[75] Mansour, Y., Mohri, M., and Rostamizadeh, A., “Domain adaptation: Learning bounds and algorithms,” in *COLT*, 2009.

[76] Maurer, A. and Pontil, M., “Structured Sparsity and Generalization,” *Journal of Machine Learning Research*, vol. 13, pp. 671–690, 2012.

[77] Mitchell, T. M., “Version spaces: an approach to concept learning.,” tech. rep., Dept. of Computer Science, Stanford University, 1978.

[78] Morris, S., “Contagion,” *The Review of Economic Studies*, vol. 67, no. 1, pp. 57–78, 2000.

[79] Neyshabur, B., Tomioka, R., and Srebro, N., “Norm-based capacity control in neural networks,” *CoRR*, vol. abs/1503.00036, 2015.

[80] Pan, S. J. and Yang, Q., “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, October 2010.

[81] Rajagopalan, S. and Vazirani, V. V., “Primal-dual RNC approximation algorithms for (multi)-set (multi)-cover and covering integer programs,” in *FOCS*, 1993.

[82] Ram, P. and Gray, A. G., “Which space partitioning tree to use for search?,” in *NIPS*, 2013.

[83] Ram, P., Lee, D., and Gray, A. G., “Nearest-neighbor search on a time budget via max-margin trees,” in *SDM*, 2012.

[84] Rigollet, P., “Generalized error bounds in semi-supervised classification under the cluster assumption,” *Journal of Machine Learning Research*, vol. 8, 2007.

[85] Rosenberg, D. S. and Bartlett, P. L., "The rademacher complexity of co-regularized kernel classes," in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

[86] Rosenblatt, F., "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386–407, 1958.

[87] Roy, N. and McCallum, A., "Toward optimal active learning through monte carlo estimation of error reduction," *ICML, Williamstown*, 2001.

[88] Sabato, S. and Munos, R., "Active regression by stratification," in *Advances in Neural Information Processing Systems 27* (Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., eds.), pp. 469–477, Curran Associates, Inc., 2014.

[89] Saha, A., Rai, P., III, H. D., Venkatasubramanian, S., and DuVall, S. L., "Active supervised domain adaptation," in *ECML/PKDD*, 2011.

[90] Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S., "Boosting the margin: A new explanation for the effectiveness of voting methods," *Annals of statistics*, pp. 1651–1686, 1998.

[91] Servedio, R. A., "PAC Analogues of Perceptron and Winnow via Boosting the Margin," in *Proceedings of the Conference on Learning Theory (COLT)*, pp. 130–151, 2000.

[92] Settles, B., "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, pp. 55–66, 2010.

[93] Shalev-Shwartz, S. and Ben-David, S., *Understanding Machine Learning.* Cambridge University Press, 2014.

[94] Shawe-Taylor, J. and Cristianini, N., "On the Generalisation of Soft Margin Algorithms," *IEEE Transactions on Information Theory*, vol. 48, 2000.

[95] Shi, Y. and Sha, F., "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," in *ICML*, 2012.

[96] Shimodaira, H., "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of statistical planning and inference*, vol. 90, no. 2, pp. 227–244, 2000.

[97] Stone, C. J., "Consistent nonparametric regression," *The Annals of Statistics*, vol. 5, pp. 595–620, 07 1977.

[98] Sugiyama, M. and Mueller, K., "Generalization error estimation under covariate shift," in *Workshop on Information-Based Induction Sciences*, 2005.

[99] Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M., "Direct importance estimation for covariate shift adaptation," *Annals of the Institute of Statistical Mathematics*, vol. 60, no. 4, pp. 699–746, 2008.

[100] Tommasi, T., Patricia, N., Caputo, B., and Tuytelaars, T., "A Deeper Look at Dataset Bias," *ArXiv e-prints*, May 2015.

[101] Tommasi, T. and Tuytelaars, T., "A testbed for cross-dataset analysis," in *TASK-CV Workshop at ECCV*, 2014.

[102] Tong, S. and Koller, D., "Active learning for structure in bayesian networks," in *International joint conference on artificial intelligence*, vol. 17, pp. 863–869, LAWRENCE ERLBAUM ASSOCIATES LTD, 2001.

[103] Tong, S. and Koller, D., "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[104] Valiant, L. G., "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[105] Vapnik, V. N. and Chervonenkis, A. J., "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability & Its Applications*, vol. 16, no. 2, pp. 264–280, 1971.

[106] Vazirani, V., *Approximation Algorithms*. Springer, 2001.

[107] Verma, N., Kpotufe, S., and Dasgupta, S., "Which spatial partition trees are adaptive to intrinsic dimension?," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pp. 565–574, AUAI Press, 2009.

[108] Yang, L., *Mathematical Theories of Interaction with Oracles*. PhD thesis, CMU Dept. of Machine Learning, 2013.

[109] Zhang, C. and Chaudhuri, K., "Beyond disagreement-based agnostic active learning," *CoRR*, vol. abs/1407.2657, 2014.

[110] Zhang, T., "On the dual formulation of regularized linear systems with convex risks," *Machine Learning*, vol. 46, pp. 91–129, 2002.

[111] Zhao, Y. and Teng, S.-H., "Combinatorial and spectral aspects of nearest neighbor graphs in doubling dimensional and nearly-euclidean spaces," in *TAMC*, 2007.

[112] Zhu, X., "Semi-supervised learning," in *Encyclopedia of Machine Learning* (Sammut, C. and Webb, G. I., eds.), Springer, 2010.

[113] Zhu, X., Ghahramani, Z., and Lafferty, J., "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 2003.