Active Perspectives on Computational Learning and Testing

Thesis Proposal

Liu Yang

Machine Learning Department
Carnegie Mellon University
liuy@cs.cmu.edu

Abstract. We investigate the potential benefits of interaction in a variety of classic learning theory topic areas, as well as the potential benefits that active learning can itself derive from these areas. Specifically, in the context of Bayesian learning, we find that access to a prior over target concepts can greatly benefit active learning algorithms, which are thereby enabled to significantly improve over their prior-dependent passive learning counterparts. We further show that these same benefits can be realized by a form of transfer learning, thereby eliminating the need for direct access to the target's prior distribution. Finally, we describe several future research directions to be pursued in the near future: namely, the potential benefits of the pool-based active learning protocol in the context of property testing, using a specialized type of query to learn DNF formulae, and the ability of active learning to adapt to a drifting distribution.

Table of Contents

Active Perspectives on Computational Learning and Testing 1
Liu Yang
1 Overview 2
2 Bayesian Active Learning Using Arbitrary Binary Valued Queries 3
2.1 Definitions and Notation 4
2.2 Main Result 6
3 The Sample Complexity of Self-Verifying Bayesian Active Learning 6
3.1 Introduction and Background 7
3.2 Definitions and Preliminaries 8
3.3 Main Result 9
4 Active Transfer Learning : Identifiability of Priors from Bounded Sample Sizes with Applications to Transfer Learning 9
4.1 Introduction 10
4.2 Definitions and Related Work 11
4.3 Relation to Existing Theoretical Work on Transfer Learning 12
4.4 Estimating the Prior 13
4.5 Identifiability from d Points 14
4.6 Transfer Learning 14
4.7 Application to Self-Verifying Active Learning 15
5 Future Research Problems 15
5.1 Active Property Testing 15
5.2 Efficient Learning with General Queries 16
5.3 Online Active Learning under Distribution Drift 17
6 Tentative Timeline for Completing the Work 18

1 Overview

In this work, we explore a variety of learning theory topics, approached in an interactive learning protocol.

To begin, we explore the classic active learning setting, in a Bayesian context. We are specifically interested in the question of what benefits we can derive from having access to a prior distribution for the target concept.

We begin to address this question by first asking the fundamental question of what benefits we may gain from having access to this prior, given that we have the ability to ask *arbitrary* binary valued queries. In the non-Bayesian setting, the query complexity in this setting is characterized by the logarithm of the size of a maximal ϵ-packing of the concept space. In contrast, in the Bayesian setting, we find that the query complexity is characterized by the *entropy* of a partition induced by the maximal ϵ-packing. Since this entropy is never larger than the logarithm of the size of the ϵ-packing, and is often significantly smaller, this reflects that there are some benefits to having access to the target's prior distribution. Additionally, as label requests are a special case of binary valued queries, these results also have lower-bound implications for Bayesian active learning from label requests.

Continuing this investigation into the benefits of access to the target's prior, we investigate the classic active learning setting, where we are restricted to making label requests queries among examples in a large pool of unlabeled examples. In this context, we again find benefits from having access to the prior. Specifically, in the context of active learning without access to the prior, it is known that it is always possible to learn to expected error rate ϵ using an expected number of queries $o(1/\epsilon)$, when the target resides in a known VC class. However, if we also require the active learning algorithm to be *self-verifying*, in the sense that it itself adaptively decides how many queries to makes and then halts, while maintaining the guarantee that the classifier it produces upon termination has expected error rate ϵ, then the expected number of queries the algorithm must make can be as large as $\Omega(1/\epsilon)$ in many cases, which is no better than passive learning. In contrast, in the Bayesian setting, where the algorithm has direct access to the target's prior distribution, we find that there even exist self-verifying algorithms capable of guaranteeing expected error rate ϵ after an expected number of queries only $o(1/\epsilon)$. Thus, in this context, the advantage of having access to the prior is to allow for self-verification at a label complexity that would not have been possible without access to the prior.

As the above results indicate, we find that having access to the target's prior distribution is often advantageous. However, we have not mentioned how one obtains access to this distribution in practice. The usual answer to this question is that we learn about the target's prior distribution *from experience*. To formalize this intuition, we study the general problem of *transfer learning*, where we are faced with a sequence of learning problems, each of which has its target concept sampled independently according to a fixed prior distribution. We construct a method that solves these tasks sequentially, and as the number of solved tasks grows large, constructs an estimator of the targets' prior distribution which converges to the true prior. In particular, we show that if we use this estimator in a certain way combined with the prior-dependent methods mentioned above, we are capable of achieving the same long-run average label complexity for prior-dependent self-verifying active learning as would be achievable by learning algorithms having direct access to the targets' prior distribution.

In addition to exploring the classic active *learning* settings, we additionally explore a number of other topics that involve interaction.

In the context of property testing, we investigate the question of whether label request queries, of the type used in active learning, can provide some benefit in terms of the label complexity of testing, compared to passive testing. We find that, in a number of cases, the label complexity of testing with label request queries has a significantly better dependence on the dimensionality compared to passive testing. In some cases, it may even become independent of the dimension, where passive is not. These benefits are often not as dramatic as available for testing with membership queries, but in many practical contexts, it is more realistic to expect label request queries (for unlabeled examples in a reaonably-sized pool) to be answerable by an expert, rather than membership queries which may often appear to an expert to be incoherent or to have ambiguous labels.

We also discuss a general interactive learning topic, in which we approach learning problems that are computationally hard for passive learning, but which may be made computationally tractable via specially designed queries. For instance, in the context of learning DNF boolean functions, we explore a type of target-dependent similarity measure, which asks whether a pair of examples from a given pool both satisfy the same term in a parsimonious DNF representation of the target function. We find that this type of query enables efficient learning, and construct an efficient algorithm to achieve this.

Finally, we discuss active learning in a stream-based online learning model, where the distribution of data may drift over time. Specifically, on each round, the algorithm is presented with an independent unlabeled example and asked to make a prediction. It may then optionally request the label. We study the total number of mistakes and number of label requests as a function of the number of rounds. We are specifically interested in conditions under which the number of mistakes and number of queries are sublinear in the number of rounds. We further study this problem when the distribution generating the data is allowed to change from round to round. We find that as long as the distributions reside in a totally bounded space, under certain other conditions on the concept space and space of distributions, we are able to maintain this sublinearity. This is true in both the realizable case, and certain noisy settings. We further pursue an explicit minimax analysis of the number of queries and number of mistakes under conditions on the concept space, space of distributions, and noise conditions.

2 Bayesian Active Learning Using Arbitrary Binary Valued Queries

In (Yang et al., 2010), we explore a general Bayesian active learning setting, in which the learner can ask arbitrary yes/no questions. We derive upper and lower bounds on the expected number of queries required to achieve a specified expected risk.

In this work, we study the fundamental complexity of Bayesian active learning by examining the basic problem of learning from binary-valued queries. We are particularly interested in identifying a key quantity that characterizes the number of queries required to learn to a given accuracy, given knowledge of the prior distribution from which the target is sampled. This topic is interesting both in itself, and also as a general setting in which to derive lower bounds, which apply broadly to any active learning scenario in which binary-valued queries are employed, such as the popular setting of active learning with label requests (membership queries). The analysis of the Bayesian variant of this setting is important for at least two reasons: first, for practical reasons, as minimax analyses tend to emphasize scenarios much more difficult to learn from than what the world often offers us, so that the smoothed or average-case analysis offered by a Bayesian setting can often be an informative alternative, and second, for philosophical reasons, owing to the decision-theoretic interpretation of rational inference, which is typically formulated in a Bayesian setting.

There is much related work on active learning with binary-valued queries. However, perhaps the most relevant for us is the result of (Kulkarni et al., 1993). In this classic work, they allow a learning algorithm to ask *any* question with a yes/no answer, and derive a precise characterization of the number of these binary-valued queries necessary and sufficient for learning a target classifier to a prescribed accuracy, in a PAC-like framework. In particular, they find this quantity is essentially characterized by $\log \mathcal{M}(\epsilon)$, where $1-\epsilon$ is the desired accuracy, and $\mathcal{M}(\epsilon)$ is the size of a maximal ϵ-packing of the concept space.

In addition to being quite interesting in their own right, these results have played a significant role in the recent developments in active learning with "label request" queries for binary classification (Hanneke, 2007b; Hanneke, 2007a; Dasgupta, 2005). Specifically, since label requests can be viewed as a type of binary-valued query, the number of label requests necessary for learning is naturally lower bounded by the number of arbitrary binary-valued queries necessary for learning. We therefore always expect to see some term relating to $\log \mathcal{M}(\epsilon)$ in our sample complexity bounds for active learning with label requests (though this factor is typically represented by its upper bound: $\propto VC(\mathbb{C}) \log(1/\epsilon)$, where $VC(\mathbb{C})$ is the VC dimension).

Also related is a certain thread of the literature on sample complexity bounds for Bayesian learning. In particular, (Haussler et al., 1994a) study the passive learning problem in a Bayesian setting, and study the effect of the information made available via access to the prior. In many cases, the learning problem is made significantly easier than the worst-case scenarios of the PAC model. In particular, (building from the work of (Haussler et al., 1994b)) they find that $VC(\mathbb{C})/\epsilon$ random labeled examples are sufficient to achieve expected error rate at most ϵ using the Bayes classifier.

Allowing somewhat more general types of queries than (Haussler et al., 1994a), a paper by (Freund et al., 1997; Seung et al., 1992) studied an algorithm known as Query by Committee (QBC). Specifically, QBC is allowed to sequentially decide which points to select, observing each response before selecting the next data point to observe. They found this additional flexibility can sometimes pay off significantly, reducing the expected number of queries needed exponentially to only $O(\log(1/\epsilon))$. However, these results only seem to apply to a very narrow family of problems, where a certain expected information gain quantity is lower bounded by a constant, a situation which seems fairly uncommon among the types of learning problems we are typically most interested in (informative priors, or clustered data). Thus, to our knowledge, the general questions, such as how much advantage we actually get from having access to the prior π, and what fundamental quantities describe the intrinsic complexity of the learning problem, remain virtually untouched in the published literature.

The "label request" query discussed in these Bayesian analyses represents a type of binary-valued query, though quite restricted compared to the powerful queries analyzed in the present work. As a first step toward a more complete understanding of the Bayesian active learning problem, we propose to return to the basic question of how many binary-valued queries are necessary and sufficient in general; but unlike the (Kulkarni et al., 1993) analysis, we adopt the Bayesian perspective of (Haussler et al., 1994a) and (Freund et al., 1997), so that the algorithms in question will directly depend on the prior π. In fact, we investigate the problem in a somewhat more general form, where reference to the underlying data distribution is replaced by direct reference to the induced pseudo-metric between elements of the concept space. As we point out below, this general problem has deep connections to many problems commonly studied in information theory (e.g., the analysis of lossy compression); for instance, one might view the well-known asymptotic results of rate distortion theory as a massively multitask variant of this problem. However, to our knowledge, the basic question of the number of binary queries necessary to approximate a single random target h^* to a given accuracy, given access to the distribution π of h^*, has not previously been addressed in generality.

Below, we are able to derive upper and lower bounds on the query complexity based on a natural analogue of the bounds of (Kulkarni et al., 1993). Specifically, we find that in this Bayesian setting, under an assumption of bounded doubling dimension, the query complexity is controlled by the entropy of a partition induced by a maximal ϵ-packing (specifically, the natural Voronoi partition); in particular, the worst-case value of this entropy is the $\log \mathcal{M}(\epsilon)$ bound of (Kulkarni et al., 1993), which represents a uniform prior over the regions of the partition. The upper bound is straightforward to derive, but nice to have; but our main contribution is the lower bound, the proof of which is somewhat more involved.

The rest of this section is organized as follows. In Section 2.1, we introduce a few important quantities used in the statement of the main theorem. Following this, Section 2.2 contains a statement of our main result, along with some explanation.

2.1 Definitions and Notation

We will formalize our discussion in somewhat more abstract terms.

Formally, throughout this discussion, we will suppose $\mathbb{C}^*$ is an arbitrary (nonempty) collection of objects, equipped with a separable pseudo-metric $\rho : \mathbb{C}^* \times \mathbb{C}^* \to [0, \infty)$. [1] We suppose $\mathbb{C}^*$ is equipped with its Borel σ-algebra induced by ρ. There is additionally a (nonempty, measurable) set $\mathbb{C} \subseteq \mathbb{C}^*$, and we denote by $\bar{\rho} = \sup_{h_1, h_2 \in \mathbb{C}} \rho(h_1, h_2)$. Finally, there is a probability measure π with $\pi(\mathbb{C}) = 1$, known as the "prior," and a $\mathbb{C}$-valued random variable h^* with distribution π, known as the "target." As the prior is essentially arbitrary, the results below will hold for *any* prior π.

As an example, in the special case of the binary classifier learning problem studied by (Haussler et al., 1994a) and (Freund et al., 1997), $\mathbb{C}^*$ is the set of all measurable classifiers $h : \mathcal{X} \to \{-1, +1\}$, $\mathbb{C}$ is the "concept space," h^* is the "target function," and $\rho(h_1, h_2) = \mathbb{P}_{X \sim \mathcal{D}}(h_1(X) \neq h_2(X))$, where $\mathcal{D}$ is the distribution of the (unlabeled) data; in particular, $\rho(h, h^*) = er(h)$ is the "error rate" of h.

To discuss the fundamental limits of learning with binary-valued queries, we define the quantity $\mathrm{QueryComplexity}(\epsilon)$, for $\epsilon > 0$, as the minimum possible expected number of binary queries for any learning algorithm guaranteed to return $\hat{h}$ with $\mathbb{E}[\rho(\hat{h}, h^*)]$ $\leq \epsilon$, where the only random variable in the expectation is $h^* \sim \pi$ (and $\hat{h}$, which is itself determined by h^* and the sequence of queries). For simplicity, we restrict ourselves to deterministic algorithms in this section, so that the only source of randomness is h^*.

Alternatively, there is a particularly simple interpretation of the notion of an algorithm based on arbitrary binary-valued queries, which leads to an equivalent definition of $\mathrm{QueryComplexity}(\epsilon)$: namely, a prefix-free code. That is, any deterministic algorithm that asks a sequence of yes/no questions before terminating and returning some $\hat{h} \in \mathbb{C}^*$ can be thought of as a binary decision tree (no = left, yes = right), with the return $\hat{h}$ values stored in the leaf nodes. Transforming each root-to-leaf path in the decision tree into a codeword (left = 0, right = 1), we see that the algorithm corresponds to a prefix-free binary code. Conversely, given any prefix-free binary code, we can construct an algorithm based on sequentially asking queries of the form "what is the first bit in the codeword $C(h^*)$ for h^*?", "what is the second bit in the codeword $C(h^*)$ for h^*?", etc., until we obtain a complete codeword, at which point we return the value that codeword decodes to. From this perspective, we can state an equivalent definition of $\mathrm{QueryComplexity}(\epsilon)$ in the language of lossy codes.

Formally, a *code* is a pair of (measurable) functions (C, D). The *encoder*, C, maps any element $h \in \mathbb{C}$ to a binary sequence $C(h) \in \bigcup_{q=0}^{\infty} \{0, 1\}^q$ (the *codeword*). The *decoder*, D, maps any element $c \in \bigcup_{q=0}^{\infty} \{0, 1\}^q$ to an element $D(c) \in \mathbb{C}^*$. For any $q \in \{0, 1, \ldots\}$ and $c \in \{0, 1\}^q$, let $|c| = q$ denote the *length* of c. A *prefix-free* code is any code (C, D) such that no $h_1, h_2 \in \mathbb{C}$ have $c^{(1)} = C(h_1)$ and $c^{(2)} = C(h_2)$ with $c^{(1)} \neq c^{(2)}$ but $\forall i \leq |c^{(1)}|$, $c_i^{(2)} = c_i^{(1)}$: that is, no codeword is a prefix of another (longer) codeword.

Here, we consider a setting where the code (C, D) may be *lossy*, in the sense that for some values of $h \in \mathbb{C}$, $\rho(D(C(h)), h) > 0$. Our objective is to design the code to have small expected loss (in the ρ sense), while maintaining as small of an expected codeword length as possible, where expectations are over the target h^*, which is also the element of $\mathbb{C}$ we encode. The following defines the optimal such length.

Definition 1. *For any $\epsilon > 0$, define the* query complexity *as*

$$\begin{aligned}&\mathrm{QueryComplexity}(\epsilon)\\&= \inf\left\{\mathbb{E}\Big[|C(h^*)|\Big] : (C, D) \textit{ is a prefix-free code with } \mathbb{E}\left[\rho\Big(D(C(h^*)), h^*\Big)\right] \leq \epsilon\right\},\end{aligned}$$

where the random variable in both expectations is $h^* \sim \pi$.

Recalling the equivalence between prefix-free binary codes and deterministic learning algorithms making arbitrary binary-valued queries, note that this definition is equivalent to the earlier definition.

Returning to the specialized setting of binary classification for a moment, we see that this corresponds to the minimum possible expected number of binary queries for a learning algorithm guaranteed to have expected error rate at most ϵ.

Given this coding perspective, we should not be surprised to see an entropy quantity appear in the results of the next section. Specifically, define the following quantities.

Definition 2. *For any $\epsilon > 0$, define $\mathcal{Y}(\epsilon) \subseteq \mathbb{C}$ as a maximal ϵ-packing of $\mathbb{C}$. That is, $\forall h_1, h_2 \in \mathcal{Y}(\epsilon)$, $\rho(h_1, h_2) \geq \epsilon$, and $\forall h \in \mathbb{C} \setminus \mathcal{Y}(\epsilon)$, the set $\mathcal{Y}(\epsilon) \cup \{h\}$ does not satisfy this property.*

[1] The set $\mathbb{C}^*$ will not play any significant role in the analysis, except to allow for improper learning scenarios to be a special case of our setting.

For our purposes, if multiple maximal ϵ-packings are possible, we can choose to define $\mathcal{Y}(\epsilon)$ arbitrarily from among these; the results below hold for any such choice. Recall that any maximal ϵ-packing of $\mathbb{C}$ is also an ϵ-cover of $\mathbb{C}$, since otherwise we would be able to add to $\mathcal{Y}(\epsilon)$ the $h \in \mathbb{C}$ that escapes the cover.

Next we define a complexity measure, a type of entropy, which serves as our primary quantity of interest in the analysis of $\mathrm{QueryComplexity}(\epsilon)$. It is specified in terms of a partition induced by $\mathcal{Y}(\epsilon)$, defined as follows.

Definition 3. *For any $\epsilon > 0$, define*

$$\mathcal{P}(\epsilon) = \left\{ \left\{ h \in \mathbb{C} : f = \operatorname*{argmin}_{g \in \mathcal{Y}(\epsilon)} \rho(h, g) \right\} : f \in \mathcal{Y}(\epsilon) \right\},$$

where we break ties in the argmin *arbitrarily but consistently (e.g., based on a predefined preference ordering of $\mathcal{Y}(\epsilon)$). If the* argmin *is not defined (i.e., the* min *is not realized), take any $f \in \mathcal{Y}(\epsilon)$ with $\rho(f, h) \leq \epsilon$ (one must exist by maximality of $\mathcal{Y}(\epsilon)$).*

Definition 4. *For any finite (or countable) partition $\mathcal{S}$ of $\mathbb{C}$ into measurable regions (subsets), define the* entropy *of $\mathcal{S}$*

$$\mathcal{H}(\mathcal{S}) = -\sum_{S \in \mathcal{S}} \pi(S) \log_2 \pi(S).$$

In particular, we will be interested in the quantity $\mathcal{H}(\mathcal{P}(\epsilon))$ in the analysis below.

Finally, we will require a notion of dimensionality for the pseudo-metric ρ. For this, we adopt the well-known *doubling dimension* (Gupta et al., 2003).

Definition 5. *Define the* doubling dimension *d as the smallest value d such that, for any $h \in \mathbb{C}$, and any $\epsilon > 0$, the size of the minimal $\epsilon/2$-cover of the ϵ-radius ball around h is at most 2^d.*

That is, for any $h \in \mathbb{C}$ and $\epsilon > 0$, there exists a set $\{h_i\}_{i=1}^{2^d}$ of 2^d elements of $\mathbb{C}$ such that

$$\{h' \in \mathbb{C} : \rho(h', h) \leq \epsilon\} \subseteq \bigcup_{i=1}^{2^d} \{h' \in \mathbb{C} : \rho(h', h_i) \leq \epsilon/2\}.$$

Note that, as defined here, d is a constant (i.e., has no dependence on h or ϵ). See (Bshouty et al., 2009) for a discussion of the doubling dimension of spaces $\mathbb{C}$ of binary classifiers, in the context of learning theory.

2.2 Main Result

Our main result can be summarized as follows. Note that, since we took the prior to be *arbitrary* in the above definitions, this result holds for *any* prior π.

Theorem 1. *If $d < \infty$ and $\bar{\rho} < \infty$, then there is a constant $c = O(d)$ such that $\forall \epsilon \in (0, \bar{\rho}/2)$,*

$$\mathcal{H}\left(\mathcal{P}\left(\epsilon \log_2(\bar{\rho}/\epsilon)\right)\right) - c \leq \mathrm{QueryComplexity}(\epsilon) \leq \mathcal{H}\left(\mathcal{P}\left(\epsilon\right)\right) + 1.$$

Due to the deep connections of this problem to information theory, it should not be surprising that entropy terms play a key role in this result. Indeed, this type of entropy seems to give a good characterization of the asymptotic behavior of the query complexity in this setting. We should expect the upper bound to be tight when the regions in $\mathcal{P}(\epsilon)$ are point-wise well-separated. However, it may be looser when this is not the case, for reasons discussed in the next section.

3 The Sample Complexity of Self-Verifying Bayesian Active Learning

In (Yang et al., 2011), we prove that access to a prior distribution over target functions can dramatically improve the sample complexity of self-terminating active learning algorithms, so that it is always better than the known results for prior-dependent passive learning. In particular, this is in stark contrast to the analysis of prior-independent algorithms, where there are simple known learning problems for which no self-terminating algorithm can provide this guarantee for all priors.

3.1 Introduction and Background

Active learning is a powerful form of supervised machine learning characterized by interaction between the learning algorithm and supervisor during the learning process. In this work, we consider a variant known as *pool-based* active learning, in which a learning algorithm is given access to a (typically very large) collection of unlabeled examples, and is able to select any of those examples, request the supervisor to label it (in agreement with the target concept), then after receiving the label, selects another example from the pool, etc. This sequential label-requesting process continues until some halting criterion is reached, at which point the algorithm outputs a function, and the objective is for this function to closely approximate the (unknown) target concept in the future. The primary motivation behind pool-based active learning is that, often, unlabeled examples are inexpensive and available in abundance, while annotating those examples can be costly or time-consuming; as such, we often wish to select only the informative examples to be labeled, thus reducing information-redundancy to some extent, compared to the baseline of selecting the examples to be labeled uniformly at random from the pool (passive learning).

There has recently been an explosion of fascinating theoretical results on the advantages of this type of active learning, compared to passive learning, in terms of the number of labels required to obtain a prescribed accuracy (called the *sample complexity*): e.g., (Freund et al., 1997; Dasgupta, 2004; Dasgupta et al., 2009; Dasgupta, 2005; Hanneke, 2007b; Balcan et al., 2010; Balcan et al., 2009; Wang, 2009; Kääriäinen, 2006; Hanneke, 2007a; Dasgupta et al., 2007; Friedman, 2009; Castro & Nowak, 2008; Nowak, 2008; Balcan et al., 2007; Hanneke, 2011; Koltchinskii, 2010; Hanneke, 2009; Beygelzimer et al., 2009). In particular, (Balcan et al., 2010) show that in noise-free binary classifier learning, for any passive learning algorithm for a concept space of finite VC dimension, there exists an active learning algorithm with asymptotically much smaller sample complexity for any nontrivial target concept. In later work, (Hanneke, 2009) strengthens this result by removing a certain strong dependence on the distribution of the data in the learning algorithm. Thus, it appears there are profound advantages to active learning compared to passive learning.

However, the ability to rapidly converge to a good classifier using only a small number of labels is only one desirable quality of a machine learning method, and there are other qualities that may also be important in certain scenarios. In particular, the ability to *verify* the performance of a learning method is often a crucial part of machine learning applications, as (among other things) it helps us determine whether we have enough data to achieve a desired level of accuracy with the given method. In passive learning, one common practice for this verification is to hold out a random sample of labeled examples as a *validation sample* to evaluate the trained classifier (e.g., to determine when training is complete). It turns out this technique is not feasible in active learning, since in order to be really useful as an indicator of whether we have seen enough labels to guarantee the desired accuracy, the number of labeled examples in the random validation sample would need to be much larger than the number of labels requested by the active learning algorithm itself, thus (to some extent) canceling the savings obtained by performing active rather than passive learning. Another common practice in passive learning is to examine the training error rate of the returned classifier, which can serve as a reasonable indicator of performance (after adjusting for model complexity). However, again this measure of performance is not necessarily reasonable for active learning, since the set of examples the algorithm requests the labels of is typically distributed very differently from the test examples the classifier will be applied to after training.

This reasoning indicates that performance verification is (at best) a far more subtle issue in active learning than in passive learning. Indeed, (Balcan et al., 2010) note that although the number of labels required to achieve good accuracy is significantly smaller than passive learning, it is often the case that the number of labels required to *verify* that the accuracy is good is not significantly improved. In particular, this phenomenon can dramatically increase the sample complexity of active learning algorithms that adaptively determine how many labels to request before terminating. In short, if we require the algorithm both to *learn* an accurate concept and to *know* that its concept is accurate, then the number of labels required by active learning is often not significantly smaller than the number required by passive learning.

We should note, however, that the above results were proven for a learning scenario in which the target concept is considered a constant, and no information about the process that generates this concept is known a priori. Alternatively, we can consider a modification of this problem, so that the target concept can be thought of as a random variable, a sample from a known distribution (called a *prior*) over the space of possible concepts. Such a setting has been studied in detail in the context of passive learning for noise-free binary classification. In particular, (Haussler et al., 1994a) found that for any concept space of finite VC dimension d, for any prior and distribution over data points, $O(d/\varepsilon)$ random labeled examples are sufficient for the expected error rate of the Bayes classifier produced under the posterior distribution to be at most ε. Furthermore, it is easy to construct learning problems for which there is an $\Omega(1/\varepsilon)$ lower bound on the number of random labeled examples required to achieve expected error rate at most ε, by any passive learning algorithm; for instance, the problem of learning threshold classifiers on $[0, 1]$ under a uniform data distribution and uniform prior is one such scenario.

In the context of active learning (again, with access to the prior), (Freund et al., 1997) analyze the *Query by Committee* algorithm, and find that if a certain information gain quantity for the points requested by the algorithm is lower-bounded by a value g, then the algorithm requires only $O((d/g)\log(1/\varepsilon))$ labels to achieve expected error rate at most ε. In particular, they show that this is satisfied for *constant* g for linear separators under a near-uniform prior, and a near-uniform data distribution over the unit sphere. This represents a marked improvement over the results of (Haussler et al., 1994a) for passive learning, and since the Query by Committee algorithm is self-verifying, this result is highly relevant to the present discussion. However, the condition that the information gains be lower-bounded by a constant is quite restrictive, and many interesting learning problems are precluded by this requirement. Furthermore, there exist learning problems (with finite VC dimension) for which the Query by Committee algorithm makes an expected number of label requests exceeding $\Omega(1/\varepsilon)$. To date, there has not been a general analysis of how the value of g can behave as a function of ε, though such an analysis would likely be quite interesting.

In the present section, we take a more general approach to the question of active learning with access to the prior. We are interested in the broad question of whether access to the prior bridges the gap between the sample complexity of *learning* and the sample complexity of learning *with verification*. Specifically, we ask the following question.

Can a prior-dependent self-terminating active learning algorithm for a concept class of finite VC dimension always achieve expected error rate at most ε using $o(1/\varepsilon)$ label requests?

After some basic definitions in Section 3.2, In Section 3.3, we present a general result that the answer is *always* "yes." As the known results for the sample complexity of passive learning with access to the prior are typically $\propto 1/\varepsilon$ (Haussler et al., 1994a), and this is sometimes tight, this represents an improvement over passive learning. The proof (Yang et al., 2011) is simple and accessible, yet represents an important step in understanding the problem of self-termination in active learning algorithms, and the general issue of the complexity of verification. Also, as this is a result that does *not* generally hold for prior-independent algorithms (even for their "average-case" behavior induced by the prior) for certain concept spaces, this also represents a significant step toward understanding the inherent value of having access to the prior.

3.2 Definitions and Preliminaries

First, we introduce some notation and formal definitions. We denote by $\mathcal{X}$ the *instance space*, representing the range of the unlabeled data points, and we suppose a distribution $\mathcal{D}$ on $\mathcal{X}$, which we will refer to as the *data distribution*. We also suppose the existence of a sequence $X_1, X_2, \ldots$ of i.i.d. random variables, each with distribution $\mathcal{D}$, referred to as the unlabeled data sequence. Though one could potentially analyze the achievable performance as a function of the number of unlabeled points made available to the learning algorithm (cf. (Dasgupta, 2005)), for simplicity in the present work, we will suppose this unlabeled sequence is essentially inexhaustible, corresponding to the practical fact that unlabeled data are typically available in abundance as they are often relatively inexpensive to obtain. Additionally, there is a set $\mathbb{C}$ of measurable classifiers $h : \mathcal{X} \to \{-1, +1\}$, referred to as the *concept space*. We denote by d the VC dimension of $\mathbb{C}$, and in our present context we will restrict ourselves to spaces $\mathbb{C}$ with $d < \infty$, referred to as a *VC class*. We also have a probability distribution π, called the *prior*, over $\mathbb{C}$, and a random variable $h^* \sim \pi$, called the *target function*; we suppose h^* is independent from the data sequence $X_1, X_2, \ldots$. We adopt the usual notation for conditional expectations and probabilities (Ash & Doléans-Dade, 2000); for instance, $\mathbb{E}[A|B]$ can be thought of as an expectation of the value A, under the conditional distribution of A given the value of B (which itself is random), and thus the value of $\mathbb{E}[A|B]$ is essentially determined by the value of B. For any measurable $h : \mathcal{X} \to \{-1, +1\}$, define the *error rate* $\mathrm{er}(h) = \mathcal{D}(\{x : h(x) \neq h^*(x)\})$. So far, this setup is essentially identical to that of (Haussler et al., 1994a; Freund et al., 1997).

The protocol in active learning is the following. An active learning algorithm $\mathcal{A}$ is given as input the prior π, the data distribution $\mathcal{D}$, and a value $\varepsilon \in (0, 1]$. It also (implicitly) depends on the data sequence $X_1, X_2, \ldots$, and has an indirect dependence on the target function h^* via the following type of interaction. The algorithm may inspect the values X_i for any initial segment of the data sequence, select an index $i \in \mathbb{N}$ to "request" the label of; after selecting such an index, the algorithm receives the value $h^*(X_i)$. The algorithm may then select another index, request the label, receive the value of h^* on that point, etc. This happens for a number of rounds, $N(\mathcal{A}, h^*, \varepsilon, \mathcal{D}, \pi)$, before eventually the algorithm halts and returns a classifier $\hat{h}$. An algorithm is said to be *correct* if $\mathbb{E}\left[\mathrm{er}\left(\hat{h}\right)\right] \leq \varepsilon$ for every $(\varepsilon, \mathcal{D}, \pi)$; that is, given direct access to the prior and the data distribution, and given a specified value ε, a correct algorithm must be guaranteed to have expected error rate at most ε. Define the *expected sample complexity* of $\mathcal{A}$ for $(\mathcal{X}, \mathbb{C}, \mathcal{D}, \pi)$ to be the function $SC(\varepsilon, \mathcal{D}, \pi) = \mathbb{E}[N(\mathcal{A}, h^*, \varepsilon, \mathcal{D}, \pi)]$: the expected number of label requests the algorithm makes.

We will be interested in proving that certain algorithms achieve a sample complexity $SC(\varepsilon, \mathcal{D}, \pi) = o(1/\varepsilon)$. For some $(\mathcal{X}, \mathbb{C}, \mathcal{D})$, it is known that there are π-independent algorithms (meaning the algorithm's behavior is independent of the π argument) $\mathcal{A}$ such that we always have $\mathbb{E}[N(\mathcal{A}, h^*, \varepsilon, \mathcal{D}, \pi)|h^*] = o(1/\varepsilon)$; for instance, threshold classifiers have this property under any $\mathcal{D}$, homogeneous linear separators have this property under a uniform $\mathcal{D}$ on the unit sphere in k dimensions, and intervals with positive width on $\mathcal{X} = [0, 1]$ have this property under $\mathcal{D} = \mathrm{Uniform}([0, 1])$ (see e.g., (Dasgupta, 2005)). It is straightforward to show that any such $\mathcal{A}$ will also have $SC(\varepsilon, \mathcal{D}, \pi) = o(1/\varepsilon)$ for every π. In particular, the law of total expectation and the dominated convergence theorem imply

$$\lim_{\varepsilon \to 0} \varepsilon \cdot SC(\varepsilon, \mathcal{D}, \pi) = \lim_{\varepsilon \to 0} \varepsilon \cdot \mathbb{E}[\mathbb{E}[N(\mathcal{A}, h^*, \varepsilon, \mathcal{D}, \pi)|h^*]] = \mathbb{E}\left[\lim_{\varepsilon \to 0} \varepsilon \cdot \mathbb{E}[N(\mathcal{A}, h^*, \varepsilon, \mathcal{D}, \pi)|h^*]\right] = 0.$$

In these cases, we can think of SC as a kind of "average-case" analysis of these algorithms. However, there are also many $(\mathcal{X}, \mathbb{C}, \mathcal{D})$ for which no such π-independent algorithm exists, achieving $o(1/\varepsilon)$ sample complexity for *all* priors. For instance, this is the case for $\mathbb{C}$ as the space of interval classifiers (including the empty interval) on $\mathcal{X} = [0, 1]$ under $\mathcal{D} = \mathrm{Uniform}([0, 1])$ (this essentially follows from a proof of (Balcan et al., 2010)). Thus, any general result on $o(1/\varepsilon)$ expected sample complexity for π-dependent algorithms would signify that there is a real advantage to having access to the prior.

3.3 Main Result

In this section, we present our main result: a general result stating that $o(1/\varepsilon)$ expected sample complexity is always achievable by some correct active learning algorithm, for *any* $(\mathcal{X}, \mathbb{C}, \mathcal{D}, \pi)$ for which $\mathbb{C}$ has finite VC dimension. Since the known results for the sample complexity of passive learning with access to the prior are typically $\Theta(1/\varepsilon)$, and since there are known learning problems $(\mathcal{X}, \mathbb{C}, \mathcal{D}, \pi)$ for which every passive learning algorithm requires $\Omega(1/\varepsilon)$ samples, this $o(1/\varepsilon)$ result for active learning represents an improvement over passive learning. Additionally, as mentioned, this type of result is often not possible for algorithms lacking access to the prior π, as there are well-known problems $(\mathcal{X}, \mathbb{C}, \mathcal{D})$ for which no prior-independent correct algorithm (of the self-terminating type studied here) can achieve $o(1/\varepsilon)$ sample complexity for every prior π (Balcan et al., 2010); in particular, the intervals problem studied above is one such example.

Theorem 2. *For any VC class $\mathbb{C}$, there is a correct active learning algorithm that, for every data distribution $\mathcal{D}$ and prior π, achieves expected sample complexity SC for $(\mathcal{X}, \mathbb{C}, \mathcal{D}, \pi)$ such that*

$$SC(\varepsilon, \mathcal{D}, \pi) = o(1/\varepsilon).$$

4 Active Transfer Learning : Identifiability of Priors from Bounded Sample Sizes with Applications to Transfer Learning

One question often asked in regards to Bayesian methods is, "From where should we get our prior?" Indeed, this is often a problem. However, often times the answer is "from past experience." Thus, it seems we can come to *learn* an appropriate prior from past learning problems. This motivates a type of transfer learning. In this section, we propose a model to learn the prior, with a theoretical guarantee that, using the learned prior results in just as good performance as having direct access to the prior.

This can be thought of as a transfer learning setting, in which a finite sequence of target concepts are sampled independently with an unknown distribution from a known family. We study the total number of labeled examples required to learn all targets to an arbitrary specified expected accuracy, focusing on the asymptotics in the number of tasks and the desired accuracy. Our primary interest is formally understanding the fundamental benefits of transfer learning, compared to learning each target independently from the others. Our approach to the transfer problem is general, in the sense that it can be used with a variety of learning protocols. The key insight driving our approach is that the distribution of the target concepts is identifiable from the joint distribution over a number of random labeled data points equal the Vapnik-Chervonenkis dimension of the concept space. This is not necessarily the case for the joint distribution over any smaller number of points. This work has particularly interesting implications when applied to active learning methods.

4.1 Introduction

Transfer learning reuses knowledge from past related tasks to ease the process of learning to perform a new task. The goal of transfer learning is to leverage previous learning and experience to more efficiently learn novel, but related, concepts, compared to what would be possible without this prior experience. The utility of transfer learning is typically measured by a reduction in the number of training examples required to achieve a target performance on a sequence of related learning problems, compared to the number required for unrelated problems: i.e., reduced sample complexity. In many real-life scenarios, just a few training examples of a new concept or process is often sufficient for a human learner to grasp the new concept given knowledge of related ones. For example, learning to drive a van becomes much easier a task if we have already learned how to drive a car. Learning French is somewhat easier if we have already learned English (vs Chinese), and learning Spanish is easier if we know Portuguese (vs German). We are therefore interested in understanding the conditions that enable a learning machine to leverage abstract knowledge obtained as a by-product of learning past concepts, to improve its performance on future learning problems. Furthermore, we are interested in how the magnitude of these improvements grows as the learning system gains more experience from learning multiple related concepts.

The ability to transfer knowledge gained from previous tasks to make it easier to learn a new task can potentially benefit a wide range of real-world applications, including computer vision, natural language processing, cognitive science (e.g., fMRI brain state classification), and speech recognition, to name a few. As an example, consider training a speech recognizer. After training on a number of individuals, a learning system can identify common patterns of speech, such as accents or dialects, each of which requires a slightly different speech recognizer; then, given a new person to train a recognizer for, it can quickly determine the particular dialect from only a few well-chosen examples, and use the previously-learned recognizer for that particular dialect. In this case, we can think of the transferred knowledge as consisting of the common aspects of each recognizer variant and more generally the *distribution* of speech patterns existing in the population these subjects are from. This same type of distribution-related knowledge transfer can be helpful in a host of applications, including all those mentioned above.

Supposing these target concepts (e.g., speech patterns) are sampled independently from a fixed population, having knowledge of the distribution of concepts in the population may often be quite valuable. More generally, we may consider a general scenario in which the target concepts are sampled i.i.d. according to a fixed distribution. As we show below, the number of labeled examples required to learn a target concept sampled according to this distribution may be dramatically reduced if we have direct knowledge of the distribution. However, since in many real-world learning scenarios, we do not have direct access to this distribution, it is desirable to be able to somehow *learn* the distribution, based on observations from a sequence of learning problems with target concepts sampled according to that distribution. The hope is that an estimate of the distribution so-obtained might be almost as useful as direct access to the true distribution in reducing the number of labeled examples required to learn subsequent target concepts. The focus of this section is an approach to transfer learning based on estimating the distribution of the target concepts. Whereas we acknowledge that there are other important challenges in transfer learning, such as exploring improvements obtainable from transfer under various alternative notions of task relatedness (Evgeniou & Pontil, 2004; Ben-David & Schuller, 2003), or alternative reuses of knowledge obtained from previous tasks (Thrun, 1996), we believe that learning the distribution of target concepts is a central and crucial component in many transfer learning scenarios, and can reduce the total sample complexity across tasks.

Note that it is not immediately obvious that the distribution of targets can even be learned in this context, since we do not have direct access to the target concepts sampled according to it, but rather have only indirect access via a finite number of labeled examples for each task; a significant part of the present work focuses on establishing that as long as these finite labeled samples are larger than a certain size, they hold sufficient information about the distribution over concepts for estimation to be possible. In particular, in contrast to standard results on consistent density estimation, our estimators are not directly based on the target concepts, but rather are only indirectly dependent on these via the labels of a finite number of data points from each task. One desideratum we pay particular attention to is minimizing the number of *extra* labeled examples needed for each task, beyond what is needed for learning that particular target, so that the benefits of transfer learning are obtained almost as a *by-product* of learning the targets. Our technique is general, in that it applies to any concept space with finite VC dimension; also, the process of learning the target concepts is (in some sense) decoupled from the mechanism of learning the concept distribution, so that we may apply our technique to a variety of learning protocols, including passive supervised learning, active supervised learning, semi-supervised learning, and learning with certain general data-dependent forms of interaction (Hanneke, 2009). For simplicity, we choose to formulate our transfer learning algorithms in the language of active learning; as we explain below, this problem can benefit significantly from transfer. Formulations for other learning protocols would follow along similar lines, with analogous theorems; only the results in Section 4.7 are specific to active learning.

Transfer learning is related at least in spirit to much earlier work on case-based and analogical learning (Carbonell, 1983; Carbonell, 1986; Veloso & Carbonell, 1993; Kolodner (Ed), 1993; Thrun, 1996), although that body of work predated modern machine learning, and focused on symbolic reuse of past problem solving solutions rather than on current machine learning problems such as classification, regression or structured learning. More recently, transfer learning (and the closely related problem of *multitask* learning) has been studied in specific cases with interesting (though sometimes heuristic) approaches (Caruana, 1997; Silver, 2000; Micchelli & Pontil, 2004; Baxter, 1997; Ben-David & Schuller, 2003). This section considers a general theoretical framework for transfer learning, based on an Empirical Bayes perspective, and derives rigorous theoretical results on the benefits of transfer. We discuss the relation of this analysis to existing theoretical work on transfer learning below.

The remainder of the section is organized as follows. In Section 4.2 we introduce basic notation used throughout, and survey some related work from the existing literature. In Section 4.4, we describe and analyze our proposed method for estimating the distribution of target concepts, the key ingrediant in our approach to transfer learning, which we then present in Section 4.6. Finally, in Section 4.7, we describe the particularly strong implications of these results for active learning.

4.2 Definitions and Related Work

First, we state a few basic notational conventions. We denote $\mathbb{N} = \{1, 2, \ldots\}$ and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. For any random variable X, we generally denote by $\mathbb{P}_X$ the distribution of X (the induced probability measure on the range of X), and by $\mathbb{P}_{X|Y}$ the regular conditional distribution of X given Y. For any pair of probability measures μ_1, μ_2 on a measurable space $(\Omega, \mathcal{F})$, we define

$$\|\mu_1 - \mu_2\| = \sup_{A \in \mathcal{F}} |\mu_1(A) - \mu_2(A)|.$$

Next we define the particular objects of interest to our present discussion. Let Θ be an arbitrary set (called the *parameter space*), $(\mathcal{X}, \mathcal{B}_\mathcal{X})$ be a Borel space (Schervish, 1995) (where $\mathcal{X}$ is called the *instance space*), and $\mathcal{D}$ be a fixed distribution on $\mathcal{X}$ (called the *data distribution*). For instance, Θ could be $\mathbb{R}^n$ and $\mathcal{X}$ could be $\mathbb{R}^m$, for some $n, m \in \mathbb{N}$, though more general scenarios are certainly possible as well, including infinite-dimensional parameter spaces. Let $\mathbb{C}$ be a set of measurable classifiers $h : \mathcal{X} \to \{-1, +1\}$ (called the *concept space*), and suppose $\mathbb{C}$ has VC dimension $d < \infty$ (Vapnik, 1982) (such a space is called a *VC class*). $\mathbb{C}$ is equipped with its Borel σ-algebra $\mathcal{B}$, induced by the pseudo-metric $\rho(h, g) = \mathcal{D}(\{x \in \mathcal{X} : h(x) \neq g(x)\})$. Though all of our results can be formulated for general $\mathcal{D}$ in slightly more complex terms, for simplicity throughout the discussion below we suppose ρ is actually a *metric*, in that any $h, g \in \mathbb{C}$ with $h \neq g$ have $\rho(h, g) > 0$; this amounts to a topological assumption on $\mathbb{C}$ relative to $\mathcal{D}$.

For each $\theta \in \Theta$, π_θ is a distribution on $\mathbb{C}$ (called a *prior*). Our only (rather mild) assumption on this family of prior distributions is that $\{\pi_\theta : \theta \in \Theta\}$ be totally bounded, in the sense that $\forall \varepsilon > 0$, $\exists$ *finite* $\Theta_\varepsilon \subseteq \Theta$ s.t. $\forall \theta \in \Theta, \exists \theta_\varepsilon \in \Theta_\varepsilon$ with $\|\pi_\theta - \pi_{\theta_\varepsilon}\| < \varepsilon$. See (Devroye & Lugosi, 2001) for examples of categories of classes that satisfy this.

The general setup for the learning problem is that we have a *true* parameter value $\theta_\star \in \Theta$, and a collection of $\mathbb{C}$-valued random variables $\{h^*{}_{t\theta}\}_{t \in \mathbb{N}, \theta \in \Theta}$, where for a fixed $\theta \in \Theta$ the $\{h^*{}_{t\theta}\}_{t \in \mathbb{N}}$ variables are i.i.d. with distribution π_θ.

The learning problem is the following. For each $\theta \in \Theta$, there is a sequence

$$\mathcal{Z}_t(\theta) = \{(X_{t1}, Y_{t1}(\theta)), (X_{t2}, Y_{t2}(\theta)), \ldots\},$$

where $\{X_{ti}\}_{t,i \in \mathbb{N}}$ are i.i.d. $\mathcal{D}$, and for each $t, i \in \mathbb{N}$, $Y_{ti}(\theta) = h^*{}_{t\theta}(X_{ti})$. For $k \in \mathbb{N}$ we denote by $\mathcal{Z}_{tk}(\theta) = \{(X_{t1}, Y_{t1}(\theta)), \ldots, (X_{tk}, Y_{tk}(\theta))\}$.

The algorithm receives values ε and T as input, and for each $t \in \{1, 2, \ldots, T\}$ in increasing order, it observes the sequence $X_{t1}, X_{t2}, \ldots$, and may then select an index i_1, receive label $Y_{ti_1}(\theta_\star)$, select another index i_2, receive label $Y_{ti_2}(\theta_\star)$, etc. The algorithm proceeds in this fashion, sequentially requesting labels, until eventually it produces a classifier $\hat{h}_t$. It then increments t and repeats this process until it produces a sequence $\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_T$, at which time it halts. To be called *correct*, the algorithm must have a guarantee that $\forall \theta_\star \in \Theta, \forall t \leq T, \mathbb{E}\left[\rho\left(\hat{h}, h^*{}_{t\theta_\star}\right)\right] \leq \varepsilon$. We will be interested in the expected number of label requests necessary for a correct learning algorithm, averaged over the T tasks, and in particular in how shared information between tasks can help to reduce this quantity when direct access to $\theta_\star$ is not available to the algorithm.

4.3 Relation to Existing Theoretical Work on Transfer Learning

Although we know of no existing work on the theoretical advantages of transfer learning for active learning, the existing literature contains several analyses of the advantages of transfer learning for passive learning. In his classic work, (Baxter, 1997) explores a similar setup for a general form of passive learning, except in a *full* Bayesian setting (in contrast to our setting, often referred to as "empirical Bayes," which includes a constant parameter $\theta_\star$ to be estimated from data). Essentially, (Baxter, 1997) sets up a hierarchical Bayesian model, in which (in our notation) $\theta_\star$ is a random variable with known distribution (hyper-prior), but otherwise the specialization of Baxter's setting to the pattern recognition problem is essentially identical to our setup above. This hyper-prior does make the problem slightly easier, but generally the results of (Baxter, 1997) are of a different nature than our objectives here. Specifically, Baxter's results on learning from labeled examples can be interpretted as indicating that transfer learning can improve certain *constant factors* in the asymptotic rate of convergence of the average of expected error rates across the learning problems. That is, certain constant complexity terms (for instance, related to the concept space) can be reduced to (potentially much smaller) values related to $\pi_{\theta_\star}$ by transfer learning. Baxter argues that, as the number of tasks grows large, this effectively achieves close to the known results on the sample complexity of passive learning with direct access to $\theta_\star$. A similar claim is discussed by (Ando & Zhang, 2004) (though in less detail and formality) for a setting closer to that studied here, where $\theta_\star$ is an unknown parameter to be estimated.

There are also several results on transfer learning of a slightly different variety, in which, rather than having a prior distribution for the target concept, the learner initially has several potential concept spaces to choose from, and the role of transfer is to help the learner select from among these concept spaces (Baxter, 2000; Ando & Zhang, 2004). In this case, the idea is that one of these concept spaces has the best average minimum achievable error rate per learning problem, and the objective of transfer learning is to perform nearly as well as if we knew which of the spaces has this property. In particular, if we assume the target functions for each task all reside in one of the concept spaces, then the objective of transfer learning is to perform nearly as well as if we knew which of the spaces contains the targets. Thus, transfer learning results in a sample complexity related to the number of learning problems, a complexity term for this best concept space, and a complexity term related to the diversity of concept spaces we have to choose from. In particular, as with (Baxter, 1997), these results can typically be interpretted as giving constant factor improvements from transfer in a passive learning context, at best reducing the complexity constants, from those for the union over the given concept spaces, down to the complexity constants of the single best concept space.

In addition to the above works, there are several analyses of transfer learning and multitask learning of an entirely different nature than our present discussion, in that the objectives of the analysis are somewhat different. Specifically, there is a branch of the literature concerned with task *relatedness*, not in terms of the underlying process that generates the target concepts, but rather directly in terms of relations between the target concepts themselves. In this sense, several tasks with related target concepts should be much easier to learn than tasks with unrelated target concepts. This is studied in the context of kernel methods by (Micchelli & Pontil, 2004; Evgeniou & Pontil, 2004; Evgeniou et al., 2005), and in a more general theoretical framework by (Ben-David & Schuller, 2003). As mentioned, our approach to transfer learning is based on the idea of estimating the distribution of target concepts. As such, though interesting and important, these notions of direct relatedness of target concepts are not as relevant to our present discussion.

As with (Baxter, 1997), the present work is interested in showing that as the number of tasks grows large, we can effectively achieve a sample complexity close to that achieveable with direct access to $\theta_\star$. However, in contrast, we are interested in a general approach to transfer learning and the analysis thereof, leading to concrete results for a variety of learning protocols such as active learning and semi-supervised learning. In particular, as we explain below, combining the results of this work with a result of (Yang et al., 2011) reveals the interesting phenomenon that, in the context of active learning, transfer learning can sometimes improve the asymptotic dependence on ε, rather than merely the constant factors as in the analysis of (Baxter, 1997).

Additionally, unlike (Baxter, 1997), we study the benefits of transfer learning in terms of the asymptotics as the number of learning problems grows large, *without* necessarily requiring the number of labeled examples per learning problem to also grow large. That is, our analysis reveals benefits from transfer learning even if the number of labeled examples per learning problem is *bounded*. This is desirable for the following practical reasons. In many settings where transfer learning may be useful, it is desirable that the number of labeled examples we need to collect from each particular learning problem never be significantly larger than the number of such examples required to solve that particular problem (i.e., to learn that target concept to the desired accuracy). For instance, this is the case when the learning problems are not all solved by the same individual (or company, etc.), but rather a coalition of cooperating individuals (e.g., hospitals sharing data on clinical trials); each individual may be willing to share the data they used to learn their problem, in the interest of making others' learning problems easier; however, they may not be willing to collect significantly *more* data to advance this cause than they themselves need for their own learning problem. Given a desired error rate ε for each learning problem, the number of labeled examples required to learn each particular target concept to this desired error rate is always bounded by an ε-dependent value. Therefore, an analysis that requires a growing number of examples per learning problem seems undesirable in these scenarios, since for some of the problems we would need to label a number of examples far beyond what is needed to learn a good classifier for that particular problem. We should therefore be particularly interested in studying transfer as a *by-product* of the usual learning process; failing this, we are interested in the minimum possible number of *extra* labeled examples per task to gain the benefits of transfer learning. To our knowledge, no result of this type (bounded sample size per learning problem) has yet been established at the level of generality studied here.

4.4 Estimating the Prior

The advantage of transfer learning in this setting is that each learning problem provides some information about $\theta_\star$, so that after solving several of the learning problems, we might hope to be able to *estimate* $\theta_\star$. Then with this estimate in hand, we can use the corresponding estimated prior distribution in the learning algorithm for subsequent learning problems, to help inform the learning process similarly to how direct knowledge of $\theta_\star$ might be helpful. However, the difficulty in approaching this is how to define such an estimator. Since we do not have direct access to the ${h^*}_t$ values, but rather only indirect observations via a finite number of example labels, the standard results for density estimation from i.i.d. samples cannot be applied.

The idea we pursue below is to consider the distributions on $\mathcal{Z}_{tk}(\theta_\star)$. These variables *are* directly observable, by requesting the labels of those examples. Thus, for any finite $k \in \mathbb{N}$, this distribution *is* estimable from observable data. That is, using the i.i.d. values $\mathcal{Z}_{1k}(\theta_\star), \ldots, \mathcal{Z}_{tk}(\theta_\star)$, we can apply standard techniques for density estimation to arrive at an estimator of $\mathbb{P}_{\mathcal{Z}_{tk}(\theta_\star)}$. Then the question is whether the distribution $\mathbb{P}_{\mathcal{Z}_{tk}(\theta_\star)}$ uniquely characterizes the prior distribution $\pi_{\theta_\star}$: that is, whether $\pi_{\theta_\star}$ is *identifiable* from $\mathbb{P}_{\mathcal{Z}_{tk}(\theta_\star)}$.

As an example, consider the space of *half-open interval* classifiers on $[0,1]$: $\mathbb{C} = \{\mathbb{I}^{\pm}_{[a,b)} : 0 \le a \le b \le 1\}$, where $\mathbb{I}^{\pm}_{[a,b)}(x) = +1$ if $a \le x < b$ and -1 otherwise. In this case, $\pi_{\theta_\star}$ is *not* necessarily identifiable from $\mathbb{P}_{\mathcal{Z}_{t1}(\theta_\star)}$; for instance, the distributions π_{θ_1} and π_{θ_2} characterized by $\pi_{\theta_1}(\{\mathbb{I}^{\pm}_{[0,1)}\}) = \pi_{\theta_1}(\{\mathbb{I}^{\pm}_{\emptyset}\}) = 1/2$ and $\pi_{\theta_2}(\{\mathbb{I}^{\pm}_{[0,1/2)}\}) = \pi_{\theta_2}(\{\mathbb{I}^{\pm}_{[1/2,1)}\}) = 1/2$ are not distinguished by these one-dimensional distributions. However, it turns out that for this half-open intervals problem, $\pi_{\theta_\star}$ *is* uniquely identifiable from $\mathbb{P}_{\mathcal{Z}_{t2}(\theta_\star)}$; for instance, in the θ_1 vs θ_2 scenario, the conditional probability $\mathbb{P}_{(Y_{t1}(\theta_i),Y_{t2}(\theta_i))|(X_{t1},X_{t2})}((+1,+1)|(1/4,3/4))$ will distinguish π_{θ_1} from π_{θ_2}, and this can be calculated from $\mathbb{P}_{\mathcal{Z}_{t2}(\theta_i)}$. The crucial element of the analysis below is determining the appropriate value of k to uniquely identify $\pi_{\theta_\star}$ from $\mathbb{P}_{\mathcal{Z}_{tk}(\theta_\star)}$ *in general*. As we will see, $k = d$ is *always* sufficient, a key insight for the results that follow.

To be specific, in order to transfer knowledge from one task to the next, we use a few labeled data points from each task to gain information about $\theta_\star$. For this, for each task t, we simply take the first d data points in the $\mathcal{Z}_t(\theta_\star)$ sequence. That is, we request the labels

$$Y_{t1}(\theta_\star), Y_{t2}(\theta_\star), \ldots, Y_{td}(\theta_\star)$$

and use the points $\mathcal{Z}_{td}(\theta_\star)$ to update an estimate of $\theta_\star$.

The following result shows that this technique does provide a consistent estimator of $\pi_{\theta_\star}$. Again, note that this result is not a straightforward application of the standard approach to consistent estimation, since the observations here are not the ${h^*}_{t\theta_\star}$ variables themselves, but rather a number of the $Y_{ti}(\theta_\star)$ values. The key insight in this result is that $\pi_{\theta_\star}$ is *uniquely identified* by the joint distribution $\mathbb{P}_{\mathcal{Z}_{td}(\theta_\star)}$ over the first d labeled examples; later, we prove this is *not* necessarily true for $\mathbb{P}_{\mathcal{Z}_{tk}(\theta_\star)}$ for values $k < d$.

Theorem 3. *There exists an estimator $\hat\theta_{T\theta_\star} = \hat\theta_T(\mathcal{Z}_{1d}(\theta_\star), \ldots, \mathcal{Z}_{Td}(\theta_\star))$, and functions $R : \mathbb{N}_0 \times (0,1] \to [0,\infty)$ and $\delta : \mathbb{N}_0 \times (0,1] \to [0,1]$, such that for any $\alpha > 0$, $\lim\limits_{T\to\infty} R(T,\alpha) = \lim\limits_{T\to\infty} \delta(T,\alpha) = 0$ and for any $T \in \mathbb{N}_0$ and $\theta_\star \in \Theta$,*

$$\mathbb{P}\left(\|\pi_{\hat\theta_{T\theta_\star}} - \pi_{\theta_\star}\| > R(T,\alpha)\right) \le \delta(T,\alpha) \le \alpha.$$

One important detail to note, for our purposes, is that $R(T,\alpha)$ is independent from $\theta_\star$, so that the value of $R(T,\alpha)$ can be calculated and used within a learning algorithm.

4.5 Identifiability from d Points

Inspection of the above proof reveals that the assumption that the family of priors is totally bounded is required only to establish the estimability and bounded rate guarantees. In particular, the implied identifiability condition is, in fact, *always* satisfied, as stated formally in the following corollary.

Corollary 1. *For any priors π_1, π_2 on $\mathbb{C}$, if $h^*{}_i \sim \pi_i$, $X_1, \ldots, X_d$ are i.i.d. $\mathcal{D}$ independent from $h^*{}_i$, and $Z_d(i) = \{(X_1, h^*{}_i(X_1)), \ldots, (X_d, h^*{}_i(X_d))\}$ for $i \in \{1, 2\}$, then $\mathbb{P}_{Z_d(1)} = \mathbb{P}_{Z_d(2)} \implies \pi_1 = \pi_2$.*

It is natural to wonder whether this identifiability remains true for some smaller number of points $k < d$, so that we might hope to create an estimator for $\pi_{\theta_\star}$ based on an estimator for $\mathbb{P}_{\mathcal{Z}_{tk}(\theta_\star)}$. However, one can show that d is actually the *minimum* possible value for which this remains true for all $\mathcal{D}$ and all families of priors. Formally, we have the following result, holding for every VC class $\mathbb{C}$.

Theorem 4. *There exists a data distribution $\mathcal{D}$ and priors π_1, π_2 on $\mathbb{C}$ such that, for any positive integer $k < d$, if $h^*{}_i \sim \pi_i$, $X_1, \ldots, X_k$ are i.i.d. $\mathcal{D}$ independent from $h^*{}_i$, and $Z_k(i) = \{(X_1, h^*{}_i(X_1)), \ldots, (X_k, h^*{}_i(X_k))\}$ for $i \in \{1, 2\}$, then $\mathbb{P}_{Z_k(1)} = \mathbb{P}_{Z_k(2)}$ but $\pi_1 \neq \pi_2$.*

4.6 Transfer Learning

In this section, we look at an application of the techniques from the previous section to transfer learning. Like the previous section, the results in this section are general, in that they are applicable to a variety of learning protocols, including passive supervised learning, passive semi-supervised learning, active learning, and learning with certain general types of data-dependent interaction (Hanneke, 2009). For simplicity, we restrict our discussion to the active learning formulation; the analogous results for these other learning protocols follow by similar reasoning.

The result of the previous section implies that an estimator for $\theta_\star$ based on d-dimensional joint distributions is consistent with a bounded rate of convergence R. Therefore, for certain prior-dependent learning algorithms, their behavior should be similar under $\pi_{\hat{\theta}_{T\theta_\star}}$ to their behavior under $\pi_{\theta_\star}$.

To make this concrete, we formalize this in the active learning protocol as follows. A *prior-dependent* active learning algorithm $\mathcal{A}$ takes as inputs $\varepsilon > 0$, $\mathcal{D}$, and a distribution π on $\mathbb{C}$. It initially has access to $X_1, X_2, \ldots$ i.i.d. $\mathcal{D}$; it then selects an index i_1 to request the label for, receives $Y_{i_1} = h^*(X_{i_1})$, then selects another index i_2, etc., until it eventually terminates and returns a classifier. Denote by $\mathcal{Z} = \{(X_1, h^*(X_1)), (X_2, h^*(X_2)), \ldots\}$. To be *correct*, the algorithm $\mathcal{A}$ must guarantee that for $h^* \sim \pi$, $\forall \varepsilon > 0$, $\mathbb{E}\left[\rho(\mathcal{A}(\varepsilon, \mathcal{D}, \pi), h^*)\right] \leq \varepsilon$. We define the random variable $N(\mathcal{A}, f, \varepsilon, \mathcal{D}, \pi)$ as the number of label requests $\mathcal{A}$ makes before terminating, when given ε, $\mathcal{D}$, and π as inputs, and when $h^* = f$ is the value of the target function; we make the particular data sequence $\mathcal{Z}$ the algorithm is run with implicit in this notation. We will be interested in the *expected sample complexity* $SC(\mathcal{A}, \varepsilon, \mathcal{D}, \pi) = \mathbb{E}\left[N(\mathcal{A}, h^*, \varepsilon, \mathcal{D}, \pi)\right]$.

We propose the following algorithm $\mathcal{A}_\tau$ for transfer learning, defined in terms of a given correct prior-dependent active learning algorithm $\mathcal{A}_a$. We discuss interesting specifications for $\mathcal{A}_a$ in the next section, but for now the only assumption we require is that for any $\varepsilon > 0$ and $\mathcal{D}$, there is a value $s_\varepsilon < \infty$ such that for every π and $f \in \mathbb{C}$, $N(\mathcal{A}_a, f, \varepsilon, \mathcal{D}, \pi) \leq s_\varepsilon$; this is a very mild requirement, and any active learning algorithm can be converted into one that satisfies this without significantly increasing its sample complexities for the priors it is already good for (Balcan et al., 2010). We denote by $m_\varepsilon = \frac{16d}{\varepsilon} \ln\left(\frac{24}{\varepsilon}\right)$, and $\mathrm{B}(\theta, \gamma) = \{\theta' \in \Theta : \|\pi_\theta - \pi_{\theta'}\| \leq \gamma\}$.

Theorem 5. *The algorithm $\mathcal{A}_\tau$ is correct. Furthermore, if $S_T(\varepsilon)$ is the total number of label requests made by $\mathcal{A}_\tau(T, \varepsilon)$, then $\limsup\limits_{T \to \infty} \frac{\mathbb{E}[S_T(\varepsilon)]}{T} \leq SC(\mathcal{A}_a, \varepsilon/4, \mathcal{D}, \pi_{\theta_\star}) + d$.*

The remarkable implication of Theorem 5 is that, via transfer learning, it is possible to achieve almost the *same* long-run average sample complexity as would be achievable if the target's prior distribution were *known* to the learner. We will see in the next section that this is sometimes significantly better than the single-task sample complexity.

Returning to our motivational remarks from Subsection 4.3, we can ask how many *extra* labeled examples are required from each learning problem to gain the benefits of transfer learning. This question essentially concerns the initial step of requesting the labels $Y_{t1}(\theta_\star), \ldots, Y_{td}(\theta_\star)$. Clearly this indicates that from each learning problem, we need at most d extra labeled examples to gain the benefits of transfer. Whether these d label requests are indeed *extra* depends on the particular learning algorithm $\mathcal{A}_a$; that is, in some cases (e.g., certain passive learning algorithms), $\mathcal{A}_a$ may itself use these initial d labels for learning, so that in these cases the benefits of transfer

Algorithm 1 $\mathcal{A}_\tau(T, \varepsilon)$: an algorithm for transfer learning, specified in terms of a generic subroutine $\mathcal{A}_a$.

```
for $t = 1, 2, \ldots, T$ do
  Request labels $Y_{t1}(\theta_\star), \ldots, Y_{td}(\theta_\star)$
  if $R(t-1, \varepsilon/2) > \varepsilon/8$ then
    Request labels $Y_{t(d+1)}(\theta_\star), \ldots, Y_{tm_\varepsilon}(\theta_\star)$
    Take $\hat{h}_t$ as any $h \in \mathbb{C}$ s.t. $\forall i \leq m_\varepsilon$, $h(X_{ti}) = Y_{ti}(\theta_\star)$
  else
    Let $\check{\theta}_{t\theta_\star} \in \mathrm{B}\left(\hat{\theta}_{(t-1)\theta_\star}, R(t-1, \varepsilon/2)\right)$ be such that
    $SC(\mathcal{A}_a, \varepsilon/4, \mathcal{D}, \pi_{\check{\theta}_{t\theta_\star}}) \leq \min_{\theta \in \mathrm{B}\left(\hat{\theta}_{(t-1)\theta_\star}, R(t-1,\varepsilon/2)\right)} SC(\mathcal{A}_a, \varepsilon/4, \mathcal{D}, \pi_\theta) + 1/t$
    Run $\mathcal{A}_a(\varepsilon/4, \mathcal{D}, \pi_{\check{\theta}_{t\theta_\star}})$ with data sequence $\mathcal{Z}_t(\theta_\star)$ and let $\hat{h}_t$ be the classifier it returns
  end if
end for
```

learning are essentially gained as a *by-product* of the learning processes, and essentially no additional labeling effort need be expended to gain these benefits. On the other hand, for some active learning algorithms, we may expect that at least some of these initial d labels would not be requested by the algorithm, so that some extra labeling effort is expended to gain the benefits of transfer in these cases.

4.7 Application to Self-Verifying Active Learning

Recent work of (Yang et al., 2011) shows that there exists a correct prior-dependent active learning algorithm $\mathcal{A}$ such that, for any prior π over $\mathbb{C}$, $SC(\mathcal{A}, \varepsilon, \mathcal{D}, \pi) = o(1/\varepsilon)$. This is interesting, in that it contrasts with established results for correct prior-independent active learning algorithms, where there are known problems $(\mathbb{C}, \mathcal{D})$ for which any prior-independent active learning algorithm $\mathcal{A}'$ that is correct (in the sense studied above) has some prior π for which $SC(\mathcal{A}', \varepsilon, \mathcal{D}, \pi) = \Omega(1/\varepsilon)$; for instance, the class of interval classifiers on $[0, 1]$ under a uniform distribution $\mathcal{D}$ satisfies this (Balcan et al., 2010).

Combined with the results above for transfer learning, we get an immediate corollary that, running $\mathcal{A}_\tau$ with the active learning algorithm $\mathcal{A}$ having this $o(1/\varepsilon)$ sample complexity guarantee, we have

$$\limsup_{T \to \infty} \frac{\mathbb{E}[S_T(\varepsilon)]}{T} = o(1/\varepsilon).$$

Thus, in the case of active learning, there are scenarios where transfer learning (of the type studied here) can provide significant improvements in the average expected sample complexity, including improvements to the asymptotic dependence on ε.

In conclusion, we have shown that when learning a sequence of i.i.d. target concepts from a known VC class, with an unknown distribution from a known totally bounded family, transfer learning can lead to amortized expected sample complexity close to that achievable by an algorithm with direct knowledge of the the targets' distribution. Furthermore, the number of extra labeled examples per task, beyond what is needed for learning that task, is bounded by the VC dimension of the class. The key insight leading to this result is that the prior distribution is uniquely identifiable based on the joint distribution over the first VC dimension number of points. This is not necessarily the case for the distribution over any number of points less than the VC dimension. As a particularly interesting application, we note that in the context of active learning, transfer learning of this type can even lead to improvements in the asymptotic dependence on the desired error rate guarantee ε in the average expected sample complexity, and in particular can guarantee this average is $o(1/\varepsilon)$.

5 Future Research Problems

5.1 Active Property Testing

In property testing, we are interested in properties of combinatorial objects, i.e. using only a small number of queries to the object, whether we are able to distinguish objects that have the property from objects that are far from having the property. The recent surveys report the rapid progress on property testing in recent years (Goldreich, 2010), mainly related to this thesis are sub-linear time algorithms (Parnas et al., 2002) and connections to learning theory (Ron, 2008; Ron, 2009). In PAC learning, the sample complexity is well understood. However, information theoretic measures, query complexity and sample complexity of property testing, are not well understood at all.

Work on property testing is naturally categorized by the type of functions and properties to study. The combinatorial objects we focus on studying testing with is Boolean functions, which are important in machine learning, complexity theory, coding theory, combinatorics, etc. It reduces to characterizing testable properties of boolean functions, i.e. the set of iff conditions for a property to be testable with a constant number of queries. A boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ is a function that takes a n-dimensional boolean vector as input, and outputs a binary value. The distance between two boolean functions f, g is defined as the probability mass that they disagree on an input chosen at random with a distribution $\mathcal{D}$ on $\{0,1\}^n$ (typically the uniform distribution), i.e. $\Pr_{x\sim\mathcal{D}}[f(x) \neq g(x)]$. We say a Boolean-valued function g is ϵ-far from f if $\Pr_{\mathbf{x}\sim\mathcal{D}}(f(\mathbf{x}) \neq g(\mathbf{x})) \geq \epsilon$.

Let $q(n) : \mathbb{N} \rightarrow \mathbb{N}$ and $\epsilon > 0$. A randomized algorithm $\mathcal{A}$ with oracle access to an unknown boolean function f is a (q, ϵ)-tester for the property $\mathcal{P}$ if it queries f on at most $q(n)$ inputs and accepts f with probability at least $2/3$ when $f \in \mathcal{P}$; and rejects f with probability at least $2/3$ when f is ϵ-far from $\mathcal{P}$. A tester that accepts functions in $\mathcal{P}$ with probability 1 has one-sided error. A tester that fixes all of its queries ahead of time is non-adaptive; otherwise adaptive. The query complexity of ϵ-testing the property $\mathcal{P}$ is the function $Q_{\mathcal{P},\epsilon}(n) : \mathbb{N} \rightarrow \mathbb{N}$ the minimal value of $Q(n)$ for which there is a q, ϵ-tester for $\mathcal{P}$.

We are interested in testers that have significantly lower query complexity. In the random sample model (let us call it passive testing), the tester has no control over the inputs queried. The other type of popular tester so far are designed with membership query (MQ) : the tester can construct its own example and query the oracle machine, and thus the tester has full control over the queries it chooses to make. This type of query can be unrealistic in many real application. We often find the example to query is in some sense unusual or rare, so that it may be difficult for an expert to label. We study the model of property testing using active query, which allows the tester to actively pick the examples from among a pool of random unlabeled data. It is interesting because it seeks some middle ground between the membership query property testing model and passive testing.

Below we state our results (lower and upper bounds) for the query complexity of active testing, MQ-testing, passive testing, and active learning, on a number of concept class : Testing Unions of d Intervals, Union of d Thresholds, Dictator functions, Linear Threshold functions, and Linearity.

Specifically, define the following concept spaces:

- Unions of d Intervals: this space is defined on the instance space $[0,1]$, and consists of classifiers whose positive region can be described as $\bigcup_{i=1}^{d}[a_i, b_i]$.
- Unions of d Thresholds: this space is defined on the instance space $[0,d]$, and consists of classifiers whose positive region can be described as $\bigcup_{i=1}^{d}[a_i, i]$, for values $a_i \in (i-1, i)$.
- Linear Threshold Functions: this space is defined on the instance space $\{-1,1\}^n$ (or $\mathbb{R}^n$), and consists of classifiers whose positive region can be described as those x s.t. $w \cdot x \geq 0$, for values $w \in \mathbb{R}^n$.
- Dictator Functions: this space is defined on the instance space $\{-1,1\}^n$, and consists of classifiers whose positive region can be described as those $x \in \{-1,1\}^n$ such that $x_i = 1$, for values $i \in \{1,\ldots,n\}$; that is, this concept space has n classifiers in it, and each predicts in agreement with a single feature.
- Linearity: this space is defined on the instance space $\{0,1\}^n$, and consists of classifiers whose positive region can be described as a parity function for some subset of the features.

In summary, Table 1 compares the query complexity of the active tester with MQ-tester and passive tester; and compares testing with learning. Pondering on the table, we discover some interesting patterns: there is a noticeable gap between active testing and passive testing, for both low-dimensional concept classes. For high-dimensional concept classes, there is no such gap. However, I am currently working on constructing a natural class in high-dimensional instance spaces that demonstrates such a gap. We observe that testing is easier than learning (using smaller number of queries). Active testing on various concept classes covers a variety of complexities.

Let n be the number of dimensions. In each case, we consider an appropriate uniform distribution $\mathcal{D}$.

5.2 Efficient Learning with General Queries

In this section, we describe a general topic, in which we approach learning problems for which there are no known efficient algorithms in the (passive) PAC learning model, and construct problem-specific types of queries which can then be used to efficiently learning in those settings.

Our primary example in this proposal is learning DNF boolean functions, for which there are presently no known computationally efficient PAC learning algorithms. We describe a type of query that asks, for pairs of

Table 1. Testing on various concept classes

	testing with membership query	active testing	passive testing	active learning
Union of d Thresholds	constant	constant	$\sqrt{d}$	d
Union of d Intervals	constant	constant	$\sqrt{d}$	d
Linear Threshold Functions	constant	$\sqrt{n}$	$\sqrt{n}$	n
Dictator	constant	$\log n$	$\log n$	$\log n$
Linearity	constant	$n/\log n$	$n/\log n$	n?

examples from a reasonably-sized pool, whether there are any terms in a parsimonious representation of the target DNF which are satisfied by both examples. We construct an efficient learning algorithm based on these types of queries. We additionally discuss other types of queries of a similar nature, but which do not require the "parsimonious" restriction.

This is but one example in the general topic, and we plan to investigate a variety of other such examples in the final thesis.

Efficiently Learning DNF via General Queries We are interested in the type of protocol/interaction that takes pairs of examples and asks whether they both satisfy some term in common a DNF target function. We realize this type of richer interaction than label requests can be extremely powerful in solving classic computational learning problems. In particular, we consider the problem dating back to (Valiant, 1984) of learning DNF formulas, for which no known efficient algorithms exist in the PAC model. A DNF formula can be thought of as describing a concept for which there are m different "types" of positive examples, one for each term. Suppose, then, that a user identifies m examples as "landmarks" that each satisfy a different unique term in the formula. Furthermore, suppose that if the algorithm makes a mistake on some example x — say predicting negative when the true label is positive — the user is able to indicate which landmark L_i is most similar, meaning that x and L_i both satisfy the same term. Then, this additional feedback will decompose the DNF-learning problem into m separate conjunction-learning problems, making the algorithm's task quite easy. Of course, the above is an extreme idealization. Natural further questions we might want to look at include:

- Suppose for each pair of positive examples, you are told the number of terms they satisfy in common. This could be 0 or larger. Is this enough to learn? In the case of a Decision Tree (or more generally a disjoint-DNF, which is a DNF in which it is impossible to satisfy more than one term) then learning becomes trivial because it is the same as saying which term is satisfied. But what about general DNF formulas?
- If we are able to solve the first bullet, then one weaker version would be what if we just get binary yes/no information about whether any given pair of positive examples satisfies a term in common or not.
- If we are able to solve the first bullet, then a different natural extension would be: what if we don't get the pairwise information for every pair of positive examples up front, but we explicitly have to ask. Do we really need to ask n^2 questions (all pairs) or can we get by with fewer?

To this point, we have an efficient algorithm for the case when the target DNF has a minimal representation (number of terms) consistent with the given data set. We have also explored more powerful types of queries, which allow us to remove this "minimal representation" restriction. Specifically, we have an efficient algorithm for learning general DNF (even non-minimal targets) if we are allowed to construct the feature vectors given to the pairwise queries ourselves; this is somewhat analogous to the ability to addition of membership queries to the classic PAC model. Furthermore, we have an efficient algorithm for learning general DNF (again, even non-minimal targets) if we are allowed to ask these queries for general subsets of the pool: in other words, if we can ask whether there exists a term in the target DNF satisfied by all examples in an arbitrary subset of the data (not just pairs), where the data are randomly sampled. We are still studying the case where the data are randomly drawn and we are only permitted queries for pairs of examples, for the case where the target DNF does not have a minimal representation.

5.3 Online Active Learning under Distribution Drift

Often learning systems must track changes over time. If these changes are gradual, for instance, for a given user, while a spam email is likely still a spam email if it is received a month or year later, the nature of *typical* spam (and typical non-spam) may well change substantially. Even though this only changes the distribution and not the concept, if unnoticed it can still lower accuracy of detection. We model the learning process under a drifting distribution of the feature space. We specifically study this problem in the context of online active learning. We may have a stream of observations, and the learner needs to make predictions for each example, and then may decide to request its label or not. Then we analyze the number of mistakes made by such an algorithm.

Most existing analyses of active learning algorithms are based on an i.i.d. assumption on the data. While our data are also independent, we do allow their distribution to shift over time. The existing shifting model (Bartlett, 1992) assumes the total average changing in the density function is bounded by a constant. However, this model gives so much flexibility to the adversary choosing the distributions that the error rate of the learning algorithm might never converge.

To slightly restrict the adversary, we impose an assumption that the adversary may only choose its distributions from among a known totally bounded (in total variation distance) family of distributions $\mathbb{D}$. For instance, this would be the case for the family of distributions with density functions having modulus of continuity bounded by a given function.

At this time, we have studied the classic CAL active learning algorithm (Cohn et al., 1994) in this context, for the realizable case. We have identified sufficient conditions for the number of queries among the initial T examples to be $o(T)$, while maintaining the same number of mistakes as passive learning (also $o(T)$). Specifically, this will be the case for any VC class if $\sup_{D \in \mathbb{D}} \mathbb{P}(\lim_{r \to 0} DIS(B(h^*, r))) < \infty$, where $DIS(V) = \{x : \exists h, g \in V \text{ s.t. } h(x) \neq g(x)\}$ is the region of disagreement, and $B(h^*, r) = \{h \in \mathbb{C} : \mathbb{P}(h(x) \neq h^*(x)) \leq r\}$. For instance, this is the case if the disagreement coefficients are uniformly bounded (Hanneke, 2009). We can additionally state specific bounds on this $o(T)$ rate under stronger conditions on the covering numbers of $\mathbb{D}$.

Moving forward, we would like to extend these results to settings with noise, including general benign noise settings, such as Tsybakov noise, and the general agnostic setting with arbitrary forms of noise. We would also like to extend this $o(T)$ result to more general scenarios, even where the stated condition on the limiting regions of disagreement does not hold. For instance, we may be able to achieve this via the shattering-based approach of (Hanneke, 2009), modified in some way to suit this stream-based setting.

6 Tentative Timeline for Completing the Work

Given the intrinsic uncertainty of theoretical research, it is hard to predict which of the set of open research problems mentioned in Section 5 will be solved in the future, but I intend to solve a nonempty subset of it. It might take another one year or two to complete the thesis. A tentative timeline is to have major results worked out on active property testing this spring semester. Then my major effort will be put on efficient learning with extra information and active learning under distribution drift over the summer and the fall semester.

Bibliography

Ando, R. K., & Zhang, T. (2004). *A framework for learning predictive structures from multiple tasks and unlabeled data* (Technical Report RC23462). IBM T.J. Watson Research Center.

Ash, R. B., & Doléans-Dade, C. A. (2000). *Probability & measure theory*. Academic Press.

Balcan, M.-F., Beygelzimer, A., & Langford, J. (2009). Agnostic active learning. *Journal of Computer and System Sciences*, *75*, 78–89.

Balcan, M.-F., Broder, A., & Zhang, T. (2007). Margin based active learning. *Proceedings of the 20th Conference on Learning Theory*.

Balcan, M.-F., Hanneke, S., & Vaughan, J. W. (2010). The true sample complexity of active learning. *Machine Learning*, *80*, 111–139.

Bartlett, P. L. (1992). Learning with a slowly changing distribution. *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 243–252).

Baxter, J. (1997). A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, *28*, 7–39.

Baxter, J. (2000). A model of inductive bias learning. *Journal of Artificial Intelligence Research*, *12*, 149–198.

Ben-David, S., & Schuller, R. (2003). Exploiting task relatedness for multiple task learning. *Conference on Learning Theory*.

Beygelzimer, A., Dasgupta, S., & Langford, J. (2009). Importance weighted active learning. *Proceedings of the International Conference on Machine Learning*.

Bshouty, N. H., Li, Y., & Long, P. M. (2009). Using the doubling dimension to analyze the generalization of learning algorithms. *Journal of Computer and System Sciences*, *75*, 323–335.

Carbonell, J. G. (1983). Learning by analogy: Formulating and generalizing plans from past experience. *Machine Learning, An Artificial Intelligence Approach.* Tioga Press, Palo Alto, CA.

Carbonell, J. G. (1986). Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. *Machine Learning, An Artificial Intelligence Approach, Volume II.* Morgan Kaufmann.

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*, 41–75.

Castro, R., & Nowak, R. (2008). Minimax bounds for active learning. *IEEE Transactions on Information Theory*, *54*, 2339–2353.

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, *15*, 201–221.

Dasgupta, S. (2004). Analysis of a greedy active learning strategy. *Advances in Neural Information Processing Systems* (pp. 337–344). MIT Press.

Dasgupta, S. (2005). Coarse sample complexity bounds for active learning. *In Advances in Neural Information Processing Systems 18*.

Dasgupta, S., Hsu, D., & Monteleoni, C. (2007). A general agnostic active learning algorithm. *Advances in Neural Information Processing Systems 20*.

Dasgupta, S., Kalai, A. T., & Monteleoni, C. (2009). Analysis of perceptron-based active learning. *Journal of Machine Learning Research*, *10*, 281–299.

Devroye, L., & Lugosi, G. (2001). *Combinatorial methods in density estimation*. New York, NY, USA: Springer.

Evgeniou, T., Micchelli, C., & Pontil, M. (2005). Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, *6*, 615–637.

Evgeniou, T., & Pontil, M. (2004). Regularized multi-task learning. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, *28*, 133–168.

Friedman, E. (2009). Active learning for smooth problems. *Proceedings of the 22^{nd} Conference on Learning Theory*.

Goldreich, O. (Ed.). (2010). *Property testing - current research and surveys [outgrow of a workshop at the institute for computer science (itcs) at tsinghua university, january 2010]*, vol. 6390 of *Lecture Notes in Computer Science*. Springer.

Gupta, A., Krauthgamer, R., & Lee, J. R. (2003). Bounded geometries, fractals, and low-distortion embeddings. *In Proceedings of the 44^{th} Annual IEEE Symposium on Foundations of Computer Science*.

Hanneke, S. (2007a). A bound on the label complexity of agnostic active learning. *In Proceedings of the 24^{th} International Conference on Machine Learning*.

Hanneke, S. (2007b). Teaching dimension and the complexity of active learning. *In Proceedings of the 20^{th} Annual Conference on Learning Theory*.

Hanneke, S. (2009). *Theoretical foundations of active learning*. Doctoral dissertation, Carnegie Mellon University.

Hanneke, S. (2011). Rates of convergence in active learning. *The Annals of Statistics*, *39*, 333–361.

Haussler, D., Kearns, M., & Schapire, R. (1994a). Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. *Machine Learning*, *14*, 83–113.

Haussler, D., Littlestone, N., & Warmuth, M. (1994b). Predicting $\{0, 1\}$-functions on randomly drawn points. *Information and Computation*, *115*, 248–292.

Kääriäinen, M. (2006). Active learning in the non-realizable case. *In Proc. of the 17th International Conference on Algorithmic Learning Theory*.

Kolodner (Ed), J. (1993). *Case-based learning*. Kluwer Academic Publishers, The Netherlands.

Koltchinskii, V. (2010). Rademacher complexities and bounding the excess risk in active learning. *Journal of Machine Learning Research*, *To Appear*.

Kulkarni, S. R., Mitter, S. K., & Tsitsiklis, J. N. (1993). Active learning using arbitrary binary valued queries. *Machine Learning*, *11*, 23–35.

Micchelli, C., & Pontil, M. (2004). Kernels for multi–task learning. *Advances in Neural Information Processing 18*.

Nowak, R. D. (2008). Generalized binary search. *Proceedings of the 46^{th} Annual Allerton Conference on Communication, Control, and Computing*.

Parnas, M., Ron, D., & Samorodnitsky, A. (2002). Testing basic boolean formulae. *SIAM J. Discrete Math.*, *16*, 20–46.

Ron, D. (2008). Property testing: a learning theory perspective. *Founda- tions and Trends in Machine Learning*, *1*, 307–402.

Ron, D. (2009). Algorithmic and analysis techniques in property testing. *Foundations and Trends in Theoretical Computer Science*, *5*, 73–205.

Schervish, M. J. (1995). *Theory of statistics*. New York, NY, USA: Springer.

Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. *In Proceedings of the 5^{th} Workshop on Computational Learning Theory*.

Silver, D. L. (2000). *Selective transfer of neural network task knowledge*. Doctoral dissertation, Computer Science, University of Western Ontario.

Thrun, S. (1996). Is learning the n-th thing any easier than learning the first? *In Advances in Neural Information Processing Systems 8*.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, *27*, 11341142.

Vapnik, V. (1982). *Estimation of dependencies based on empirical data*. Springer-Verlag, New York.

Veloso, M. M., & Carbonell, J. G. (1993). Derivational analogy in prodigy: Automating case acquisition, storage and utilization. *Machine Learning*, *10*, 249–278.

Wang, L. (2009). Sufficient conditions for agnostic active learnable. *Advances in Neural Information Processing Systems 22*.

Yang, L., Hanneke, S., & Carbonell, J. (2010). Bayesian active learning using arbitrary binary valued queries. *Proceedings of the 21st International Conference on Algorithmic Learning Theory (ALT)*.

Yang, L., Hanneke, S., & Carbonell, J. (2011). The sample complexity of self-verifying bayesian active learning. *Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*.