21 An Augmented PAC Model for Semi-Supervised Learning

Maria-Florina Balcan
Avrim Blum

The standard PAC-learning model has proven to be a useful theoretical framework for thinking about the problem of supervised *learning. However, it does not tend to capture the assumptions underlying many* semi-*supervised learning methods. In this chapter we describe an augmented version of the PAC model designed with semi-supervised learning in mind, that can be used to help think about the problem of learning from labeled and unlabeled data and many of the different approaches taken. The model provides a unified framework for analyzing when and why unlabeled data can help, in which one can discuss both sample-complexity and algorithmic issues.*

Our model can be viewed as an extension of the standard PAC model, where in addition to a concept class $\mathcal{C}$, one also proposes a compatibility function: a type of compatibility that one believes the target concept should have with the underlying distribution of data. For example, it could be that one believes the target should cut through a low-density region of space, or that it should be self-consistent in some way as in co-training. This belief is then explicitly represented in the model. Unlabeled data is then potentially helpful in this setting because it allows one to estimate compatibility over the space of hypotheses, and to reduce the size of the search space from the whole set of hypotheses $\mathcal{C}$ down to those that, according to one's assumptions, are a-priori reasonable with respect to the distribution.

After proposing the model, 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 provide examples of sample-complexity bounds both for uniform convergence and ϵ-cover based algorithms, as well as several algorithmic results.

A PAC Model for Semi-Supervised Learning

21.1 Introduction

As we have already seen in the previous chapters, there has been growing interest 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.

One difficulty from a theoretical point of view 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 purposefully a complete disconnect between the data distribution D and the target function f being learned [Valiant, 1984, Blumer et al., 1989, Kearns and Vazirani, 1994]. The only prior belief is that f belongs to some class $\mathcal{C}$: 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 the problem of learning a concept class such as DNF formulas [Linial et al., 1989, Verbeurgt, 1990] or an intersection of halfspaces [Baum, 1990, Blum and Kannan, 1997, Vempala, 1997, Klivans et al., 2002] 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 theoretically about the use of unlabeled data, e.g., [Castelli and Cover, 1995, 1996]. However, these results typically make strong assumptions that essentially imply that there is only one natural distinction to be made for a given (unlabeled) data distribution. 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 in principle recover the Gaussians and would need labeled data only to tell us which Gaussian is the positive one and which is the negative one.[1] This is too strong an assumption for most real-world settings. 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.[2] In addition, we would like a general framework that can be used to model many different uses of unlabeled data.

In this chapter, we present a PAC-style framework that bridges between these positions and we believe can be used to help think about many of the ways

1. Castelli and Cover [1995, 1996] do not assume Gaussians in particular, but they do assume the distributions are distinguishable, which from our perspective has the same issue.
2. In fact, there has been recent work in the generative model setting on the practical side that goes in this direction (see [Nigam et al., 2000, Nigam, 2001]). We discuss connections to generative models further in Section 21.5.2.

unlabeled data is typically used, including approaches discussed in other chapters. This framework extends the PAC model in a way that allows one to express not only the form of target function one is considering, but also relationships that one hopes the target function and underlying distribution will possess. 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 also give examples of algorithmic results in this model.

Specifically, the idea of the proposed model is to augment the PAC notion of a *concept class*, which is a set of functions (like linear separators or decision trees), with a notion of *compatibility* between a function and the data distribution that we hope the target function will satisfy. Then, rather than talking of "learning a concept class $\mathcal{C}$," we will talk of "learning a concept class $\mathcal{C}$ under compatibility notion χ." For example, suppose we believe there should exist a good linear separator, and that furthermore, if the data happens to cluster, then this separator probably does not slice through the middle of any such clusters. Then we would want a compatibility notion that penalizes functions that do, in fact, slice through clusters. In this framework, the extent to which unlabeled data helps depends on two quantities: first, the extent to which the true target function satisfies the given assumption, and second, the extent to which the distribution allows this assumption to rule out alternative hypotheses. For instance, if the data does not cluster at all, then all functions equally satisfy this compatibility notion and the assumption ends up not helping. From a Bayesian perspective, one can think of this as a PAC model for a setting in which one's prior is not just over functions, but also over how the function and underlying distribution relate to each other.

Main Idea

To make our model formal, we will need to ensure that the degree of compatibility be something that can be estimated from a finite sample. To do this, we will require that the compatibility notion χ actually be a function from $\mathcal{C} \times \mathcal{X}$ to $[0, 1]$, where the compatibility of a function f with the data distribution D is $\mathbf{E}_{x \sim D}[\chi(f, x)]$. The degree of *in*compatibility is then something we can think of as a kind of "unlabeled error rate" that measures how a-priori unreasonable we believe some proposed hypothesis to be. For instance, in the example above of a "margin-style" compatibility, we could define $\chi(f, x)$ to be an increasing function of the distance of x to the separator f. In this case, the unlabeled error rate, $1 - \chi(f, D)$, is a measure of the probability mass close to the proposed separator. In co-training, where each example x has two "views" ($x = \langle x_1, x_2 \rangle$), the underlying belief is that the true target c^* can be decomposed into functions $\langle c_1^*, c_2^* \rangle$ over each view such that for most examples, $c_1^*(x_1) = c_2^*(x_2)$. In this case, we can define $\chi(\langle f_1, f_2 \rangle, \langle x_1, x_2 \rangle) = 1$ if $f_1(x_1) = f_2(x_2)$, and 0 if $f_1(x_1) \neq f_2(x_2)$. Then the compatibility of a hypothesis $\langle f_1, f_2 \rangle$ with an underlying distribution D is $\mathbf{Pr}_{\langle x_1, x_2 \rangle \sim D}[f_1(x_1) = f_2(x_2)]$.

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 (i.e., how correct we were in our assumption) 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 the target function is highly compatible with D, then if we have enough unlabeled data to estimate compatibility over all $f \in \mathcal{C}$, we can in principle reduce the size of the search space from $\mathcal{C}$ down to just those $f \in \mathcal{C}$ whose estimated compatibility is high. For instance, if D is "helpful", then the set of such functions will be much smaller than the entire set $\mathcal{C}$.

Ways in which unlabeled data can help

- 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 [Devroye et al., 1996], Rademacher complexities [Koltchinskii, 2001, Bartlett and Mendelson, 2002, Boucheron et al., 2004] or the size of the smallest ϵ-cover [Benedek and Itai, 1991], rather than VC-dimension [Blumer et al., 1989, Kearns and Vazirani, 1994]. 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" $f \in \mathcal{C}$ 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. Roughly, the "VCdim/ϵ^2" form of standard PAC sample complexity bounds now becomes a bound on the number of *unlabeled* examples we need. However, technically, the set whose VC-dimension we now care about is not $\mathcal{C}$ but rather a set defined by both $\mathcal{C}$ and χ: that is, the overall complexity depends both on the complexity of $\mathcal{C}$ and the complexity of the notion of compatibility (see Section 21.3.1.2). One consequence of our model is that if the target function and data distribution are both well behaved with respect to the compatibility notion, then the sample-size bounds we get for labeled data can substantially beat what one could hope to achieve through pure labeled-data bounds, and we illustrate this with a number of examples through the chapter.

21.2 A Formal Framework

In this section we formally introduce what we mean by a *notion of compatibility*, and illustrate it through a number of examples including margins and co-training.

We assume that examples (both labeled and unlabeled) come according to a fixed unknown distribution D over an instance space $\mathcal{X}$, 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 $\mathcal{X}$, 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 f, the (true) error rate of f is defined as $err(f) = err_D(f) = \mathbf{Pr}_{x \sim D}[f(x) \neq c^*(x)]$. For any two hypotheses

$f_1, f_2 \in \mathcal{C}$, the distance with respect to D between f_1 and f_2 is defined as $d(f_1, f_2) = d_D(f_1, f_2) = \mathbf{Pr}_{x\sim D}[f_1(x) \neq f_2(x)]$. We will use $\widehat{err}(f)$ to denote the empirical error rate of f on a given labeled sample and $\hat{d}(f_1, f_2)$ to denote the empirical distance between f_1 and f_2 on a given unlabeled sample.

We define a *notion of compatibility* to be a mapping from a hypothesis f and a distribution D to $[0, 1]$ indicating how "compatible" f is with D. In order for this to be estimable from a finite sample, we require that compatibility be an expectation over individual examples. (Though one could imagine more general notions with this property as well.) Specifically, we define:

Legal notion of compatibility

Definition 21.1 *A* legal notion of compatibility *is a function* $\chi : \mathcal{C} \times \mathcal{X} \to [0, 1]$ *where we (overloading notation) define* $\chi(f, D) = \mathbf{E}_{x\sim D}[\chi(f, x)]$. *Given a sample* S, *we define* $\chi(f, S)$ *to be the empirical average over the sample.*

Remark 21.2 *One could also allow compatibility functions over* k*-tuples of examples, in which case our (unlabeled) sample-complexity bounds would simply increase by a factor of* k. *For settings in which* D *is actually known in advance (e.g., transductive learning, see Section 21.5.1) we can drop this requirement entirely and allow any notion of compatibility* $\chi(f, D)$ *to be legal.*

Definition 21.3 *Given compatibility notion* χ, *the* incompatibility *of* f *with* D *is* $1 - \chi(f, D)$. *We will also call this its* unlabeled error rate, $err_{unl}(f)$, *when* χ *and* D *are clear from context. For a given sample* S, *we use* $\widehat{err}_{unl}(f)$ *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 21.4 *Given threshold* τ, *we define* $\mathcal{C}_{D,\chi}(\tau) = \{f \in \mathcal{C} : err_{unl}(f) \leq \tau\}$. *So, e.g.,* $\mathcal{C}_{D,\chi}(1) = \mathcal{C}$. *Similarly, for a sample* S, *we define* $\mathcal{C}_{S,\chi}(\tau) = \{f \in \mathcal{C} : \widehat{err}_{unl}(f) \leq \tau\}$

We now give several examples to illustrate this framework:

Margins

Example 1. Suppose examples are points in $\mathbb{R}^d$ 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 (see [Joachims, 1999] and also Chapter 6 in this book). In this case, if we are given γ up front, we could define $\chi(f, x) = 1$ if x is farther than distance γ from the hyperplane defined by f, and $\chi(f, x) = 0$ otherwise. So, the incompatibility of f with D is probability mass within distance γ of $f \cdot x = 0$. Or we could define $\chi(f, x)$ 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 a hypothesis 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 would not be a good definition since one cannot distinguish "zero" from "exponentially close to zero" from a small sample of unlabeled data).

Co-training

Example 2. In co-training [Blum and Mitchell, 1998], we assume examples come as pairs $\langle x_1, x_2 \rangle$, and our goal is to learn a pair of functions $\langle f_1, f_2 \rangle$. 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. For example, *iterative co-training* uses a small amount of labeled data to get some initial information (e.g., 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 for training its hypothesis over the other half. This approach and several variants have been used for a variety of learning problems, including named entity classification [Collins and Singer, 1999], text classification [Nigam and Ghani, 2000, Ghani, 2001], natural language processing [Pierce and Cardie, 2001], large scale document classification [Park and Zhang, 2003], and visual detectors [Levin et al., 2003].[3] As mentioned in Section 21.1, the assumptions underlying co-training fit naturally into our framework. In particular, we can define the incompatibility of some hypothesis $\langle f_1, f_2 \rangle$ with distribution D as $\mathbf{Pr}_{\langle x_1, x_2 \rangle \sim D}[f_1(x_1) \neq f_2(x_2)]$.

Graph-based methods

Example 3. In transductive graph-based methods, we are given a set of unlabeled examples connected in a graph $\mathtt{g}$, where the interpretation of an edge is that we believe the two endpoints of the edge should have the *same* label. Given a few labeled vertices, various graph-based methods then attempt to use them to infer labels for the remaining points. If we are willing to view D as a distribution over *edges* (a uniform distribution if $\mathtt{g}$ is unweighted), then as in co-training we can define the incompatibility of some hypothesis f as the probability mass of edges that are cut by f, which then motivates various cut-based algorithms. For instance, if we require f to be boolean, then the mincut method of Blum and Chawla [2001] finds the most-compatible hypothesis consistent with the labeled data; if we allow f to be fractional and define $1 - \chi(f, \langle x_1, x_2 \rangle) = (f(x_1) - f(x_2))^2$, then the algorithm of Zhu et al. [2003a] finds the most-compatible consistent hypothesis. If we do not wish to view D as a distribution over edges, we could have D be a distribution over *vertices* and broaden Definition 21.1 to allow for χ to be a function over *pairs* of examples. In fact, as mentioned in Remark 21.2, since we have perfect knowledge of D in this setting we can allow any compatibility function $\chi(f, D)$ to be legal. We discuss more connections with graph-based methods in Section 21.5.1.

Example 4. As a special case of co-training, suppose examples are pairs of points in $\mathbb{R}^d$, $\mathcal{C}$ is the class of linear separators, and we believe the two points in each

3. For more discussion regarding co-training see also Chapter 2 in this book.

pair should both be on the *same* side of the target function. (So, this is a version of co-training where we require $f_1 = f_2$.) The motivation is that we want to use pairwise information as in Example 3, but we also want to use the features of each data point. For instance, in the word-sense disambiguation problem studied by Yarowsky [1995], the goal is to determine which of several dictionary definitions is intended for some target word in a piece of text (e.g., is "plant" being used to indicate a tree or a factory?). The local context around each word can be viewed as placing it into $\mathbb{R}^d$, but the edges correspond to a completely different type of information: the belief that if a word appears twice in the same document, it is probably being used in the *same* sense both times. In this setting, we could use the same compatibility function as in Example 3, but rather than having the concept class $\mathcal{C}$ be all possible functions, we reduce $\mathcal{C}$ to just linear separators.

Linear separator graph cuts

Example 5. In a related setting to co-training, considered by Leskes [2005], examples are single points in $\mathcal{X}$ but we have a pair of hypothesis spaces $\langle \mathcal{C}_1, \mathcal{C}_2 \rangle$ (or more generally a k-tuple $\langle \mathcal{C}_1, \ldots, \mathcal{C}_k \rangle$), and the goal is to find a pair of hypotheses $\langle f_1, f_2 \rangle \in \mathcal{C}_1 \times \mathcal{C}_2$ with low error over labeled data and that agree over the distribution. For instance, if data is sufficiently "well-separated", one might expect there to exist both a good linear separator and a good decision tree, and one would like to use this assumption to reduce the need for labeled data. In this case one could define compatibility of $\langle f_1, f_2 \rangle$ with D as $\mathbf{Pr}_{x \sim D}[f_1(x) = f_2(x)]$, or the similar notion given in [Leskes, 2005].

Agreement

21.3 Sample Complexity Results

We now present several sample-complexity bounds that fall out of this framework, showing how unlabeled data, together with a suitable compatibility notion, can reduce the need for labeled examples.

The basic structure of all of these results is as follows. First, given enough unlabeled data (where "enough" will be a function of some measure of the complexity of $\mathcal{C}$ and possibly of χ as well), we can uniformly estimate the true compatibilities of all functions in $\mathcal{C}$ by their empirical compatibilities over the sample. Then, by using this quantity to give a preference ordering over the functions in $\mathcal{C}$, we can reduce "$\mathcal{C}$" down to "the set of functions in $\mathcal{C}$ whose compatibility is not much larger than the true target function" in bounds for the number of *labeled* examples needed for learning. The specific bounds differ in terms of the exact complexity measures used (and a few other issues such as stratification and realizability) and we provide examples illustrating when certain complexity measures can be significantly more powerful than others. In particular, ϵ-cover bounds (Section 21.3.2) can provide especially good bounds for co-training and graph-based settings.

21.3.1 Uniform Convergence Bounds

We begin with uniform convergence bounds (later in Section 21.3.2 we give tighter ϵ-cover bounds that apply to algorithms of a particular form). For clarity, we begin with the case of finite hypothesis spaces where we measure the "size" of a set of functions by just the number of functions in the set. We then discuss several issues that arise when considering infinite hypothesis spaces, such as what is an appropriate measure for the "size" of the set of compatible functions, and the need to account for the complexity of the compatibility notion itself. Note that 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 (see [Kearns and Vazirani, 1994]). 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.

21.3.1.1 Finite hypothesis spaces

We first give a bound for the "doubly realizable" case.

Theorem 21.5 *If we see m_u unlabeled examples and m_l labeled examples, where*

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

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

Proof The probability that a given hypothesis f with $err_{unl}(f) > \epsilon$ has $\widehat{err}_{unl}(f) = 0$ is at most $(1-\epsilon)^{m_u} < \delta/(2|\mathcal{C}|)$ for the given value of m_u. Therefore, by the union bound, the number of unlabeled examples is sufficient to ensure that with probability $1-\delta/2$, only hypotheses in $\mathcal{C}_{D,\chi}(\epsilon)$ have $\widehat{err}_{unl}(f) = 0$. The number of labeled examples then similarly ensures that with probability $1-\delta/2$, none of those whose true error is at least ϵ have an empirical error of 0, yielding the theorem. ■

So, if the target function indeed is perfectly correct and compatible, then Theorem 21.5 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 efficiently PAC$_{unl}$-learns the pair $(\mathcal{C}, \chi)$ if it is able to achieve a PAC guarantee using time and sample sizes polynomial in the bounds of Theorem 21.5.

We can think of Theorem 21.5 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

Interpretation

function among them. We can get a similar bound in the situation when the target function is not fully compatible:

Theorem 21.6 *Given $t \in [0,1]$, if we see m_u unlabeled examples and m_l labeled examples, where*

$$m_u \geq \frac{2}{\epsilon^2}\left[\ln|\mathcal{C}| + \ln\frac{4}{\delta}\right] \quad \textit{and} \quad m_l \geq \frac{1}{\epsilon}\left[\ln|\mathcal{C}_{D,\chi}(t+2\epsilon)| + \ln\frac{2}{\delta}\right],$$

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

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

Proof Same as Theorem 21.5 except apply Hoeffding bounds (see Devroye et al. [1996]) to the unlabeled error rates. ∎

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

Theorem 21.7 *For any set S of unlabeled data, given m_l labeled examples, with probability at least $1-\delta$,* all *$f \in \mathcal{C}$ satisfying $\widehat{err}(f) = 0$ and $\mathsf{desc}_S(f) \leq \epsilon m_l - \ln(1/\delta)$ have $err(f) \leq \epsilon$. Furthermore, if $|S| \geq \frac{2}{\epsilon^2}[\ln|\mathcal{C}| + \ln\frac{2}{\delta}]$, then with probability at least $1-\delta$, all $f \in \mathcal{C}$ satisfy $\mathsf{desc}_S(f) \leq \epsilon$-$\mathsf{desc}_D(f)$.*

Interpretation

The point of this theorem is that an algorithm can use observable quantities to determine if it can be confident. 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.

Note that if we begin with a non-distribution-dependent ordering of hypotheses, inducing some description length $\mathsf{desc}(f)$, and our compatibility assumptions turn out to be wrong, then it could well be that $\mathsf{desc}_D(c^*) > \mathsf{desc}(c^*)$. In this case our use of unlabeled data would end up hurting rather than helping.

21.3.1.2 Infinite hypothesis spaces

To reduce notation, we will assume in the rest of this chapter that $\chi(f,x) \in \{0,1\}$ so that $\chi(f,D) = \mathbf{Pr}_{x\sim D}[\chi(f,x) = 1]$. However, all our sample complexity results can be easily extended to the general case.

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 $\chi(\mathcal{C}) = \{\chi_f : f \in \mathcal{C}\}$ where we define $\chi_f(x) = \chi(f, x)$. For instance, suppose examples are just points on the line, and $\mathcal{C} = \{f_a(x) : f_a(x) = 1$ iff $x \le a\}$. In this case, $\mathrm{VCdim}(\mathcal{C}) = 1$. However, we could imagine a compatibility function such that $\chi(f_a, x)$ depends on some complicated relationship between the real numbers a and x. In this case, $\mathrm{VCdim}(\chi(\mathcal{C}))$ is much larger, and indeed we would need many more unlabeled examples to estimate compatibility over all of $\mathcal{C}$.

A second issue is that we need an appropriate measure for the "size" of the set of surviving functions. VC-dimension tends not to be a good choice: for instance, if we consider the case of Example 1 (margins), then even if data is concentrated in two well-separated "blobs", the set of compatible separators still has as large a VC-dimension as the entire class even though they are all very similar with respect to D. Instead, it is better to consider distribution dependent complexity measures such as annealed VC-entropy or Rademacher averages. For this we introduce some notation. 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) $S \subseteq \mathcal{X}$, we will denote by $\overline{S}$ the uniform distribution over S, and by $\mathcal{C}[m, \overline{S}]$ the expected number of splits of m points (drawn i.i.d.) from $\overline{S}$ with concepts in $\mathcal{C}$. Then we can get bounds as follows:

Theorem 21.8 *An unlabeled sample of size*

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

and a labeled sample of size

$$m_l > \frac{2}{\epsilon}\left[\log(2s) + \log\frac{2}{\delta}\right], \;\; where \;\; s = \mathcal{C}_{D,\chi}(t + 2\epsilon)[2m_l, D]$$

(i.e., s is the expected number of splits of $2m_l$ points drawn from D using concepts in $\mathcal{C}$ of unlabeled error rate $\le t + 2\epsilon$) is sufficient so that with probability $1 - \delta$, all $f \in \mathcal{C}$ with $\widehat{err}(f) = 0$ and $\widehat{err}_{unl}(f) \le t + \epsilon$ have $err(f) \le \epsilon$, and furthermore all $f \in \mathcal{C}$ have $|err_{unl}(f) - \widehat{err}_{unl}(f)| \le \epsilon$.

Interpretation

This is the analog of Theorem 21.6 for the infinite case. In particular, this implies that if $err(c^*) = 0$ and $err_{unl}(c^*) \le t$, then with high probability the $f \in \mathcal{C}$ that optimizes $\widehat{err}(f)$ and $\widehat{err}_{unl}(f)$ has $err(f) \le \epsilon$.

Proof Sketch: By standard VC-bounds [Devroye et al., 1996, Vapnik, 1998], the number of unlabeled examples is sufficient to ensure that with probability $1-\delta/2$ we can estimate, within ϵ, $\mathbf{Pr}_{x \in D}[\chi_f(x) = 1]$ for all $\chi_f \in \chi(\mathcal{C})$. Since $\chi_f(x) = \chi(f, x)$, this implies we have can estimate, within ϵ, the unlabeled error rate $err_{unl}(f)$ for all $f \in \mathcal{C}$, and so the set of hypotheses with $\widehat{err}_{unl}(f) \le t + \epsilon$ is contained in $\mathcal{C}_{D,\chi}(t + 2\epsilon)$.

The bound on the number of labeled examples follows from [Devroye et al., 1996] (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 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.

Theorem 21.9 *Given $t \geq 0$, an unlabeled sample S of size*

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

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

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

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

Proof Standard VC-bounds (in the same form as for Theorem 21.8) imply that the number of *labeled* examples m_l is sufficient to guarantee the conclusion of the theorem with "$err(f)$" replaced by "$err_{\overline{S}}(f)$" (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 $f \in \mathcal{C}$, $|err(f) - err_{\overline{S}}(f)| \leq \epsilon/2$. Combining these two statements yields the theorem. ■

So, if $err(c^*)=0$ and $err_{unl}(c^*) \leq t$, then with high probability the $f \in \mathcal{C}$ that optimizes $\widehat{err}(f)$ and $\widehat{err}_{unl}(f)$ has $err(f) \leq \epsilon$. If we assume $err_{unl}(c^*)=0$ then we can use $\mathcal{C}_{S,\chi}(0)$ instead of $\mathcal{C}_{S,\chi}(t+\epsilon)$.

Interpretation

Notice that for the case of Example 1, in the worst case (over distributions D) this will essentially 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 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 $s=4$.

We finally give a stratified version of Theorem 21.9 as follows:

Theorem 21.10 *An unlabeled sample S of size*

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

is sufficient so that with probability $\geq 1-\delta$ we have that simultaneously for every $k \geq 0$ the following is true: if we label m_k examples drawn uniformly at random *from S, where*

$$m_k > \frac{4}{\epsilon}\left[\log(2s) + \log\frac{2(k+1)(k+2)}{\delta}\right] \quad \textit{and} \quad s = \mathcal{C}_{S,\chi}((k+1)\epsilon)\left[2m_k, \overline{S}\right]$$

then all $f \in \mathcal{C}$ with $\widehat{err}(f) = 0$ and $\widehat{err}_{unl}(f) \leq (k+1)\epsilon$ have $err(f) \leq \epsilon$.

This theorem is an analog of Theorem 21.7 and it essentially justifies a stratification based on the estimated unlabeled error rates. We can also imagine having data dependent bounds for both labeled and unlabeled data, and also doing a double stratification, with respect to both labeled and unlabeled error rates. In particular, we can derive a bound as follows:

Theorem 21.11 *An unlabeled sample S of size*

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

is sufficient so that with probability $\geq 1-\delta$ we have that simultaneously for every $i \geq 0$, $k \geq 0$ the following is true: if we label $m_{k,i}$ examples drawn uniformly at random from S, where

$$m_{k,i} > \frac{8}{\epsilon^2}\left[\log(2s) + \log\frac{4(k+1)(k+2)(i+1)(i+2)}{\delta}\right] \quad \textit{and} \quad s = \mathcal{C}_{S,\chi}((k+1)\epsilon)\left[2m_k, \overline{S}\right]$$

then all $f \in \mathcal{C}$ with $\widehat{err}(f) \leq (i+1)\epsilon$ and $\widehat{err}_{unl}(f) \leq (k+1)\epsilon$ have $err(f) \leq (i+2)\cdot\epsilon$.

We can similarly derive tight bounds using Rademacher averages. For different versions of our statements using recent stronger bounds ([Boucheron et al., 2000], [Boucheron et al., 2004]) see [Balcan and Blum, 2005].

21.3.2 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. Bounds based on ϵ-covers exist in the classical PAC setting, but in our framework these bounds and algorithms of this type are especially natural and convenient.

Recall that a set $C_\epsilon \subseteq 2^{\mathcal{X}}$ is an ϵ-cover for $\mathcal{C}$ with respect to D if for every $f \in \mathcal{C}$ there is a $f' \in C_\epsilon$ which is ϵ-close to f. That is, $\mathbf{Pr}_{x\sim D}(f(x) \neq f'(x)) \leq \epsilon$.

To illustrate how this can produce stronger bounds, consider the setting of Example 3 (graph-based algorithms) where the graph g consists of two cliques of $n/2$ vertices, connected together by $o(n^2)$ edges (in particular, the number of edges connecting the cliques is small compared to ϵn^2). Suppose the target function labels one of the cliques as positive and one as negative, and we define compatibility of a

Examples where ϵ-cover bounds beat uniform convergence bounds

hypothesis to be the fraction of edges in g that are cut by it (so the target function indeed has unlabeled error rate less than ϵ). Now, given any set S_L of $m_l \ll \epsilon n$ labeled examples, there is always a highly-compatible hypothesis consistent with S_L that just separates the positive points in S_L from the entire rest of the graph: the number of edges cut will be at most $nm_l \ll \epsilon n^2$. However, such a hypothesis clearly has high true error since it is so unbalanced. So, we do not have uniform convergence. On the other hand, the set of functions of unlabeled error rate less than $\epsilon/4$ has a small ϵ-cover: in particular, *any* partition of g that cuts less than $\epsilon n^2/4$ edges must be ϵ-close to (a) the all-positive function, (b) the all-negative function, (c) the target function c^*, or (d) the complement of the target function $1 - c^*$. So, ϵ-cover bounds act as if the concept class had only 4 functions, and so require only a constant number of labeled examples.[4]

For another case where ϵ-cover bounds can beat uniform-convergence bounds, imagine examples are *pairs* of points in $\{0,1\}^d$, $\mathcal{C}$ is the class of linear separators, and compatibility is determined by whether both points are on the same side of the separator (i.e., the case of Example 4). Now suppose for simplicity that the target function just splits the hypercube on the first coordinate, and the distribution is uniform over pairs having the same first coordinate (so the target is fully compatible). It is not hard to show that given polynomially many unlabeled examples S_U and $\frac{1}{4}\log d$ labeled examples S_L, with high probability there will exist high-error functions consistent with S_L and compatible with S_U.[5] So, we do not yet have uniform convergence. In contrast, the cover-size of the set of functions compatible with S_U is constant, so ϵ-cover based bounds again allow learning from just a constant number of labeled examples.

In particular, we can give an ϵ-cover based bound as follows.

Theorem 21.12 *If t is an upper bound for $err_{unl}(c^*)$ and p is the size of a minimum ϵ-cover for $\mathcal{C}_{D,\chi}(t+4\epsilon)$, then using m_u unlabeled examples and m_l labeled examples for*

$$m_u = \mathcal{O}\left(\frac{VCdim\,(\chi(\mathcal{C}))}{\epsilon^2}\log\frac{1}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{2}{\delta}\right) \quad \textit{and} \quad m_l = \mathcal{O}\left(\frac{1}{\epsilon}\ln\frac{p}{\delta}\right),$$

---

4. Effectively, ϵ-cover bounds allow one to rule out a hypothesis that, say, just separates the positive points in S_L from the rest of the graph by noting that this hypothesis is very close (with respect to D) to the all-negative hypothesis, and *that* hypothesis has a high labeled-error rate.

5. Proof: Let V be the set of all variables that (a) appear in *every* positive example of S_L and (b) appear in *no* negative example of S_L. Over the draw of S_L, each variable has a $(1/2)^{2|S_L|} = 1/\sqrt{d}$ chance of belonging to V, so with high probability V has size at least $\frac{1}{2}\sqrt{d}$. Now, consider the hypothesis corresponding to the conjunction of all variables in V. This correctly classifies the examples in S_L, and w.h.p. it classifies *every* other example in S_U negative because each example in S_U has only a $1/2^{|V|}$ chance of satisfying every variable in V, and the size of S_U is much less than $2^{|V|}$. So, this means it is compatible with S_U and consistent with S_L, even though its true error is high.

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

Proof Sketch: First, given the unlabeled sample S_U, define $H_\epsilon \subseteq \mathcal{C}$ as follows: for every labeling of S_U that is consistent with some f in $\mathcal{C}$, choose a hypothesis in $\mathcal{C}$ for which $\widehat{err}_{unl}(f)$ 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 $\overline{S_U}$, 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 (see for instance [Benedek and Itai, 1991]), the number of labeled examples is enough to ensure that with probability $\geq 1-\delta/2$ 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. ■

As an interesting case where unlabeled data helps substantially, consider a co-training setting where the target c^* is fully compatible *and* D satisfies the conditional independence given the label property. As shown by Blum and Mitchell [1998], one can boost any weak hypothesis from unlabeled data in this setting (assuming one has enough labeled data to produce a weak hypothesis). Related sample complexity results are given in [Dasgupta et al., 2001]. We can actually show that given enough unlabeled data, in fact we can learn from just a single labeled example. Specifically, it is possible to show that for any concept classes $\mathcal{C}_1$ and $\mathcal{C}_2$, we have:

Theorem 21.13 *Assume that* $err(c^*) = err_{unl}(c^*) = 0$ *and* D *satisfies independence given the label. Then using* m_u *unlabeled examples and* m_l *labeled examples we can find a hypothesis that with probability* $1-\delta$ *has error at most* ϵ*, provided that*

$$m_u = \mathcal{O}\left(\frac{1}{\epsilon} \cdot \left[(VCdim(\mathcal{C}_1) + VCdim(\mathcal{C}_2)) \cdot \ln\left(\frac{1}{\epsilon}\right) + \ln\left(\frac{1}{\delta}\right)\right]\right)$$

and

$$m_l = \mathcal{O}\left(\log_{\left(\frac{1}{\epsilon}\right)}\left(\frac{1}{\delta}\right)\right)$$

ϵ-Cover Bounds for Co-training

Proof Sketch: For convenience we will show a bound with 6ϵ instead of ϵ, 3δ instead of δ, and we will assume for simplicity the setting of Example 3, where $c^* = c_1^* = c_2^*$ and also that $D_1 = D_2 = \overline{D}$ (the general case is handled similarly, but just requires more notation). We first characterize the hypotheses with true unlabeled error rate at most ϵ. Recall that $\chi(f, D) = \mathbf{Pr}_{\langle x_1,x_2\rangle\sim D}[f(x_1) = f(x_2)]$, and for concreteness assume f predicts using x_1 if $f(x_1) \neq f(x_2)$. Consider $f \in \mathcal{C}$ with $err_{unl}(f) \leq \epsilon$ and let's define $p_- = \mathbf{Pr}_{x\in\overline{D}}[c^*(x) = 0]$, $p_+ = \mathbf{Pr}_{x\in\overline{D}}[c^*(x) = 1]$ and for $i, j \in \{0, 1\}$ define $p_{ij} = \mathbf{Pr}_{x\in\overline{D}}[f(x) = i, c^*(x) = j]$. We clearly have $err(f) = p_{10} + p_{01}$. From $err_{unl}(f) = \mathbf{Pr}_{(x_1,x_2)\sim D}[f(x_1) \neq f(x_2)] \leq \epsilon$, using the independence given the label of D, we get $\frac{2p_{10}p_{00}}{p_-} + \frac{2p_{01}p_{11}}{p_+} \leq \epsilon$. This implies that the almost compatible hypothesis f must be of one the following four types:

1. f is "close to c^*" or more exactly $err(f) \leq 2\epsilon$.
2. f is "close to the opposite of c^*" or more exactly $err(f) \geq 1 - 2\epsilon$.
3. f "predicts almost always negative" or more exactly $p_{10} + p_{11} \leq 3\epsilon$.
4. f "predicts almost always positive" or more exactly $p_{01} + p_{00} \leq 3\epsilon$.

Now, consider f_1 to be the constant positive function, f_0 to be the constant negative function. The unlabeled sample S_U is sufficient to ensure that probability $\geq 1 - \delta$, every hypothesis with zero estimated unlabeled error has true unlabeled error at most ϵ. Therefore, by our previous analysis, there are only four kinds of hypotheses consistent with unlabeled data: those close to c^*, those close to its complement $\overline{c^*}$, those close to f_0, and those close to f_1. Furthermore, c^*, $\overline{c^*}$, f_0, and f_1 *are* compatible with the unlabeled data.

We now check if there exists a hypothesis $g \in \mathcal{C}$ with $\widehat{err}_{unl}(g) = 0$ such that $\hat{d}_{f_1,g} \geq 4\epsilon$ and $\hat{d}_{f_0,g} \geq 4\epsilon$. If such a hypothesis g exists, then we know that one of $\{g, \overline{g}\}$, where $\overline{g}$ is the opposite of g, is 2ϵ-close to c^*. If not, we must have $p_+ \leq 6\epsilon$ or $p_- \leq 6\epsilon$, in which case we know that one of $\{f_0, f_1\}$ is 6ϵ-close to c^*. So, we have a set of two functions, opposite to each other, one of which is at least 6ϵ-close to c^*. We now use labeled data to pick one of these to output, using Lemma 21.14 below. ■

Lemma 21.14 *Consider $\epsilon < \frac{1}{8}$. Let $C_\epsilon = \left\{f, \overline{f}\right\}$ be a subset of $\mathcal{C}$ containing two opposite hypotheses with the property that one of them is ϵ-close to c^*. Then, $m_l > 6\log_{\left(\frac{1}{\epsilon}\right)}\left(\frac{1}{\delta}\right)$ labeled examples are sufficient so that with probability $\geq 1 - \delta$, the concept in C_ϵ that is ϵ-close to c^* in fact has lower empirical error.*

Proof Easy calculation: if $m_l > 6\log_{\frac{1}{\epsilon}}\left(\frac{1}{\delta}\right)$, then $\sum_{k=0}^{\lfloor\frac{m_l}{2}\rfloor}\binom{m_l}{k}\epsilon^{(m_l-k)}(1-\epsilon)^k \leq \delta$. ■

In particular, by reducing ϵ to poly(δ), we can reduce the number of labeled examples needed m_l to 1. In fact, this result can be extended to the case considered in [Balcan et al., 2004] that D^+ and D^- merely satisfy constant expansion.

This example illustrates that if data is especially well behaved with respect to the compatibility notion, then our bounds on labeled data can be extremely good. In Section 21.4.2, we show for the case of linear separators and independence given the label, we can give *efficient* algorithms, achieving the bounds in Theorem 21.13 in terms of labeled examples by a polynomial time algorithm. Note, however, that both these bounds rely heavily on the assumption that the target is fully compatible. If the assumption is more of a "hope" than a belief, then one would need additional labeled examples just to validate the hypothesis produced.

21.4 Algorithmic Results

In this section we give several examples of *efficient* algorithms in our model.

21.4.1 A simple case

We give here a simple example to illustrate the bounds in Section 21.3.1.1, and for which we can give a polynomial-time algorithm that takes advantage of them. Let the instance space $\mathcal{X} = \{0,1\}^d$, and for $x \in \mathcal{X}$, let $\mathsf{vars}(x)$ be the set of variables set to 1 by x. Let $\mathcal{C}$ be the class of monotone disjunctions (e.g., $x_1 \vee x_3 \vee x_6$), and for $f \in \mathcal{C}$, let $\mathsf{vars}(f)$ be the set of variables disjoined by f. Now, suppose we say an example x is compatible with function f if either $\mathsf{vars}(x) \subseteq \mathsf{vars}(f)$ or else $\mathsf{vars}(x) \cap \mathsf{vars}(f) = \phi$. 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 d 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 components. If the target function is fully compatible, then no component will get multiple labels (if some component does get multiple labels, we halt with failure). Finally, we produce the hypothesis f such that $\mathsf{vars}(f)$ 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 21.5, if the sizes of the data sets 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 21.5, 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 d components. In this case, unlabeled data does not help at all, and one still needs $\Omega(d)$ labeled examples (or, even $\Omega(d/\epsilon)$ if the distribution is non-uniform as in the lower bounds of Ehrenfeucht et al. [1989]). On the other hand, a helpful distribution is one such that with high probability the number of components is small, such as the case of features appearing independently given the label.

21.4.2 Co-training with linear separators

We now consider the case of co-training where the hypothesis class is the class of linear separators. For simplicity we focus first on the case of Example 4: the target function is a linear separator in $\mathbb{R}^d$ 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 hyperplane). We then show how our results can be extended to the more general setting.

As in the previous example, a natural approach is to try to solve the "consistency" problem: 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 (i.e., it gets the labeled data correct and doesn't cut too many edges). Unfortunately, this consistency problem is NP-hard: given a graph g embedded in $\mathbb{R}^d$ 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* [Flaxman, 2003]. 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 $\overline{D}$ over $\mathbb{R}^d$, and with some probability p_+, two points are drawn i.i.d. from $\overline{D}_+$ ($\overline{D}$ restricted to the positive side of the target function) and with probability $1 - p_+$, the two are drawn i.i.d from $\overline{D}_-$ ($\overline{D}$ restricted to the negative side of the target function). Note that our sample complexity results in section 21.3.2 extend to weaker assumptions such as distributional expansion introduced by Balcan et al. [2004], but we need true independence for our algorithmic results. Blum and Mitchell [1998] have also given 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[6] (which is true for linear separators [Blum et al., 1998]), and (c) we have enough labeled data to produce a weakly-useful hypothesis (defined below) on one of the halves to begin with. We give here an improvement over that result by showing how we can run the algorithm in [Blum and Mitchell, 1998] with only *a single* labeled example, thus obtaining an efficient algorithm in our model. It is worth noticing that in the process, we also simplify the results of Blum et al. [1998]

Need to assume independence for our algorithmic results

6. For a detailed description of the Statistical Query model see [Kearns, 1998] and [Kearns and Vazirani, 1994].

somewhat.

For the analysis below, we need the following definition. A *weakly-useful* predictor is a function f such that for some ϵ that is at least inverse polynomial in the input size:

$$\mathbf{Pr}[f(x) = 1 | c^*(x) = 1] > \mathbf{Pr}[f(x) = 1 | c^*(x) = 0] + \epsilon.$$

It is equivalent to the usual notion of a "weak hypothesis" (see [Kearns and Vazirani, 1994]) when the target function is balanced, but requires the hypothesis give more information when the target function is unbalanced — see [Blum and Mitchell, 1998].

Theorem 21.15 *There is a polynomial-time algorithm (in d and b, where b is the number of bits per example) 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 $S = \left\{\langle x_1^i, x_2^i \rangle\right\}$; denote by S_j the set $\left\{x_j^i\right\}$, for $j = 1, 2$. (We describe our algorithm as working with the fixed unlabeled sample S, since we just need to apply standard VC-dimension arguments to get the desired result.) The first step is to perform a transformation T on S_1 to ensure that some reasonable ($1/poly$) fraction of $T(S_1)$ has margin at least $1/poly$, which we can do via the Outlier Removal Lemma of Blum et al. [1998] and Dunagan and Vempala [2001].[7] The Outlier Removal Lemma states that one can algorithmically remove an ϵ' fraction of S_1 and ensure that for the remainder, for any vector w, $\max_{x \in S_1} (w \cdot x)^2 \leq poly(n, b, 1/\epsilon') \mathbf{E}_{x \in S_1}[(w \cdot x)^2]$, where b is the number of bits needed to describe the input points. We reduce the dimensionality (if necessary) to get rid of any of the vectors for which the above quantity is zero. We then determine a linear transformation (as described in Blum et al. [1998]) so that in that in the transformed space for all unit-length w, $\mathbf{E}_{x \in T(S_1)}[(w \cdot x)^2] = 1$). Since the maximum is bounded, this guarantees that at least a $1/poly$ fraction of the points in $T(S_1)$ have at least a $1/poly$ margin with respect to the separating hyperplane.

To avoid cumbersome notation in the rest of the discussion, we drop our use of "T" and simply use S and c^* to denote the points and separator in the transformed space. (If the distribution originally had a reasonable probability mass

7. If the reader is willing to allow running time polynomial in the margin of the data set, then this part of the argument is not needed.

at a reasonable margin from c^*, then T could be the identity anyway.)

The second step is we argue that a *random* halfspace has at least a $1/poly$ chance of being a weak predictor on S_1. ([Blum et al., 1998] uses the Perceptron algorithm to get weak learning; here, we need something simpler since we do not yet have any labeled data.) Specifically, consider a point x such that the angle between x and c^* is $\pi/2 - \gamma$, and imagine that we draw f at random subject to $f \cdot c^* \geq 0$ (half of the f's will have this property). Then,

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

Since at least a $1/poly$ fraction of the points in S_1 have at least a $1/poly$ margin this implies that:

$$\mathbf{Pr}_{f,x}[f(x) = 1 | c^*(x) = 1] > \mathbf{Pr}_{f,x}[f(x) = 1 | c^*(x) = 0] + 1/poly.$$

This means that a $1/poly$ probability mass of functions f must in fact be weakly-useful predictors.

The final step of the algorithm is as follows. Using the above observation, we pick a random f, and plug it into the bootstrapping theorem of [Blum and Mitchell, 1998] (which, given unlabeled pairs $\langle x_1^i, x_2^i \rangle \in S$, will use $f(x_1^i)$ as a noisy label of x_2^i, feeding the result into an SQ algorithm), repeating this process $poly(n)$ times. With high probability, our random f 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 the runs of the algorithm, we have no guarantees. We now observe the following. First of all, any function f with small $err(f)$ must have small $err_{unl}(f)$. Secondly, because of the assumption of independence given the label, as shown in Theorem 21.13, 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.

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 f or $\neg f$ is close to c^*. We can now just draw a single labeled example to determine which case is which. ■

We can easily extend our algorithm to the standard co-training setting (where c_1^* can be different from c_2^*) as follows: we repeat the procedure in a symmetric way, and then, in order to find a good pair of functions, just try all combinations of pairs of functions to find one of small unlabeled error rate, not close to "all positive", or "all negative". Finally we use one labeled example to produce a low error hypothesis (and here we use only one part of the example and only one of the functions in the pair).

21.5 Related Models and Discussion

21.5.1 A Transductive Analog of our Model

We can also talk about a transductive analog of our (inductive) model, that incorporates many of the existing transductive methods for learning with labeled and unlabeled data. In a transductive setting one assumes that the unlabeled sample S is given, a random small subset is labeled, and the goal is to predict well on the rest of S. In order to make use of unlabeled examples, we will again express the relationship we hope the target function has with the distribution through a compatibility notion χ. However, since in this case the compatibility between a given hypothesis and D is completely determined by S (which is known), we will not need to require that compatibility be an expectation over unlabeled examples. Given this setup, from the sample complexity point of view we only care about how much labeled data we need, and algorithmically we need to find a highly compatible hypothesis with low error on the labeled data.

Rather than presenting general theorems, we instead focus on the modeling aspect and give here several examples in the context of graph-based semi-supervised algorithms for binary classification. In these methods one usually assumes that there is weighted graph g defined over S, which is given a-priori and encodes the prior knowledge. In the following we denote by W the weighted adjacency matrix of g and by $\mathcal{C}_S$ the set of all binary functions over S.

Minimum cut

Minimum cut: Suppose for $f \in \mathcal{C}_S$ we define the incompatibility of f to be the weight of the cut in g determined by f. This is the implicit notion of compatibility considered in [Blum and Chawla, 2001], and algorithmically the goal is to find the most compatible hypothesis that gets the labeled data correct, which can be solved efficiently using network flow. From a sample-complexity point of view, the number of labeled examples we need is proportional to the VC-dimension of the class of hypotheses that are at least as compatible as the target function, which is known to be $\mathcal{O}(k/\lambda)$ (see [Kleinberg, 2000], [Kleinberg et al., 2004]), where k is the number of edges cut by c^* and λ is the size of the global minimum cut in the graph. Also note that the Randomized Mincut algorithm (considered by Blum et al. [2004]), which is an extension of the basic mincut approach can be viewed as motivated by a PAC-Bayes sample complexity analysis of the problem.

Normalized Graph Cuts with Constraints

Normalized Cut: Consider the normalized cut setting of Joachims [2003] and for $f \in \mathcal{C}_S$ define $size(f)$ to be the weight of the cut in g determined by f, and let f_{neg} and f_{pos} be the number of points in S on which h predicts negative and positive, respectively. For $f \in \mathcal{C}_S$, define the incompatibility of f to be $\frac{size(f)}{f_{neg} \cdot f_{pos}}$. Note that this is the implicit compatibility function used in Joachims [2003], and again, algorithmically the goal would be to find a highly compatible hypothesis that gets the labeled data correct. Unfortunately, the corresponding optimization problem is in this case is NP-hard. Still, several approximate solutions have been considered, leading to different semi-supervised learning algorithms. For instance, Joachims [2003] considers a spectral relaxation that leads to the "SGT algorithm"; another relaxation based on Semi-Definite programming is considered by Bie and Cristianini [2004].[8]

Gaussian Random Field and Harmonic Function

Harmonic Function: We can also model the algorithms introduced in [Zhu et al., 2003a], [Zhu et al., 2003b] as follows. If we consider f to be a probabilistic prediction function defined over S, then the incompatibility of f is given by $\sum_{i,j} w_{i,j} \left(f(i) - f(j)\right)^2 = f^T L f$, where L is the un-normalized Laplacian of g. Similarly we can model the algorithm introduced by Zhou et al. [2004] by noticing that the incompatibility of f is given by $f^T \mathcal{L} f$ where $\mathcal{L}$ is the normalized Laplacian of g. More generally, all the Graph Kernel methods can be viewed in our framework if we consider that the incompatibility of f is given by $||f||_K = f^T K f$ where K is a kernel derived from the graph (see for instance [Zhu et al., 2003c]).

21.5.2 Connections to Generative Models

How the generative models fit into our model

It is also interesting to consider how generative models fit into our model. As mentioned in Section 21.1, a typical assumption in a generative setting is that D is a mixture with the probability density function $p(x|\theta) = p_0 \cdot p_0(x|\theta_0) + p_1 \cdot p_1(x|\theta_1)$ (see for instance [Ratsaby and Venkatesh, 1995], [Castelli and Cover, 1995, 1996]). That means that the labeled examples are generated according to the following mechanism: a label $y \in \{0, 1\}$ is drawn according to the distribution of classes $\{p_0, p_1\}$ and then a corresponding random feature vector is drawn according to the class-conditional density p_y. The assumption typically used is that the mixture is identifiable. Identifiability ensures that the Bayes optimal decision border $\{x : p_0 \cdot p_0(x|\theta_0) = p_1 \cdot p_1(x|\theta_1)\}$ can be deduced if $p(x|\theta)$ is known, and therefore one can construct an estimate of the Bayes border by using $p(x|\hat{\theta})$ instead of $p(x|\theta)$. Essentially once the decision border is estimated, a small labeled sample suffices to learn (with high confidence and small error) the appropriate class labels associated with the two disjoint regions generated by the estimate of the Bayes decision border. To see how we can incorporate this setting in our model, consider for illustration the setting in Ratsaby and Venkatesh [1995]; there they assume that $p_0 = p_1$, and that the class conditional densities are d-dimensional Gaussians with unit covariance and unknown mean vectors $\theta_i \in \mathbb{R}^d$. The algorithm used is the following: the unknown parameter vector $\theta = (\theta_0, \theta_1)$ is estimated from unlabeled data using a maximum likelihood estimate; this determines a hypothesis which is a linear separator that passes through the point $(\hat{\theta}_0 + \hat{\theta}_1)/2$ and is orthogonal to the vector $\hat{\theta}_1 - \hat{\theta}_0$; finally each of the two decision regions separated by the hyperplane is labeled according to the majority of the labeled examples in the region. Given this setting, a natural notion of compatibility we can consider is the expected log-likelihood function (where the expectation is taken with respect to the unknown distribution specified

8. For a more detailed discussion on this see also Chapter 7 in this book.

by θ). Specifically, we can identify a legal hypothesis $f_{\overline{\theta}}$ with the set of parameters $\overline{\theta} = (\overline{\theta}_0, \overline{\theta}_1)$ that determine it, and then we can define $\chi(f_{\overline{\theta}}, D) = \mathbf{E}_{x \in D}[\log(p(x|\overline{\theta}))]$. Ratsaby and Venkatesh [1995] show that if the unlabeled sample is large enough, then all hypotheses specified by parameters $\overline{\theta}$ which are close enough to θ, will have the property that their empirical compatibilities will be close enough to their true compatibilities. This then implies (together with other observations about Gaussian mixtures) that the maximum likelihood estimate will be close enough to θ, up to permutations. (This actually motivates χ as a good compatibility function in our model.)

More generally, if we deal with other parametric families (but we are in the same setting), we can use the same compatibility notion; however, we will need to impose certain constraints on the distributions allowed in order to ensure that the compatibility is actually well defined (the expected log-likelihood is bounded).

As mentioned in section 21.1 this kind of generative setting is really at the extreme of our model. The assumption that the distribution that generates the data is really a mixture implies that if we knew the distribution, then there are only two possible concepts left (and this makes the unlabeled data extremely useful).

21.5.3 Connections to the Luckiness Framework

Relationship to the luckiness framework

It is worth noticing that there is a strong connection between our approach and the luckiness framework (see [Shawe-Taylor et al., 1998], [Mendelson and Philips, 2003]). 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 (being designed for supervised learning only) 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 4 described in Section 21.2, for any non-degenerate distribution, a dataset of $d/2$ pairs can with probability 1 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, and learn from $o(d)$ or even $\mathcal{O}(1)$ labeled examples depending on the distribution – see Section 21.3.2 and 21.4. 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 true compatibility. There are also a number of differences at the technical level of the definitions.

21.5.4 Conclusions

Given the easy availability of unlabeled data in many settings, there has been growing interest in methods that try to use such data together with the (more expensive) labeled data for learning. Nonetheless, there is still substantial disagreement and no clear consensus about when unlabeled data helps and by how much. In this chapter, we have provided a PAC-style model for semi-supervised learning that captures many of the ways unlabeled data is typically used, and provides a very general framework for thinking about this issue. The high level main implication of our analysis is that unlabeled data is useful if (a) we have a good notion of compatibility so that 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. One consequence of our model is that if the target function and data distribution are both well behaved with respect to the compatibility notion, then the sample-size bounds we get for labeled data can substantially beat what one could hope to achieve through pure labeled-data bounds, and we have illustrated this with a number of examples through the chapter.

References

Y. S. Abu-Mostafa. Machines that learn from hints. *Scientific American*, 272(4):64–69, 1995.

A. K. Agrawala. Learning with a probabilistic teacher. *IEEE Transactions on Information Theory*, 16:373–379, 1970.

M.-F. Balcan and A. Blum. An augmented PAC model for semi-supervised learning. *Manuscript*, 2005.

M.-F. Balcan, A. Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. In *NIPS*, 2004.

P. Bartlett and S. Mendelson. Rademacher and gaussian complexities risk bounds and structural results. *Journal of Machine Learning Research*, pages 463–482, 2002.

E. B. Baum. Polynomial time algorithms for learning neural nets. In *Proceedings of the third annual workshop on Computational learning theory*, pages 258 – 272, 1990.

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

T. De Bie and N. Cristianini. Convex transduction with the normalized cut. Manuscript, 2004.

A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 19–26, 2001.

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.

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, 1998.

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

A. Blum, J. Lafferty, R. Reddy, and M. R. Rwebangira. Semi-supervised learning using randomized mincuts. In *ICML '04*, 2004.

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.

S. Boucheron, G. Lugosi, and P. Massart. A sharp concentration inequality with applications. *Random Structures and Algorithms*, 16:277–292, 2000.

S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: A survey of recent advances. Manuscript, 2004.

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

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.

M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 189–196, 1999.

S. Dasgupta, M. L. Littman, and D. McAllester. Pac generalization bounds for co-training. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2001. MIT Press.

L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-

Verlag, 1996.

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

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.

A. Flaxman. Personal communication, 2003.

S. C. Fralick. Learning to recognize patterns wothout a teacher. *IEEE Transactions on Information Theory*, 13:57–64, 1967.

R. Ghani. Combining labeled and unlabeled data for text classification with a large number of categories. In *Proceedings of the IEEE International Conference on Data Mining*, 2001.

T. Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of the International Conference on Machine Learning (ICML 2003)*, 2003.

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.

M. Kearns. Efficient noise-tolerant learning from statistical queries. In *Journal of the ACM (JACM)*, pages 983 – 1006, 1998.

M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

J. Kleinberg. Detecting a network failure. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 231–239, 2000.

J. Kleinberg, M. Sandler, and A. Slivkins. Network failure detection and graph connectivity. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 231–239, 2004.

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, 2002.

V. Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Trans. Inform. Theory*, pages 1902–1914, 2001.

B. Leskes. The value of agreement, a new boosting algorithm. In *COLT*, pages 51 – 56, 2005.

A. Levin, P. Viola, and Y. 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.

N. Linial, Y. Mansour, and N. 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.

S. Mendelson and P. Philips. Random subclass bounds. In *Proceedings of the 16th Annual Conference on Computational Learning Theory (COLT)*, 2003.

K. Nigam. Using unlabeled data to improve text classification. Technical Report Doctoral Disertation, CMU-CS-01-126, Carnegie Mellon University, 2001.

K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proc. ACM CIKM Int. Conf. on Information and Knowledge Management*, pages 86–93, 2000.

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.

S. Park and B. Zhang. Large scale unstructured document classification using unlabeled data and syntactic information. In *PAKDD 2003*, LNCS vol. 2637, pages 88–99. Springer, 2003.

D. Pierce and C. Cardie. Limitations of Co-Training for natural language learning from large datasets. In *Proc. Conference on Empirical Methods in NLP*, pages 1–9, 2001.

J. Ratsaby and S. Venkatesh. Learning from a mixture of labeled and unlabeled examples with parametric side information. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, pages 412–417, 1995.

H. J. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11:363–371, 1965.

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.

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

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

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, 1997.

K. A. Verbeurgt. Learning dnf under the uniform distribution in quasi-polynomial time. In *COLT*, pages 314–326, 1990.

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

D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schlkopf. Learning with local and global consistency. In *NIPS*, 2004.

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

X. Zhu, Z. Ghahramani, and J. Lafferty. Combinig active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning*, pages 912–912, Washington, DC, USA, 2003b.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning: From gaussian fields to gaussian processes. Technical report, Carnegie Mellon University, 2003c.