Combining Labeled and Unlabeled Data with Co-Training[*†]

Avrim Blum
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
`avrim+@cs.cmu.edu`

Tom Mitchell
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
`mitchell+@cs.cmu.edu`

Abstract

We consider the problem of using a large unlabeled sample to boost performance of a learning algorithm when only a small set of labeled examples is available. In particular, we consider a problem setting motivated by the task of learning to classify web pages, in which the description of each example can be partitioned into two distinct views. For example, the description of a web page can be partitioned into the words occurring on that page, and the words occurring in hyperlinks that point to that page. We assume that either view of the example would be sufficient for learning if we had enough labeled data, but our goal is to use both views together to allow inexpensive *unlabeled* data to augment a much smaller set of labeled examples. Specifically, the presence of two distinct views of each example suggests strategies in which two learning algorithms are trained separately on each view, and then each algorithm's predictions on new unlabeled examples are used to enlarge the training set of the other. Our goal in this paper is to provide a PAC-style analysis for this setting, and, more broadly, a PAC-style framework for the general problem of learning from both labeled and unlabeled data. We also provide empirical results on real web-page data indicating that this use of unlabeled examples can lead to significant improvement of hypotheses in practice.

[*]This paper is to appear in the *Proceedings of the 1998 Conference on Computational Learning Theory.*

[†]This research was supported in part by the DARPA HPKB program under contract F30602-97-1-0215 and by NSF National Young Investigator grant CCR-9357793.

1 INTRODUCTION

In many machine learning settings, unlabeled examples are significantly easier to come by than labeled ones [4, 15]. One example of this is web-page classification. Suppose that we want a program to electronically visit some web site and download all the web pages of interest to us, such as all the CS faculty member pages, or all the course home pages at some university [1]. To train such a system to automatically classify web pages, one would typically rely on hand labeled web pages. These labeled examples are fairly expensive to obtain because they require human effort. In contrast, the web has hundreds of millions of unlabeled web pages that can be inexpensively gathered using a web crawler. Therefore, we would like our learning algorithm to be able to take as much advantage of the unlabeled data as possible.

This web-page learning problem has an interesting feature. Each example in this domain can naturally be described using several different "kinds" of information. One kind of information about a web page is the text appearing on the document itself. A second kind of information is the anchor text attached to hyperlinks pointing *to* this page, from other pages on the web.

The two problem characteristics mentioned above (availability of both labeled and unlabeled data, and the availability of two different "kinds" of information about examples) suggest the following learning strategy. Using an initial small set of labeled examples, find weak predictors based on each kind of information; for instance, we might find that the phrase "research interests" on a web page is a weak indicator that the page is a faculty home page, and we might find that the phrase "my advisor" on a link is an indicator that the page being pointed to is a faculty page. Then, attempt to bootstrap from these weak predictors using *un*labeled data. For instance, we could search for pages pointed to with links having the phrase "my advisor" and use them as "probably positive" examples to further train a learning algorithm based on the words on the text page, and vice-versa. We call this type of bootstrapping *co-training*, and it has a close connection to bootstrapping from incomplete data in the Expectation-Maximization setting; see, for instance, [5, 13]. The question this raises is: is there any reason to believe co-training will help? Our goal is to address this question by developing a

PAC-style theoretical framework to better understand the issues involved in this approach. We also give some preliminary empirical results on classifying university web pages (see Section 6) that are encouraging in this context.

More broadly, the general question of how unlabeled examples can be used to augment labeled data seems a slippery one from the point of view of standard PAC assumptions. We address this issue by proposing a notion of "compatibility" between a data distribution and a target function (Section 2) and discuss how this relates to other approaches to combining labeled and unlabeled data (Section 3).

2 A FORMAL FRAMEWORK

We define the co-training model as follows. We have an instance space $X = X_1 \times X_2$, where X_1 and X_2 correspond to two different "views" of an example. That is, each example x is given as a pair (x_1, x_2). We assume that each view in itself is sufficient for correct classification. Specifically, let $\mathcal{D}$ be a distribution over X, and let C_1 and C_2 be concept classes defined over X_1 and X_2, respectively. What we assume is that all labels on examples with non-zero probability under $\mathcal{D}$ are consistent with some target function $f_1 \in C_1$, and are also consistent with some target function $f_2 \in C_2$. In other words, if f denotes the combined target concept over the entire example, then for any example $x = (x_1, x_2)$ observed with label ℓ, we have $f(x) = f_1(x_1) = f_2(x_2) = \ell$. This means in particular that $\mathcal{D}$ assigns probability zero to any example (x_1, x_2) such that $f_1(x_1) \neq f_2(x_2)$.

Why might we expect unlabeled data to be useful for amplifying a small labeled sample in this context? We can think of this question through the lens of the standard PAC supervised learning setting as follows. For a given distribution $\mathcal{D}$ over X, we can talk of a target function $f = (f_1, f_2) \in C_1 \times C_2$ as being "compatible" with $\mathcal{D}$ if it satisfies the condition that $\mathcal{D}$ assigns probability zero to the set of examples (x_1, x_2) such that $f_1(x_1) \neq f_2(x_2)$. That is, the pair (f_1, f_2) is compatible with $\mathcal{D}$ if f_1, f_2, and $\mathcal{D}$ are legal together in our framework. Notice that even if C_1 and C_2 are large concept classes with high complexity in, say, the VC-dimension measure, for a given distribution $\mathcal{D}$ the set of *compatible* target concepts might be much simpler and smaller. Thus, one might hope to be able to use unlabeled examples to gain a better sense of which target concepts are compatible, yielding information that could reduce the number of labeled examples needed by a learning algorithm. In general, we might hope to have a tradeoff between the number of unlabeled examples and the number of labeled examples needed.

To illustrate this idea, suppose that $X_1 = X_2 = \{0,1\}^n$ and $C_1 = C_2 =$ "conjunctions over $\{0,1\}^n$." Say that it is known that the first coordinate is relevant to the target concept f_1 (i.e., if the first coordinate of x_1 is 0, then $f_1(x_1) = 0$ since f_1 is a conjunction). Then, any unlabeled example (x_1, x_2) such that the first coordinate of x_1 is zero can be used to produce a (labeled) negative example x_2 of f_2. Of course, if $\mathcal{D}$ is an

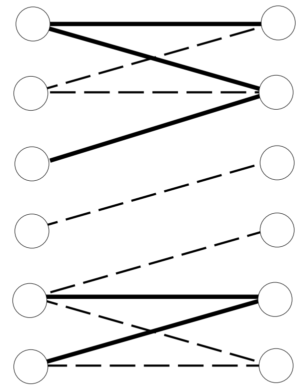

Figure 1: Graphs $G_{\mathcal{D}}$ and G_S. Edges represent examples with non-zero probability under $\mathcal{D}$. Solid edges represent examples observed in some finite sample S. Notice that given our assumptions, even without seeing any labels the learning algorithm can deduce that any two examples belonging to the same connected component in G_S must have the same classification.

"unhelpful" distribution, such as one that has nonzero probability only on pairs where $x_1 = x_2$, then this may give no useful information about f_2. However, if x_1 and x_2 are not so tightly correlated, then perhaps it does. For instance, suppose $\mathcal{D}$ is such that x_2 is conditionally independent of x_1 given the classification. In that case, given that x_1 has its first component set to 0, x_2 is now a *random* negative example of f_2, which could be quite useful. We explore a generalization of this idea in Section 5, where we show that any weak hypothesis can be boosted from unlabeled data if $\mathcal{D}$ has such a conditional independence property and if the target class is learnable with random classification noise.

In terms of other PAC-style models, we can think of our setting as somewhat in between the uniform distribution model, in which the distribution is particularly neutral, and teacher models [6, 8] in which examples are being supplied by a helpful oracle.

2.1 A BIPARTITE GRAPH REPRESENTATION

One way to look at the co-training problem is to view the distribution $\mathcal{D}$ as a weighted bipartite graph, which we write as $G_{\mathcal{D}}(X_1, X_2)$, or just $G_{\mathcal{D}}$ if X_1 and X_2 are clear from context. The left-hand side of $G_{\mathcal{D}}$ has one node for each point in X_1 and the right-hand side has one node for each point in X_2. There is an edge (x_1, x_2) if and only if the example (x_1, x_2) has non-zero probability under $\mathcal{D}$. We give this edge a weight equal to its probability. For convenience, remove any vertex of degree 0, corresponding to those views having zero probability. See Figure 1.

In this representation, the "compatible" concepts in C are exactly those corresponding to a partition of this graph with no cross-edges. One could also reasonably define the extent to which a partition is *not* compatible as the weight of the cut it induces in G. In other words, the degree of compatibility of a target function $f = (f_1, f_2)$ with a distribution $\mathcal{D}$ could be defined as a number $0 \leq p \leq 1$ where $p = 1 - \Pr_{\mathcal{D}}[(x_1, x_2) : f_1(x_1) \neq f_2(x_2)]$. In this paper, we assume full compatibility ($p = 1$).

Given a set of unlabeled examples S, we can similarly define a graph G_S as the bipartite graph having one edge (x_1, x_2) for each $(x_1, x_2) \in S$. Notice that given our assumptions, any two examples belonging to the same connected component in S must have the same classification. For instance, two web pages with the exact same content (the same representation in the X_1 view) would correspond to two edges with the same left endpoint and would therefore be required to have the same label.

3 A HIGH LEVEL VIEW AND RELATION TO OTHER APPROACHES

In its most general form, what we are proposing to add to the PAC model is a notion of compatibility between a concept and a data distribution. If we then postulate that the target concept must be compatible with the distribution given, this allows unlabeled data to reduce the class C to the smaller set C' of functions in C that are also compatible with what is known about $\mathcal{D}$. (We can think of this as intersecting C with a concept class $C_{\mathcal{D}}$ associated with $\mathcal{D}$, which is partially known through the unlabeled data observed.) For the co-training scenario, the specific notion of compatibility given in the previous section is especially natural; however, one could imagine postulating other forms of compatibility in other settings.

We now discuss relations between our setting and other methods that have been used for combining labeled and unlabeled data.

One standard approach to learning with missing values (e.g., such as when some of the labels are unknown) is the EM algorithm [3]. The EM algorithm is typically analyzed under the assumption that the data is generated according to some simple known parametric model. For instance, a common assumption is that the positive examples are generated according to an n-dimensional Gaussian $\mathcal{D}_+$ centered around the point θ_+, and negative examples are generated according to Gaussian $\mathcal{D}_-$ centered around the point θ_-, where θ_+ and θ_- are unknown to the learning algorithm. Examples are generated by choosing either a positive point from $\mathcal{D}_+$ or a negative point from $\mathcal{D}_-$, each with probability 1/2. In this case, the Bayes-optimal hypothesis is the linear separator defined by the hyperplane bisecting and orthogonal to the line segment $\theta_+\theta_-$.

This parametric model is less rigid than our "PAC with compatibility" setting in the sense that it incorporates noise: even the Bayes-optimal hypothesis is not a perfect classifier. On the other hand, it is significantly more restrictive in that the underlying probability distribution is effectively forced to commit to the target concept. If we consider the class C of all linear separators, then really only two concepts in C are "compatible" with the underlying distribution on unlabeled examples: namely, the Bayes-optimal one and its negation. In other words, if we knew the underlying distribution, then there are only two possible target concepts left. Given this view, it is not surprising that unlabeled data can be so helpful under this set of assumptions. Our proposal of a compatibility function between a concept and a probability distribution is an attempt to more broadly consider distributions that do not completely commit to a target function and yet are not completely uncommitted either.

A second approach to using unlabeled data, given by Yarowsky [15] in the context of the "word sense disambiguation" problem is much closer in spirit to co-training, and can be nicely viewed in our model. The problem Yarowsky considers is the following. Many words have several quite different dictionary definitions. For instance, "plant" can mean a type of life form or a factory. Given a text document and an instance of the word "plant" in it, the goal of the algorithm is to determine which meaning is intended. Yarowsky [15] makes use of unlabeled data via the following observation: within any fixed document, it is highly likely that all instances of a word like "plant" have the *same* intended meaning, whichever meaning that happens to be. He then uses this observation, together with a learning algorithm that learns to make predictions based on local context, to achieve good results with only a few labeled examples and many unlabeled ones.

We can think of Yarowsky's approach in the context of co-training as follows. Each example (an instance of the word "plant") is described using two distinct representations. The first representation is the unique-ID of the document that the word is in. The second representation is the local context surrounding the word. (For instance, in the bipartite graph view, each node on the left represents a document, and its degree is the number of instances of "plant" in that document; each node on the right represents a different local context.) The assumptions that any two instances of "plant" in the same document have the same label, and that local context is also sufficient for determining a word's meaning, are equivalent to our assumption that all examples in the same connected component must have the same classification.

4 ROTE LEARNING

In order to get a feeling for the co-training model, we consider in this section the simple problem of rote learning. In particular, we consider the case that $C_1 = 2^{X_1}$ and $C_2 = 2^{X_2}$, so all partitions consistent with $\mathcal{D}$ are possible, and we have a learning algorithm that simply outputs "I don't know" on any example whose label it cannot deduce from its training data and the compatibility assumption. Let $|X_1| = |X_2| = N$, and imagine that N is a "medium-size" number in the sense that gathering $O(N)$ unlabeled examples is feasible but labeling them all is not.[1] In this case, given just a single view (i.e., just the X_1 portion), we might need to see $\Omega(N)$ labeled examples in order to cover a substantial fraction of $\mathcal{D}$. Specifically, the probability that the $(m+1)$st example has not yet been seen is

$$\sum_{x_1 \in X_1} \Pr_{\mathcal{D}}[x_1](1 - \Pr_{\mathcal{D}}[x_1])^m.$$

If, for instance, each example has the same probability under $\mathcal{D}$, our rote-learner will need $\Omega(N)$ labeled examples in order to achieve low error.

On the other hand, the two views we have of each example allow a potentially much smaller number of labeled examples to be used if we have a large unlabeled sample. For instance, suppose at one extreme that our unlabeled sample contains every edge in the graph $G_{\mathcal{D}}$ (every example with nonzero probability). In this case, our rote-learner will be confident about the label of a new example exactly when it has previously seen a labeled example in the same connected component of $G_{\mathcal{D}}$. Thus, if the connected components in $G_{\mathcal{D}}$ are $c_1, c_2, \ldots$, and have probability mass $P_1, P_2, \ldots$, respectively, then the probability that given m labeled examples, the label of an $(m+1)$st example cannot be deduced by the algorithm is just

$$\sum_{c_j \in G_{\mathcal{D}}} P_j(1 - P_j)^m. \quad (1)$$

For instance, if the graph $G_{\mathcal{D}}$ has only k connected components, then we can achieve error ϵ with at most $O(k/\epsilon)$ examples.

More generally, we can use the two views to achieve a tradeoff between the number of labeled and unlabeled examples needed. If we consider the graph G_S (the graph with one edge for each observed example), we can see that as we observe more unlabeled examples, the number of connected components will drop as components merge together, until finally they are the same as the components of $G_{\mathcal{D}}$. Furthermore, for a given set S, if we now select a random subset of m of them to label, the probability that the label of a random $(m+1)$st example chosen from the remaining portion of S cannot be deduced by the algorithm is

$$\sum_{c_j \in G_S} \frac{s_j \binom{|S|-s_j}{m}}{\binom{|S|}{m+1}},$$

where s_j is the number of edges in component c_j of S. If $m \ll |S|$, the above formula is approximately

$$\sum_{c_j \in G_S} \frac{s_j}{|S|} \left(1 - \frac{s_j}{|S|}\right)^m,$$

in analogy to Equation 1.

In fact, we can use recent results in the study of random graph processes [9] to describe quantitatively how we expect the components in G_S to converge to those of $G_{\mathcal{D}}$ as we see more unlabeled examples, based on properties of the distribution $\mathcal{D}$. For a given connected component H of $G_{\mathcal{D}}$, let α_H be the value of the minimum cut of H (the minimum, over all cuts of H, of the sum of the weights on the edges in the cut). In other words, α_H is the probability that a random example will cross this specific minimum cut. Clearly, for our sample S to contain a spanning tree of H, and therefore to include all of H as one component, it must have at least one edge in that minimum cut. Thus, the expected number of unlabeled samples needed for this to occur is at least $1/\alpha_H$. Of course, there are many cuts in H and to have a spanning tree one must include at least one edge from every cut. Nonetheless, Karger [9] shows that this is nearly sufficient as well. Specifically, Theorem 2.1 of [9] shows that $O((\log N)/\alpha_H)$ unlabeled samples are sufficient to ensure that a spanning tree is found with high probability.[2] So, if $\alpha = \min_H\{\alpha_H\}$, then $O((\log N)/\alpha)$ unlabeled samples are sufficient to ensure that the number of connected components in our sample is equal to the number in $\mathcal{D}$, minimizing the number of labeled examples needed.

For instance, suppose $N/2$ points in X_1 are positive and $N/2$ are negative, and similarly for X_2, and the distribution $\mathcal{D}$ is uniform subject to placing zero probability on illegal examples. In this case, each legal example has probability $p = 2/N^2$. To reduce the observed graph to two connected components we do not need to see all $O(N^2)$ edges, however. All we need are two spanning trees. The minimum cut for each component has value $pN/2$, so by Karger's result, $O(N \log N)$ unlabeled examples suffice. (This simple case can be analyzed easily from first principles as well.)

More generally, we can bound the number of connected components we expect to see (and thus the number of labeled examples needed to produce a perfect hypothesis if we imagine the algorithm is allowed to select which unlabeled examples will be labeled) in terms of the number of unlabeled examples m_u as follows. For a given $\alpha < 1$, consider a greedy process in which any cut of value less that α in $G_{\mathcal{D}}$ has all its edges removed, and this process is then repeated until no connected component has such a cut. Let $N_{CC}(\alpha)$ be the number of connected components remaining. If we let $\alpha = c\log(N)/m_u$, where c is the constant from Karger's theorem, and if m_u is large enough so that there are no singleton components (components having no edges) remaining after the above process, then $N_{CC}(\alpha)$ is an

[1] To make this more plausible in the context of web pages, think of x_1 as not the document itself but rather some small set of attributes of the document.

[2] This theorem is in a model in which each edge e *independently* appears in the observed graph with probability mp_e, where p_e is the weight of edge e and m is the expected number of edges chosen. (Specifically, Karger is concerned with the network reliability problem in which each edge goes "down" independently with some known probability and you want to know the probability that connectivity is maintained.) However, it is not hard to convert this to the setting we are concerned with, in which a fixed m samples are drawn, each independently from the distribution defined by the p_e's. In fact, Karger in [10] handles this conversion formally.

upper bound on the expected number of labeled examples needed to cover all of $\mathcal{D}$. On the other hand, if we let $\alpha = 1/(2m_u)$, then $\frac{1}{2}N_{CC}(\alpha)$ is a lower bound since the above greedy process must have made at most $N_{CC} - 1$ cuts, and for each one the expected number of edges crossing the cut is at most $1/2$.

5 LEARNING IN LARGE INPUT SPACES

In the previous section we saw how co-training could provide a tradeoff between the number of labeled and unlabeled examples needed in a setting where $|X|$ is relatively small and the algorithm is performing rote-learning. We now move to the more difficult case where $|X|$ is large (e.g., $X_1 = X_2 = \{0,1\}^n$) and our goal is to be polynomial in the description length of the examples and the target concept.

What we show is that given a *conditional independence* assumption on the distribution $\mathcal{D}$, if the target class is learnable from random classification noise in the standard PAC model, then any initial weak predictor can be boosted to arbitrarily high accuracy using *unlabeled examples only* by co-training.

Specifically, we say that target functions f_1, f_2 and distribution $\mathcal{D}$ together satisfy the *conditional independence* assumption if, for any fixed $(\hat{x}_1, \hat{x}_2) \in X$ of non-zero probability,

$$\Pr_{(x_1,x_2)\in\mathcal{D}}\Big[x_1 = \hat{x}_1 \mid x_2 = \hat{x}_2\Big] = \Pr_{(x_1,x_2)\in\mathcal{D}}\Big[x_1 = \hat{x}_1 \mid f_2(x_2) = f_2(\hat{x}_2)\Big],$$

and similarly,

$$\Pr_{(x_1,x_2)\in\mathcal{D}}\Big[x_2 = \hat{x}_2 \mid x_1 = \hat{x}_1\Big] = \Pr_{(x_1,x_2)\in\mathcal{D}}\Big[x_2 = \hat{x}_2 \mid f_1(x_1) = f_1(\hat{x}_1)\Big].$$

In other words, x_1 and x_2 are conditionally independent given the label. For instance, we are assuming that the words on a page P and the words on hyperlinks pointing to P are conditionally independent given the classification of P. This seems to be a somewhat plausible starting point given that the page itself is constructed by a *different user than the one who made the link*. On the other hand, Theorem 1 below can be viewed as showing why this is not really so plausible after all.[3]

In order to state the theorem, we define a "weakly-useful predictor" h of a function f to be a function such that

1. $\Pr_{\mathcal{D}}\Big[h(x) = 1\Big] \geq \epsilon$, and
2. $\Pr_{\mathcal{D}}\Big[f(x) = 1 | h(x) = 1\Big] \geq \Pr_{\mathcal{D}}\Big[f(x) = 1\Big] + \epsilon$,

for some $\epsilon > 1/poly(n)$. For example, seeing the word "handouts" on a web page would be a weakly-useful predictor that the page is a course homepage if (1) "handouts" appears on a non-negligible fraction of pages, and (2) the probability a given page is a course homepage given that "handouts" appears is non-negligibly higher than the probability without that word. If f is unbiased in the sense that $\Pr_{\mathcal{D}}(f(x) = 1) = \Pr_{\mathcal{D}}(f(x) = 0) = 1/2$, then this is the same as the usual notion of a weak predictor, namely $\Pr_{\mathcal{D}}(h(x) = f(x)) \geq 1/2 + \epsilon$. Otherwise, it is equivalent (assuming that f is not overwhelmingly often 0 or 1) to the statement that h is a weak predictor over the distribution "normalized" to make f appear unbiased.

Theorem 1 *If C_2 is learnable in the PAC model with classification noise, and if the conditional independence assumption is satisfied, then (C_1, C_2) is learnable in the Co-training model from unlabeled data only, given an initial weakly-useful predictor $h(x_1)$.*

Thus, for instance, the conditional independence assumption implies that any concept class learnable in the Statistical Query model [11] is learnable from unlabeled data and an initial weakly-useful predictor.

Before proving the theorem, it will be convenient to define a variation on the standard classification noise model where the noise rate on positive examples may be different from the noise rate on negative examples. Specifically, let (α, β) classification noise be a setting in which true positive examples are incorrectly labeled (independently) with probability α, and true negative examples are incorrectly labeled (independently) with probability β. In this case we have the following simple lemma:

Lemma 1 *If concept class C is learnable in the classification noise model, then it is also learnable with (α, β) classification noise so long as $\alpha + \beta < 1$ (running time is polynomial in $1/(1 - \alpha - \beta)$).*

Proof. First, suppose α and β are known to the learning algorithm. Without loss of generality, assume $\alpha < \beta$. To learn C with (α, β) noise, simply flip each positive label to a negative label independently with probability $(\beta - \alpha)/(\beta + (1 - \alpha))$. This results in standard classification noise with noise rate $\nu = \beta/(\beta + (1 - \alpha)) = \beta/(2\beta + \epsilon)$, where $\epsilon = 1 - (\alpha + \beta)$.

If α and β are not known, this can be dealt with in the usual way. For instance, given a data set S of m examples of which m_+ are labeled positive, we can create $m + 1$ hypotheses, where hypothesis i ($0 \leq i \leq m_+$) is produced by flipping the labels on i random positive examples in S and running the classification noise algorithm, and hypothesis j ($m_+ < j \leq m$) is produced by flipping the labels on j random negative examples in S and then running the algorithm. These hypotheses

[3] Using our bipartite graph view from Section 2.1, it is easy to see that for this distribution $\mathcal{D}$, the only "compatible" target functions are the pair (f_1, f_2), its negation, and the all-positive and all-negative functions (assuming $\mathcal{D}$ does not give probability zero to any example). Theorem 1 can be interpreted as showing how, given access to $\mathcal{D}$ and a slight bias towards (f_1, f_2), the unlabeled data can be used in polynomial time to discover this fact.

can then be evaluated on a separate test set. We expect at least one hypothesis to be good since the procedure when α and β are known can be viewed as a probability distribution over these $m+1$ experiments. ■

The (α, β) classification noise model can be thought of as a kind of constant-partition classification noise [2]. However, the results in [2] require that each noise rate be less than 1/2. We will need the stronger statement presented here, namely that it suffices to assume only that the sum of α and β is less than 1.

Proof of Theorem 1. Let $f(x)$ be the target concept and $p = \Pr_{\mathcal{D}}(f(x)=1)$ be the probability that a random example from $\mathcal{D}$ is positive. Let $q = \Pr_{\mathcal{D}}(f(x) = 1|h(x_1)=1)$ and let $c = \Pr_{\mathcal{D}}(h(x_1)=1)$. So,

$$\begin{aligned}
&\Pr_{\mathcal{D}}\Big[h(x_1)=1|f(x)=1\Big] \\
&\quad = \frac{\Pr_{\mathcal{D}}\big[f(x)=1|h(x_1)=1\big]\Pr_{\mathcal{D}}\big[h(x_1)=1\big]}{\Pr_{\mathcal{D}}\big[f(x)=1\big]} \\
&\quad = \frac{qc}{p} \qquad (2)
\end{aligned}$$

and

$$\Pr_{\mathcal{D}}\Big[h(x_1)=1|f(x)=0\Big] \;=\; \frac{(1-q)c}{1-p}. \qquad (3)$$

By the conditional independence assumption, for a random example $x = (x_1, x_2)$, $h(x_1)$ is independent of x_2 given $f(x)$. Thus, if we use $h(x_1)$ as a noisy label of x_2, then this is equivalent to (α, β)-classification noise, where $\alpha = 1 - qc/p$ and $\beta = (1-q)c/(1-p)$ using equations (2) and (3). The sum of the two noise rates satisfies

$$\alpha + \beta \;=\; 1 - \frac{qc}{p} + \frac{(1-q)c}{1-p} \;=\; 1 - c\left(\frac{q-p}{p(1-p)}\right).$$

By the assumption that h is a weakly-useful predictor, we have $c \geq \epsilon$ and $q - p \geq \epsilon$. Therefore, this quantity is at most $1 - \epsilon^2/(p(1-p))$, which is at most $1 - 4\epsilon^2$. Applying Lemma 1, we have the theorem. ■

6 EXPERIMENTS

In order to test the idea of co-training, we applied it to the problem of learning to classify web pages. This particular experiment was motivated by a larger research effort [1] to apply machine learning to the problem of extracting information from the world wide web.

The data for this experiment[4] consists of 1051 web pages collected from Computer Science department web sites at four universities: Cornell, University of Washington, University of Wisconsin, and University of Texas. These pages have been hand labeled into a number of categories. For our experiments we considered the category "course home page" as the target function; thus, course home pages are the positive examples and all other pages are negative examples. In this dataset, 22% of the web pages were course pages.

For each example web page x, we considered x_1 to be the bag (multi-set) of words appearing on the web page, and x_2 to be the bag of words underlined in all links pointing into the web page from other pages in the database. Classifiers were trained separately for x_1 and for x_2, using the naive Bayes algorithm. We will refer to these as the page-based and the hyperlink-based classifiers, respectively. This naive Bayes algorithm has been empirically observed to be successful for a variety of text-categorization tasks [12].

The co-training algorithm we used is described in Table 1. Given a set L of labeled examples and a set U of unlabeled examples, the algorithm first creates a smaller pool U' containing u unlabeled examples. It then iterates the following procedure. First, use L to train two distinct classifiers: h_1 and h_2. h_1 is a naive Bayes classifier based only on the x_1 portion of the instance, and h_2 is a naive Bayes classifier based only on the x_2 portion. Second, allow each of these two classifiers to examine the unlabeled set U' and select the p examples it most confidently labels as positive, and the n examples it most confidently labels negative. We used $p = 1$ and $n = 3$, to match the ratio of positive to negative examples in the underlying data distribution. Each example selected in this way is added to L, along with the label assigned by the classifier that selected it. Finally, the pool U' is replenished by drawing $2p + 2n$ examples from U at random. In earlier implementations of Co-training, we allowed h_1 and h_2 to select examples directly from the larger set U, but have obtained better results when using a smaller pool U', presumably because this forces h_1 and h_2 to select examples that are more representative of the underlying distribution $\mathcal{D}$ that generated U.

Experiments were conducted to determine whether this co-training algorithm could successfully use the unlabeled data to outperform standard supervised training of naive Bayes classifiers. In each experiment, 263 (25%) of the 1051 web pages were first selected at random as a test set. The remaining data was used to generate a labeled set L containing 3 positive and 9 negative examples drawn at random. The remaining examples that were not drawn for L were used as the unlabeled pool U. Five such experiments were conducted using different training/test splits, with Co-training parameters set to $p = 1$, $n = 3$, $k = 30$ and $u = 75$.

To compare Co-training to supervised training, we trained naive Bayes classifiers that used only the 12 labeled training examples in L. We trained a hyperlink-based classifier and a page-based classifier, just as for co-training. In addition, we defined a third combined classifier, based on the outputs from the page-based and hyperlink-based classifier. In keeping with the naive Bayes assumption of conditional independence, this combined classifier computes the probability $P(c_j|x)$ of class c_j given the instance $x = (x_1, x_2)$ by multiplying the probabilities output by the page-based and hyperlink-

[4]This data is available at http://www.cs.cmu.edu/afs/cs/project/theo-11/www/wwkb/

Given:

- a set L of labeled training examples
- a set U of unlabeled examples

Create a pool U' of examples by choosing u examples at random from U
Loop for k iterations:

Use L to train a classifier h_1 that considers only the x_1 portion of x
Use L to train a classifier h_2 that considers only the x_2 portion of x
Allow h_1 to label p positive and n negative examples from U'
Allow h_2 to label p positive and n negative examples from U'
Add these self-labeled examples to L
Randomly choose $2p + 2n$ examples from U to replenish U'

Table 1: The Co-Training algorithm. In the experiments reported here both h_1 and h_2 were trained using a naive Bayes algorithm, and algorithm parameters were set to $p = 1$, $n = 3$, $k = 30$ and $u = 75$.

	Page-based classifier	Hyperlink-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

Table 2: Error rate in percent for classifying web pages as course home pages. The top row shows errors when training on only the labeled examples. Bottom row shows errors when co-training, using both labeled and unlabeled examples.

based classifiers:

$$P(c_j|x) \leftarrow P(c_j|x_1)P(c_j|x_2)$$

The results of these experiments are summarized in Table 2. Numbers shown here are the test set error rates averaged over the five random train/test splits. The first row of the table shows the test set accuracies for the three classifiers formed by supervised learning; the second row shows accuracies for the classifiers formed by co-training. Note that for this data the default hypothesis that always predicts "negative" achieves an error rate of 22%. Figure 2 gives a plot of error versus number of iterations for one of the five runs.

Notice that for all three types of classifiers (hyperlink-based, page-based, and combined), the co-trained classifier outperforms the classifier formed by supervised training. In fact, the page-based and combined classifiers achieve error rates that are half the error achieved by supervised training. The hyperlink-based classifier is helped less by co-training. This may be due to the fact that hyperlinks contain fewer words and are less capable of expressing an accurate approximation to the target function.

This experiment involves just one data set and one target function. Further experiments are needed to determine the general behavior of the co-training algorithm, and to determine what exactly is responsible for the pattern of behavior observed. However, these results do indicate that co-training can provide a useful way of taking advantage of unlabeled data.

7 CONCLUSIONS AND OPEN QUESTIONS

We have described a model in which unlabeled data can be used to augment labeled data, based on having two views (x_1, x_2) of an example that are redundant but not completely correlated. Our theoretical model is clearly an over-simplification of real-world target functions and distributions. In particular, even for the optimal pair of functions $f_1, f_2 \in C_1 \times C_2$ we would expect to occasionally see inconsistent examples (i.e., examples (x_1, x_2) such that $f_1(x_1) \neq f_2(x_2)$). Nonetheless, it provides a way of looking at the notion of the "friendliness" of a distribution (in terms of the components and minimum cuts) and at how unlabeled examples can potentially be used to prune away "incompatible" target concepts to reduce the number of labeled examples needed to learn. It is an open question to what extent the consistency constraints in the model and the mutual independence assumption of Section 5 can be relaxed and still allow provable results on the utility of co-training from unlabeled data. The preliminary experimental results presented suggest that this method of using unlabeled data has a potential for significant benefits in practice, though further studies are clearly needed.

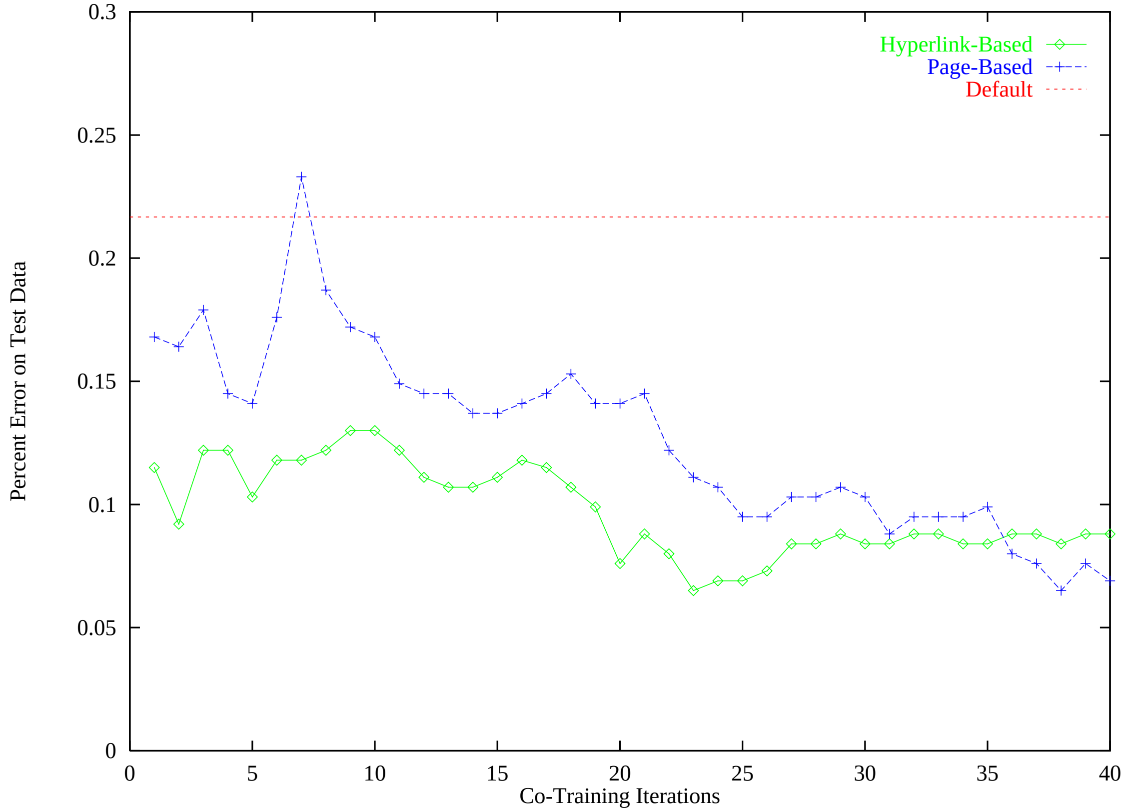

Figure 2: Error versus number of iterations for one run of co-training experiment.

We conjecture that there are many practical learning problems that fit or approximately fit the co-training model. For example, consider the problem of learning to classify segments of television broadcasts [7, 14]. We might be interested, say, in learning to identify televised segments containing the US President. Here X_1 could be the set of possible video images, X_2 the set of possible audio signals, and X their cross product. Given a small sample of labeled segments, we might learn a weakly predictive recognizer h_1 that spots full-frontal images of the president's face, and a recognizer h_2 that spots his voice when no background noise is present. We could then use co-training applied to the large volume of unlabeled television broadcasts, to improve the accuracy of both classifiers. Similar problems exist in many perception learning tasks involving multiple sensors. For example, consider a mobile robot that must learn to recognize open doorways based on a collection of vision (X_1), sonar (X_2), and laser range (X_3) sensors. The important structure in the above problems is that each instance x can be partitioned into subcomponents x_i, where the x_i are not perfectly correlated, where each x_i can in principle be used on its own to make the classification, and where a large volume of unlabeled instances can easily be collected.

References

[1] M. Craven, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and C.Y. Quek. Learning to extract symbolic knowledge from the world wide web. Technical report, Carnegie Mellon University, January 1997.

[2] S. E. Decatur. PAC learning with constant-partition classification noise and applications to decision tree induction. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 83–91, July 1997.

[3] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[4] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

[5] Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via an EM approach. In *Advances in Neural Information Processing Systems (NIPS 6)*. Morgan Kauffman, 1994.

[6] S. A. Goldman and M. J. Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31, February 1995.

[7] A.G. Hauptmann and M.J. Witbrock. Informedia: News-on-demand - multimedia information acquisition and retrieval. In M. Maybury, editor, *Intelligent Multimedia Information Retrieval*, 1997.

[8] J. Jackson and A. Tomkins. A computational model of teaching. In *Proceedings of the Fifth Annual Workshop on Computational Learning The-*

ory, pages 319–326. Morgan Kaufmann, 1992.

[9] D. R. Karger. Random sampling in cut, flow, and network design problems. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing*, pages 648–657, May 1994.

[10] D. R. Karger. Random sampling in cut, flow, and network design problems. Journal version draft, 1997.

[11] M. Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 392–401, 1993.

[12] D. D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *Third Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, 1994.

[13] Joel Ratsaby and Santosh S. Venkatesh. Learning from a mixture of labeled and unlabeled examples with parametric side information. In *Proceedings of the 8th Annual Conference on Computational Learning Theory*, pages 412–417. ACM Press, New York, NY, 1995.

[14] M.J. Witbrock and A.G. Hauptmann. Improving acoustic models by watching television. Technical Report CMU-CS-98-110, Carnegie Mellon University, March 19 1998.

[15] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.