A PAC-style Model for Learning from Labeled and Unlabeled Data

Maria-Florina Balcan* Avrim Blum*

April 22, 2004
(DRAFT)

Abstract

There has recently been substantial interest in practice in using unlabeled data together with labeled data in machine learning, and a number of different approaches have been developed. However, the assumptions these methods are based on are often quite distinct and not captured by standard theoretical models. In this paper we describe a PAC-style model that captures many of these assumptions, and analyze sample-complexity issues in this setting: that is, how much of each type of data one should expect to need in order to learn well, and what are the basic quantities that these numbers depend on. Our model can be viewed as an extension of the standard PAC model, where in addition to a concept class C, one also proposes a type of compatibility that one believes the target concept should have with the underlying distribution. In this view, unlabeled data can be helpful because it allows one to estimate compatibility over the space of hypotheses, and reduce the size of the search space to those that are, in a sense, a-priori reasonable with respect to the distribution. We discuss a number of technical issues that arise in this context, and provide sample-complexity bounds both for uniform convergence and specific ϵ-cover based algorithms. We also consider algorithmic issues, and give an algorithm for a natural problem of learning a linear separator from example-pairs, motivated by co-training.

1 Introduction

There has recently been substantial interest in using unlabeled data together with labeled data for machine learning. The motivation is that unlabeled data can often be much cheaper and more plentiful than labeled data, and so if useful information can be extracted from it that reduces dependence on labeled examples, this can be a significant benefit. A number of techniques have been developed for doing this, along with experimental results on a variety of different learning problems. These include label propagation for word-sense disambiguation [25]; co-training for classifying web pages [5], parsing [14], and improving visual detectors [18]; transductive SVM [15] and EM [20] for text classification; graph-based methods [26], and others. In fact, the problem of learning from labeled and unlabeled data was the topic of a workshop at the most recent ICML conference [13].

The difficulty from a theoretical point of view, however, is that standard discriminative learning models do not really capture how and why unlabeled data can be of help. In particular, in the PAC model there is a complete disconnect between the data distribution D and the target function f being learned [6, 22]. The only prior belief is that f belongs to some class

*Department of Computer Science, Carnegie Mellon University. {ninamf,avrim}@cs.cmu.edu

$\mathcal{C}$: even if D is known fully, any function $f \in \mathcal{C}$ is still possible. For instance, it is perfectly natural (and common) to talk about, say, the problem of learning AC^0 circuits [19] or an intersection of halfspaces [4, 24, 17] over the uniform distribution; but clearly in this case unlabeled data is useless — you can just generate it yourself. For learning over an unknown distribution (the standard PAC setting), unlabeled data can help somewhat, by allowing one to use distribution-specific sample-complexity bounds, but this does not seem to fully capture the power of unlabeled data in practice.

In *generative*-model settings, one *can* easily talk about the use of unlabeled data, e.g., [7, 8]. However, these results make strong assumptions that imply that the unlabeled distribution in essence is committing to the target function or its complement as the correct answer, and the only use of labeled data then is to decide which is which. For instance, a typical generative-model setting would be that we assume positive examples are generated by one gaussian, and negative examples are generated by another gaussian: in this case, given enough unlabeled data, we could recover the gaussians and would need labeled data only to tell us which gaussian is which.[1] Instead, we would like our model to allow for a distribution over data (e.g., documents we want to classify) where there are a number of plausible distinctions we might want to make.

The goal of this paper is to provide a PAC-style framework that bridges between these positions and captures many of the ways unlabeled data is typically used. We extend the PAC model in a way that allows one to express relationships that one hopes the target function and underlying distribution will possess, but without going so far as is done in generative models. We then analyze sample-complexity issues in this setting: that is, how much of each type of data one should expect to need in order to learn well, and what are the basic quantities that these numbers depend on. We also give algorithmic results for a natural problem of learning a linear separator from pairs of examples, motivated by co-training.

The high-level idea of the proposed model is to augment the notion of a *concept class*, which is a set of functions (like linear separators or decision trees), with a notion of *compatibility* between a target function and the data distribution. That is, rather than talking of "learning a concept class $\mathcal{C}$," we would talk of "learning a concept class $\mathcal{C}$ under compatibility notion χ." Furthermore, we require that the degree of compatibility be something that can be estimated from a finite sample: specifically, χ is actually a function from $\mathcal{C} \times X$ to $[0, 1]$, where the compatibility of h with D is $\mathbf{E}_{x \in D}[\chi(h, x)]$. The degree of *in*compatibility is then something we can think of as a kind of "unlabeled error rate". For example,

Example 1 (margins): Suppose examples are points in R^n and $\mathcal{C}$ is the class of linear separators. A natural belief in this setting is that data should be "well-separated": not only should the target function separate the positive and negative examples, but it should do so by some reasonable *margin* γ. This is the assumption used by Transductive SVM [15]. In this case, we could define $\chi(h, x) = 1$ if x is farther than distance γ from the hyperplane defined by h, and $\chi(h, x) = 0$ otherwise. So, the incompatibility of h with D is probability mass within distance γ of $h \cdot x = 0$. Or we could define $\chi(h, x)$ to be a smooth function of the distance of x to the separator, if we do not want to commit to a specific γ in advance. (In contrast, defining compatibility of based on the largest γ such that D has probability mass *exactly zero* within distance γ of the separator would *not* fit our model: it cannot be written as an expectation over individual examples and indeed one cannot distinguish "zero" from "exponentially close to zero" with a small sample.)

[1] Castelli and Cover [7, 8] do not assume gaussians in particular, but they do assume the distributions are distinguishable, which from our perspective has the same issue.

Example 2 (Co-training): In co-training [5], we assume examples come as pairs $\langle x_1, x_2 \rangle$, and our goal is to learn a pair of functions $\langle h_1, h_2 \rangle$. For instance, if our goal is to classify web pages, x_1 might represent the words on the page itself and x_2 the words attached to links pointing *to* this page from other pages. The hope that underlies co-training is that the two parts of the example are consistent, which then allows the co-training algorithm to bootstrap from unlabeled data.[2] In this case, we might naturally define the incompatibility of some hypothesis $\langle h_1, h_2 \rangle$ as $\mathbf{Pr}_{\langle x_1, x_2 \rangle \in D}[h_1(x_1) \neq h_2(x_2)]$.

Example 3 (Linear separator graph cuts): As a special case of Example 2 above, suppose examples are *pairs* of points in R^n, $\mathcal{C}$ is the class of linear separators, and we believe the two points in each pair should both be on the *same* side of the target function (i.e., like co-training but we are requiring $h_1 = h_2$).[3] Again we can define the incompatibility of some h to be the probability mass on examples $\langle x_1, x_2 \rangle$ such that $h(x_1) \neq h(x_2)$. What makes this problem interesting is that we can view examples as edges, view the data as a graph embedded in R^n, and given a set of labeled and unlabeled data, view our objective as finding a linear separator minimum s-t cut.

This setup allows us to analyze the ability of a finite unlabeled sample to reduce our dependence on labeled examples, as a function of the compatibility of the target function and various measures of the "helpfulness" of the distribution. In particular, in our model, we find that unlabeled data can help in several distinct ways.

- If we assume (or hope) the target function is highly compatible with D, then if we have enough unlabeled data to estimate compatibility over all $h \in \mathcal{C}$, we can in principle reduce the size of the search space from $\mathcal{C}$ down to just those $h \in \mathcal{C}$ whose estimated unlabeled error rate is low.
- By providing an estimate of D, unlabeled data can allow us to use a more refined distribution-specific notion of "hypothesis space size" such as Annealed VC-entropy [9] or the size of the smallest ϵ-cover [2], rather than VC-dimension. In fact, for natural cases (such as those above) we find that the sense in which unlabeled data reduces the "size" of the search space is best described in these distribution-specific measures.
- Finally, if the distribution is especially nice, we may find that not only does the set of "compatible" $h \in C$ have a small ϵ-cover, but also the elements of the cover are far apart. In that case, if we assume the target function is fully compatible, we may be able to learn from even fewer labeled examples than the $1/\epsilon$ needed just to *verify* a good hypothesis! (Though here D is effectively committing to the target as in generative models.)

Our framework also allows us to address the issue of how much *unlabeled* data we should expect to need. In the end, we are using unlabeled data primarily to estimate compatibility of each $h \in C$, so the "VCdim/ϵ^2" form of standard PAC sample complexity bounds now becomes

[2] The co-training algorithm trains two learning algorithms, one for each half. It uses a small amount of labeled data to get some initial information (e.g., it might learn that if a link with the words "my advisor" points to a page then that page is probably a faculty member's home page) and then when it finds an unlabeled example where one half is confident (e.g., the link says "my advisor"), it uses that to label the example and give it to the other learning algorithm.

[3] As a motivating example, consider the problem of *word-sense disambiguation*: given the text surrounding some target word (like "plant") we want to determine which dictionary definition is intended (tree or factory?). Yarowsky [25] uses the fact that if a word appears twice in the same document, it is probably being used in the *same* sense both times. If we consider hypotheses that assign real-valued weights to context words and compute a weighted vote over words in the surrounding text, this then corresponds to finding a linear separator.

a bound on the number of *unlabeled* examples needed to perform this estimation. However, technically, the set whose VC-dimension we now care about is not C but rather a set defined by both C and χ: that is, the overall complexity depends both on the complexity of C and the complexity of the notion of compatibility (see Section 5).

Relationship to the luckiness framework. There is a strong connection between our approach and the luckiness framework [21]. In both cases, the idea is to define an ordering of hypotheses that depends on the data, in the hope that we will be "lucky" and find that not too many other functions are as compatible as the target. There are two main differences, however. The first is that the luckiness framework uses labeled data both for estimating compatibility and for learning: this is a more difficult task, and as a result our bounds on labeled data can be significantly better. For instance, in Example 3 above, for any non-degenerate distribution, a dataset of $\leq n/2$ pairs can be completely shattered by fully-compatible hypotheses, so the luckiness framework does not help. In contrast, with a larger (unlabeled) sample, one can potentially reduce the space of compatible functions quite significantly – see Section 4 and 6. Secondly, the luckiness framework talks about compatibility between a hypothesis and a *sample*, whereas we define compatibility with respect to a distribution. This allows us to talk about the amount of unlabeled data needed to estimate compatibility. There are also a number of differences at the technical level of the definitions.

Outline of results. We give results both for *sample complexity* (in principle, how much data is needed to learn) and efficient algorithms. In terms of sample-complexity, in Section 3, we give the simplest version of our results, for the case of finite hypothesis spaces. For infinite hypothesis spaces, VC-dimension is generally not a good way to measure the "size" of the set of compatible functions: even if all compatible functions are very similar, their VC-dimension may be as high as for the entire class. So, we instead in Section 5 we use a distribution-specific analog, the expected number of partitions of a sample drawn from D, that in many natural cases *does* decrease substantially (and can be estimated from the unlabeled dataset if desired).

These results give bounds on the number of examples needed for any learning algorithm that produces a compatible hypothesis of low empirical error. To achieve tighter bounds, in Section 6 we give results based on the notion of ϵ-cover size. These bounds hold for algorithms of a very specific type, that first use the unlabeled data to choose a small set of "representative" hypotheses (every compatible $h \in \mathcal{C}$ is close to at least one of them), and then choose among the representatives based on the labeled data. We point out that this can yield bounds substantially better than with uniform convergence (e.g., we can learn even though there will whp exist bad $h \in \mathcal{C}$ consistent with the labeled and unlabeled samples).

In Section 4, we give our algorithmic results. We begin with a particularly simple $\mathcal{C}$ and χ for illustration, and then give our main algorithmic result: an efficient algorithm for the case of Example 3, if we assume both elements of the pair are drawn independently given the label. In the process, we get a simplification to the noisy halfspace learning algorithm of [3].

2 A Formal Framework

We assume that examples (both labeled and unlabeled) come according to a fixed unknown distribution D over an instance space X, and they are labeled by some unknown target function c^*. As in the standard PAC model, a *concept class* or *hypothesis space* is a set of functions over the instance space X, and we will often make the assumption (the "realizable case") that the

target function belongs to a given class $\mathcal{C}$. For a given hypothesis h, the (true) error rate of h is defined as $err(h) = err_D(h) = \mathbf{Pr}_{x \in D}[h(x) \neq c^*(x)]$. For any two hypotheses $h_1, h_2 \in C$, the distance with respect to D between h_1 and h_2 is defined as $d(h_1, h_2) = d_D(h_1, h_2) = \mathbf{Pr}_{x \in D}[h_1(x) \neq h_2(x)]$. We will use $\widehat{err}(h)$ to denote the empirical error rate of h on a given labeled sample and $\hat{d}(h_1, h_2)$ to denote the empirical distance between h_1 and h_2 on a given unlabeled sample.

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

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

Definition 2 *Given compatibility notion χ, the* incompatibility *of h with D is $1 - \chi(h, D)$. We will also call this its* unlabeled error rate, *$err_{unl}(h)$, when χ and D are clear from context. For a given sample S, we use $\widehat{err}_{unl}(h)$ to denote the empirical average over S.*

Finally, we need a notation for the set of functions whose incompatibility is at most some given value τ.

Definition 3 *Given threshold τ, we define $\mathcal{C}_{D,\chi}(\tau) = \{h \in \mathcal{C} : err_{unl}(h) \leq \tau\}$. So, e.g., $\mathcal{C}_{D,\chi}(1) = \mathcal{C}$. Similarly, for a sample S, we define $\mathcal{C}_{S,\chi}(\tau) = \{h \in \mathcal{C} : \widehat{err}_{unl}(h) \leq \tau\}$*

For simplicity, and to reduce notation, we will assume in the rest of this paper that $\chi(h, x) \in \{0, 1\}$ so that $\chi(h, D) = \mathbf{Pr}_{x \in D}[\chi(h, x) = 1]$.

The main high-level implication in this framework is that unlabeled data can decrease the number of labeled examples needed if (a) the target function indeed has a low unlabeled error rate, (b) the distribution D is *helpful* in the sense that not too many other hypotheses also have a low unlabeled error rate, and (c) we have enough *unlabeled* data to estimate unlabeled error rates well.

3 Finite hypothesis spaces

We begin with some immediate sample-complexity bounds, for the case where we measure the "size" of a set of function by just the number of functions in the set. In the standard PAC model, one typically talks of either the realizable case, where we assume that $c^* \in \mathcal{C}$, or the agnostic case where we do not. In our setting, we have the additional issue of *unlabeled* error rate, and can either make an a-priori assumption that the target function's unlabeled error is low, or else aim for a more Occam-style bound in which we have a stream of labeled examples and halt once they are sufficient to justify the hypothesis produced. For simplicity, we first give a bound for the "doubly realizable" case.

Theorem 1 *If we see*

$$m_u \geq \frac{1}{\epsilon}\left[\ln|\mathcal{C}| + \ln\frac{2}{\delta}\right]$$

unlabeled examples and

$$m_l \geq \frac{1}{\epsilon}\left[\ln|\mathcal{C}_{D,\chi}(\epsilon)| + \ln\frac{2}{\delta}\right]$$

labeled examples, then with probability $\geq 1-\delta$, *all* $h \in \mathcal{C}$ *with* $\widehat{err}(h) = 0$ *and* $\widehat{err}_{unl}(h) = 0$ *have* $err(h) \leq \epsilon$.

Proof: The number of unlabeled examples is sufficient to ensure that with probability $1-\delta/2$, the set of hypotheses with $\widehat{err}_{unl}(h) = 0$ is contained in $\mathcal{C}_{D,\chi}(\epsilon)$. The number of labeled examples then ensures that with probability $1-\delta/2$, none of those whose true (labeled) error is at least ϵ have an empirical (labeled) error of 0. ■

So, if the target function indeed has zero labeled and unlabeled error, Theorem 1 gives sufficient conditions on the number of examples needed to ensure that an algorithm that optimizes both quantities over the observed data will, in fact, achieve a PAC guarantee. To emphasize this, we will say that an algorithm PAC$_{unl}$-learns the pair $(\mathcal{C}, \chi)$ if it is able to achieve a PAC guarantee using sample sizes polynomial in the bounds of Theorem 1.

We can think of theorem 1 as bounding the number of labeled examples we need as a function of the "helpfulness" of the distribution D with respect to our notion of compatibility. That is, in our context, a helpful distribution is one in which $\mathcal{C}_{D,\chi}(\epsilon)$ is small, and so we do not need much labeled data to identify a good function among them. We can get a similar bound in the situation when the target function is not fully compatible:

Theorem 2 *Given* $t \in [0,1]$, *if we see*

$$m_u \geq \frac{2}{\epsilon^2}\left[\ln|\mathcal{C}| + \ln\frac{4}{\delta}\right]$$

unlabeled examples and

$$m_l \geq \frac{1}{\epsilon}\left[\ln|\mathcal{C}_{D,\chi}(t+2\epsilon)| + \ln\frac{2}{\delta}\right]$$

labeled examples, then with probability $\geq 1-\delta$, *all* $h \in \mathcal{C}$ *with* $\widehat{err}(h) = 0$ *and* $\widehat{err}_{unl}(h) \leq t+\epsilon$ *have* $err(h) \leq \epsilon$, *and furthermore all* $h \in \mathcal{C}$ *with* $err_{unl}(h) \leq t$ *have* $\widehat{err}_{unl}(h) \leq t+\epsilon$.

Proof: Same as Theorem 1 except apply Hoeffding bounds to the unlabeled error rates. ■

In particular, this implies that if $err_{unl}(c^*) \leq t$ and $err(c^*) = 0$ then with probability $\geq 1-\delta$ the $h \in \mathcal{C}$ that optimizes $\widehat{err}(h)$ and $\widehat{err}_{unl}(h)$ has $err(h) \leq \epsilon$.

Finally, we give a simple Occam/luckiness type of bound for this setting. Given a sample S, let us define $\mathsf{desc}_S(h) = \ln|\mathcal{C}_{S,\chi}(\widehat{err}_{unl}(h))|$. That is, $\mathsf{desc}_S(h)$ is the description length of h (in "nats") if we sort hypotheses by their empirical compatibility and output the index of h in this ordering. Similarly, define $\epsilon\text{-}\mathsf{desc}_D(h) = \ln|\mathcal{C}_{D,\chi}(err_{unl}(h)+\epsilon)|$. This is an upper-bound on the description length of h if we sort hypotheses by an ϵ-approximation to the their true compatibility.

Theorem 3 *For any set* S *of unlabeled data, given* m_l *labeled examples, with probability* $1-\delta$, all $h \in \mathcal{C}$ *satisfying* $\widehat{err}(h) = 0$ *and* $\mathsf{desc}_S(h) \leq \epsilon m_l - \ln(1/\delta)$ *have* $err(h) \leq \epsilon$. *Furthermore, if* $|S| \geq \frac{2}{\epsilon^2}[\ln|\mathcal{C}| + \ln\frac{2}{\delta}]$, *then with probability* $1-\delta$, *all* $h \in \mathcal{C}$ *satisfy* $\mathsf{desc}_S(h) \leq \epsilon\text{-}\mathsf{desc}_D(h)$.

The point of this theorem is that an algorithm can use observable quantities to determine if it can be confident, and furthermore if we have enough unlabeled data, the observable quantities will be no worse than if we were learning a slightly less compatible function using an infinite-size unlabeled sample.

4 Algorithmic results

4.1 A simple computational example

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

Given this setup, we can give a simple PAC$_{unl}$-learning algorithm for this pair $(\mathcal{C}, \chi)$. We begin by using our unlabeled data to construct a graph on n vertices (one per variable), putting an edge between two vertices i and j if there is any example x in our unlabeled sample with $i, j \in \mathsf{vars}(x)$. We now use our labeled data to label the connected components. If the target function is fully compatible, then no component will get both positive and negative labels. Finally, we produce the hypothesis h such that $\mathsf{vars}(h)$ is the union of the positively-labeled components. This is fully compatible with the unlabeled data and has zero error on the labeled data, so by Theorem 1, if the sizes of the datasets are as given in the bounds, with high probability the hypothesis produced will have error $\leq \epsilon$.

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

Also, it is interesting to see the difference between a "helpful" and "non-helpful" distribution for this problem. An especially non-helpful distribution would be the uniform distribution over all examples x with $|\mathsf{vars}(x)| = 1$, in which there are n components. In this case, unlabeled data does not help at all, and one still needs $\Omega(n)$ labeled examples (or, even $\Omega(n/\epsilon)$ if the distribution is a non-uniform as in VC-dimension lower bounds [11]). On the other hand, a helpful distribution would be one such that with high probability the number of components is small, leading to a lower number of labeled examples.

4.2 Learning linear separators from example pairs

We now consider the case of Example 3: the target function is a linear separator in R^n and each example is a *pair* of points both of which are assumed to be on the same side of the separator (i.e., an example is a line-segment that does not cross the target plane). Given a set of labeled and unlabeled data, our goal is to find a separator that is consistent with the labeled examples and compatible with the unlabeled ones. (Even though $\mathcal{C}$ is infinite, this would then with high probability have low true error by the bounds in Section 5.)

Unfortunately, the consistency problem for this task is NP-hard: given a graph G embedded in R^n with two distinguished points s and t, it is NP-hard to find the linear separator that cuts the minimum number of edges, *even if the minimum is 0* [12]. For this reason, we will make an additional assumption, that the two points in an example are each drawn independently given the label. That is, there is a single distribution D over R^n, and with some probability p, two points are drawn iid from D_+ (D restricted to the positive side of the target function) and with

probability $1-p$, the two are drawn iid from D_- (D restricted to the negative side of the target function).

Blum and Mitchell [5] give positive algorithmic results for co-training when (a) the two halves of an example are drawn independently given the label (which we are assuming now), (b) the underlying function is learnable via Statistical Query algorithms (which is true for linear separators by [3]), and (c) we have enough labeled data to produce a weakly-useful hypothesis on one of the halves to begin with.[4] Our contribution in this paper is to show how we can run that algorithm with only *a single labeled example*. In the process, we simplify the results of [3].

Theorem 4 *There is a polynomial-time algorithm to learn a linear separator under the above assumptions, from a polynomial number of unlabeled examples and a single labeled example.*

Proof Sketch: Assume for convenience that the target separator passes through the origin, and let us denote the separator by $c^* \cdot x = 0$. We will also assume for convenience that $\mathbf{Pr}_D(c^*(x) = 1) \in [\epsilon/2, 1-\epsilon/2]$; that is, the target function is not overwhelmingly positive or overwhelmingly negative (if it is, this is actually an easy case, but it makes the arguments more complicated). Define the *margin* of some point x as the distance of $x/|x|$ to the separating plane, or equivalently, the cosine of the angle between c^* and x.

We begin by drawing a large unlabeled sample S. The first step is to ensure that some reasonable $(1/poly)$ fraction of S has margin at least $1/poly$, which we can do via the Outlier Removal Lemma of [3, 10]. The Outlier Removal Lemma states that one can algorithmically remove an ϵ' fraction of S and ensure that for the remainder, for any vector w, $\max_{x \in S}(w \cdot x)^2 \leq poly(n, b, 1/\epsilon')\mathbf{E}_{x \in S}[(w \cdot x)^2]$, where b is the number of bits needed to describe the input points. We can now perform a linear transformation to put the points into isotropic position (for all unit-length w, $\mathbf{E}[(w \cdot x)^2] = 1$), which then guarantees that (since the maximum is bounded), at least a $1/poly$ fraction of the points have at least a $1/poly$ distance to the separator.

The second step is we argue that a *random* halfspace has at least a $1/poly$ chance of being a weak predictor. ([3] uses the perceptron algorithm to get weak learning; here, we need something simpler since we do not yet have any labeled data.) Specifically, consider some positive point x such that the angle between x and c^* is $\pi/2 - \gamma$, and imagine that we draw h at random subject to $h \cdot c^* \geq 0$ (half of the h's will have this property). Then we have:

$$\mathbf{Pr}_h(h \cdot x < 0 | h \cdot c^* \geq 0) = (\pi/2 - \gamma)/\pi = 1/2 - \gamma/\pi.$$

Since we have that at least a $1/poly$ fraction of the positive $x \in S$ have margin at least $1/poly$, this means that

$$\mathbf{Pr}_{h,x}(h \cdot x < 0 | h \cdot c^* \geq 0, x \cdot c^* \geq 0) < 1/2 - 1/poly,$$

and similarly for the negative examples with signs reversed. This means that a $1/poly$ probability mass of functions h must in fact be weakly-useful predictors.

The final step of the algorithm is as follows. Using the above observation, we pick a random h, and plug it into the bootstrapping theorem of [5] (which, given unlabeled pairs $\langle x, x' \rangle$, will use $h(x)$ as a noisy label of x', feeding the result into an SQ algorithm), repeating this process $poly(n)$ times. With high probability, our random h was a weakly-useful predictor on at least one of these steps, and we end up with a low-error hypothesis. For the rest of

[4] A weakly-useful predictor is defined to be a predictor h such that $\mathbf{Pr}[h(x) = 1 | c^*(x) = 1] > \mathbf{Pr}[h(x) = 1 | c^*(x) = 0] + \epsilon$; it is equivalent to the usual notion of a "weak hypothesis" when the target function is balanced, but requires the hypothesis give more information when the target function is unbalanced. Kalai and Servedio [16] show that this property is also sufficient for boosting in the presence of noise.

the runs of the algorithm, we have no guarantees. We now observe the following. First of all, any function h with small $err(h)$ must have small $err_{unl}(h)$. Secondly, because of the assumption of independence given the label, the *only* functions with low unlabeled error rate are functions close to c^*, close to $\neg c^*$, close to the "all positive" function, or close to the "all negative" function.[5] That is because if h is not close to one of these, it must be the case that for some $\ell \in \{0,1\}$, $\mathbf{Pr}_x(h(x) = \ell | c^*(x) = \ell) \in [\alpha, 1-\alpha]$ for $\alpha = \Omega(\epsilon)$, which implies $\mathbf{Pr}_{x,x'}(h(x) \neq h(x') | c^*(x) = c^*(x') = \ell) \geq \alpha(1-\alpha)$. So, if we simply examine all the hypotheses produced by this procedure, and pick some h with a low unlabeled error rate that is at least $\epsilon/2$-far from the "all-positive" or "all-negative" functions, then either h or $\neg h$ is close to c^*. We can now just draw a single labeled example to determine which case is which. ■

5 Infinite hypothesis spaces: uniform convergence bounds

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

A second issue is that even if we assume the target function is fully compatible and the distribution is helpful, VC-dimension may not be a good way to measure the "size" of the set of surviving functions. For instance, if we consider a margin-based notion of compatibility (the case of Example 1), then even if data is concentrated in two well-separated "blobs", the set of compatible functions still has as large a VC-dimension as for the entire class, even if all compatible functions are very similar with respect to D. So, instead of VC-dimension, a better measure is the *expected* number of splits of a sample of size m drawn from D (its logarithm is sometimes called "annealed VC-entropy"). Specifically, for any $\mathcal{C}$, we denote by $\mathcal{C}[m, D]$ the expected number of splits of m points (drawn i.i.d.) from D with concepts in $\mathcal{C}$. Also, for a given (fixed) $S \subseteq X$, we will denote by $\overline{S}$ the uniform distribution over S, and by $\mathcal{C}\left[m, \overline{S}\right]$ the expected number of splits of m points (drawn i.i.d.) from $\overline{S}$ with concepts in $\mathcal{C}$. We can now get a bound as follows:

Theorem 5

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

unlabeled examples and

$$m_l > \frac{2}{\epsilon}\left[\log(2s) + \log\frac{2}{\delta}\right]$$

labeled examples, where

$$s = \mathcal{C}_{D,\chi}(t + 2\epsilon)[2m_l, D]$$

are sufficient so that with probability $\geq 1-\delta$, all $h \in \mathcal{C}$ with $\widehat{err}(h) = 0$ and $\widehat{err}_{unl}(h) \leq t + \epsilon$ have $err(h) \leq \epsilon$, and furthermore all $h \in \mathcal{C}$ have $|err_{unl}(h) - \widehat{err}_{unl}(h)| \leq \epsilon$.

[5] I.e., exactly the case of the generative models we maligned at the start of this paper.

This is the analog of Theorem 2 for the infinite case. In particular, this implies that if $err_{unl}(c^*) \leq t$ then with probability $\geq 1-\delta$ the $h \in \mathcal{C}$ that optimizes $\widehat{err}(h)$ and $\widehat{err}_{unl}(h)$ has $err(h) \leq \epsilon$.

Proof Sketch: By standard VC-bounds [9, 23], the number of unlabeled examples is sufficient to ensure that with probability $1-\delta/2$ we can estimate, within ϵ, $\mathbf{Pr}_{x \in D}[\chi_h(x) = 1]$ for all $\chi_h \in \chi(\mathcal{C})$. Since $\chi_h(x) = \chi(h,x)$, this implies we have can estimate, within ϵ, the unlabeled error rate $err_{unl}(h)$ for all $h \in \mathcal{C}$, and so the set of hypotheses with $\widehat{err}_{unl}(h) \leq t+\epsilon$ is contained in $\mathcal{C}_{D,\chi}(t+2\epsilon)$.

The bound on the number of labeled examples follows from [9] and [1] (where it is shown that the expected number of partitions can be used instead of the maximum in the standard VC proof). This bound ensures that with probability $1-\delta/2$, none of the functions in $\mathcal{C}_{D,\chi}(t+2\epsilon)$ whose whose true (labeled) error is at least ϵ have an empirical (labeled) error of 0. ■

We can also give a bound where we specify the number of labeled examples as a function of the *unlabeled sample*; this is useful because we can imagine our learning algorithm performing some calculations over the unlabeled data and then deciding how many labeled examples to purchase. For simplicity, let us just consider the "doubly realizable case" in which we imagine the target function satisfies $err(c^*) = 0$ and $err_{unl}(c^*) = 0$.

Theorem 6 *An unlabeled sample S of size*

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

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

$$m_l > \frac{4}{\epsilon}\left[\log(2s) + \log\frac{2}{\delta}\right] \quad \text{and} \quad s = \mathcal{C}_{S,\chi}(0)\left[2m_l, \overline{S}\right]$$

then with probability $\geq 1-\delta$, all $h \in \mathcal{C}$ with $\widehat{err}(h) = 0$ and $\widehat{err}_{unl}(h) = 0$ have $err(h) \leq \epsilon$.

Proof: Standard VC-bounds (in the same form as for Theorem 5) imply that the number of *labeled* examples m_l is sufficient to guarantee the conclusion of the theorem with "$err(h)$" replaced by "$err_{\overline{S}}(h)$" (the error with respect to $\overline{S}$) and "ϵ" replaced with "$\epsilon/2$". The number of *unlabeled* examples is enough to ensure that, with probability $\geq 1-\delta/2$, for all $h \in \mathcal{C}$, $|err(h) - err_{\overline{S}}(h)| \leq \epsilon/2$. Combining these two statements yields the theorem. ■

For example, for Example 1, in the worst case (over distributions D) this will recover the standard margin sample-complexity bounds. In particular, $\mathcal{C}_{S,\chi}(0)$ contains only those separators that split S with margin $\geq \gamma$, and therefore, s is no greater than the maximum number of ways of splitting $2m_l$ points with margin γ. However, if the distribution is especially nice, then the bounds can be much better because there may be many fewer ways of splitting S with margin γ. For instance, in the case of two well-separated "blobs" discussed above, if S is large enough, we would have just $s = 4$.

6 ϵ-Cover-based Bounds

The bounds in the previous section are for uniform convergence: they provide guarantees for *any* algorithm that optimizes well on the observed data. In this section, we consider stronger bounds based on ϵ-covers that can be obtained for algorithms that behave in a specific way:

they first use the unlabeled examples to choose a "representative" set of compatible hypotheses, and then use the labeled sample to choose among these. Recall that a set $C_\epsilon \subseteq 2^X$ is an ϵ-cover for $\mathcal{C}$ with respect to D if for every $c \in \mathcal{C}$ there is a $c' \in C_\epsilon$ which is ϵ-close to c.

To illustrate how this can produce stronger bounds, imagine the case of Section 4.2. We saw there we could learn from just a single labeled example, using essentially an ϵ-cover based algorithm. However, it is not hard to show that even for a labeled set L as large as $o(\log n)$, and an unlabeled set U of size $poly(n)$, for natural D there will with high probability exist high-error functions consistent with L and compatible with U.[6] So, we do not yet have uniform convergence.

Theorem 7 *If t is an upper bound for $err_{unl}(c^*)$ and p is the size of a minimum $\epsilon-$*cover *for $\mathcal{C}_{D,\chi}(t+4\epsilon)$, then using*

$$m_u = O\left(\frac{VCdim\,(\chi_{\mathcal{C}})}{\epsilon^2} log\frac{1}{\epsilon} + \frac{1}{\epsilon^2} log\frac{2}{\delta}\right)$$

unlabeled examples and

$$m_l = O\left(\frac{1}{\epsilon}\ln\frac{p}{\delta}\right)$$

labeled examples, we can with probability $\geq 1-\delta$ identify a hypothesis which is 10ϵ close to c^*.

Proof Sketch: First, given the unlabeled sample U, define $H_\epsilon \subseteq \mathcal{C}$ as follows: for every labeling of U that is consistent with some h in $\mathcal{C}$, choose a hypothesis in $\mathcal{C}$ for which $\widehat{err}_{unl}(h)$ is smallest among all the hypotheses corresponding to that labeling. Next, we obtain C_ϵ by eliminating from H_ϵ those hypotheses f with the property that $\widehat{err}_{unl}(f) > t + 3\epsilon$. We then apply a greedy procedure on C_ϵ, and we obtain $G_\epsilon = \{g_1, \cdots, g_s\}$, as follows:

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

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

Our bound on m_u is sufficient to ensure that, with probability $\geq 1-\delta/2$, H_ϵ is an ϵ-cover of $\mathcal{C}$, which implies that, with probability $\geq 1-\delta/2$, C_ϵ is an ϵ-cover for $\mathcal{C}_{D,\chi}(t)$. It is then possible to show G_ϵ is, with probability $\geq 1-\delta/2$, a 5ϵ-cover for $\mathcal{C}_{D,\chi}(t)$ of size at most p. The idea here is that by greedily creating a 3ϵ-cover of C_ϵ with respect to distribution $\overline{U}$, we are creating a 4ϵ-cover of C_ϵ with respect to D, which is a 5ϵ-cover of $\mathcal{C}_{D,\chi}(t)$ with respect to D. Furthermore, we are doing this using no more functions than would a greedy 2ϵ-cover procedure for $\mathcal{C}_{D,\chi}(t+4\epsilon)$ with respect to D, which is no more than the optimal ϵ-cover of $\mathcal{C}_{D,\chi}(t+4\epsilon)$.

Now to learn c^* we use labeled data and we do empirical risk minimization on G_ϵ. By standard bounds [2], the number of labeled examples is enough to ensure that with probability

[6] Proof: Let D be uniform on $\{0,1\}^n$ (we have n boolean variables $x_1, \ldots, x_n$) and for simplicity let $c^*(x) = x_1$. Now, let V be the set of all variables that (a) appear in *every* positive example of L and (b) appear in *no* negative example of L. Since L has size $o(\log n)$, with high probability V has size $n^{1-o(1)}$ (on average, each labeled example cuts the size of V by a factor of 2). Now, consider the hypothesis corresponding to the conjunction of all variables in V. This correctly classifies the examples in L, and whp it classifies *every* other example in U negative because each example in U has only a $1/2^{|V|}$ chance of satisfying every variable in V, and the size of U is much less than $2^{|V|}$. So, this means it is compatible with U and consistent with L, even though its true error is high.

$\geq 1 - \delta/2$ the empirical optimum hypothesis in G_ϵ has true error at most 10ϵ. This implies that overall, with probability $\geq 1 - \delta$, we find a hypothesis of error at most 10ϵ. ■

7 Conclusions

We have provided a PAC-style model that incorporates both labeled and unlabeled data, and have given a number of sample-complexity bounds. The intent of this model is to capture many of the ways unlabeled data is typically used, and to provide a framework for thinking about when and why unlabeled data can help.

Our best (ϵ-cover based) bounds apply to strategies that use the unlabeled data first to select a small set of "reasonable" rules and then use labeled data to select among them, as do our algorithms of Section 4. It is interesting to consider how this relates to algorithms (like the original co-training algorithm) that use labeled data first, and then use unlabeled data to bootstrap from them.

Acknowledgements. We thank Santosh Vempala for a number of useful discussions.

References

[1] M. Anthony and J. Shawe-Taylor. A sufficient condition for polynomial distribution-dependent learnability. *Discrete Applied Mathematics*, 77:1–12, 1997.

[2] G.M. Benedek and A. Itai. Learnability with respect to a fixed distribution. *Theoretical Computer Science*, 86:377–389, 1991.

[3] A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22:35–52, 1998.

[4] A. Blum and R. Kannan. Learning an intersection of k halfspaces over a uniform distribution. *Journal of Computer and Systems Sciences*, 54(2):371–380, 1997.

[5] A. Blum and T. M. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100, Madison, Wisconsin, USA, 1998. ACM.

[6] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[7] V. Castelli and T.M. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 16:105–111, 1995.

[8] V. Castelli and T.M. Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory*, 42(6):2102–2117, 1996.

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

[10] J. Dunagan and S. Vempala. Optimal outlier removal in high-dimensional spaces. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, 2001.

[11] A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Inf. and Comput*, 82:246–261, 1989.

[12] A. Flaxman. Personal communication, 2003.

[13] R. Ghani, R. Jones, and C. Rosenberg, editors. *The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*. Workshop, ICML'03, 2003.

[14] R. Hwa, M. Osborne, A. Sarkar, and M. Steedman. Corrected co-training for statistical parsers. In *ICML-03 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, Washington D.C., 2003.

[15] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999)*, pages 200–209, Bled, Slovenia, 1999. Morgan Kaufmann.

[16] A. Kalai and R. Servedio. Boosting in the presence of noise. In *Proceedings of the 35th Annual Symposium on the Theory of Computing*, 2003.

[17] A. R. Klivans, R. O'Donnell, and R. Servedio. Learning intersections and thresholds of halfspaces. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 177–186, Vancouver, BC, Canada, 2002.

[18] A. Levin, Paul Viola, and Yoav Freund. Unsupervised improvement of visual detectors using co-training. In *Proceedings of the Nineth IEEE International Conference on Computer Vision (ICCV 2003)*, pages 626–633, Nice, France, 2003. IEEE.

[19] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. In *Proceedings of the Thirtieth Annual Symposium on Foundations of Computer Science*, pages 574–579, Research Triangle Park, North Carolina, October 1989.

[20] K. Nigam, A. McCallum, S. Thrun, and T. M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[21] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

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

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

[24] S. Vempala. A random sampling based algorithm for learning the intersection of half-spaces. In *Proceedings of the 38th Symposium on Foundations of Computer Science*, pages 508–513, Miami Beach, Florida, 1997.

[25] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

[26] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *The Twentieth International Conference on Machine Learning (ICML-2003)*, pages 912–912, Washington, DC, USA, 2003. AAAI Press.