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
Tom.Mitchell@cmu.edu

February 10, 1998

Abstract

Motivated by the problem of learning to classify web pages, we consider a learning setting in which each example can be described using two separate "kinds" of features. We assume that either way of viewing examples would be sufficient for learning if we had enough labeled data, but are interested in the possibility of using both views together to allow cheaper *unlabeled* data to be used when only a much smaller set of labeled examples is available. In particular, the presence of two distinct views of each example suggests a learning strategy in which two 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 framework and analysis for this setting. We also give preliminary empirical results on real web-page data that suggest that this use of unlabeled examples can potentially lead to significant improvement of hypotheses in practice.

1 Introduction

In many machine learning settings, unlabeled examples are significantly easier to come by than labeled ones. One example of this is web-page classification. Suppose that we want a program to electronically visit some organization and download all the web pages of interest to us. For instance, we might want to find all the CS faculty member pages, or all the course home pages at some university [1]. In this case, to train our learning algorithm, unlabeled examples can be cheaply gathered using a web crawler, but labeling them is more expensive, requiring human input. Therefore, we would like our learning algorithm to be able to take as much advantage of the unlabeled data as possible.

The web-page learning problem has, in addition, another interesting feature: each example can naturally be described using several different "kinds" of information. For instance,

*Submitted (1/98) to the *1998 Conference on Computational Learning Theory (COLT-98).* This research was supported in part by the Darpa HPKB program under contract F30602-97-1-0215.

one kind of information about a web page is the text appearing on the document itself. Another kind of information is the text attached to links pointing *to* this document from other documents on the web.

The two issues mentioned above (labeled versus unlabeled data, and having 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, perhaps the phrase "research interests" on a web page is a weak indicator that the page is a faculty home page, and perhaps the phrase "my advisor is" 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 is" 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, [3]. The question this raises is: is there any reason to believe co-training will actually 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 5) that seem encouraging in this context.

2 A formal framework

We consider 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 for some pair of target functions $f_1 \in C_1$, $f_2 \in C_2$, for any labeled example $((x_1, x_2), \ell)$ observed, we have $f_1(x_1) = f_2(x_2) = \ell$. That is, the labels are consistent with target functions defined over each view separately. This means in particular that $\mathcal{D}$ assigns probability zero to any example (x_1, x_2) such that $f_1(x_1) \neq f_2(x_2)$. We will use f to denote the combined target concept over the entire example; i.e., if we are given example $x = (x_1, x_2)$ with label ℓ, then $f(x) = f_1(x_1) = f_2(x_2) = \ell$.

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 pair of target functions $(f_1, f_2) \in C_1 \times C_2$ as being "natural" for $\mathcal{D}$ 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, f_1 and f_2 are natural for $\mathcal{D}$ if f_1, f_2, and $\mathcal{D}$ are legal together in our framework. Thus, even if C_1 and C_2 are large concept classes with high complexity in, say, the VC-dimension sense, for a given distribution $\mathcal{D}$ the set of *natural* target concept pairs 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 pairs are natural, 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.

For instance, 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 a "bad" 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 $\mathcal{D}$ is more flexible, then perhaps it does. For instance, if x_2 is conditionally independent of x_1 given the label, then x_2 is a *random* negative example of f_2. We explore a generalization of this idea in Section 4 below, 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 [4, 5] in which examples are being supplied by a helpful oracle.

3 A bipartite graph representation

One way to look at the co-training problem is through the following weighted bipartite graph, which we write as $G(X_1, X_2, \mathcal{D})$, or just $G(\mathcal{D})$ if X_1 and X_2 are clear from context. The left-hand side of G has one node for each point in X_1 and the right-hand side of G 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, let us now remove any vertex of degree 0, corresponding to those views having zero probability.

In this representation, the "natural" concepts in C are exactly those corresponding to a partition of this graph with no cross-edges.[1] Given a set of unlabeled examples S, we can similarly define $G(S)$ 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.

Given the above description as a bipartite graph, one first case to consider is that where $C_1 = 2^{X_1}$ and $C_2 = 2^{X_2}$. In other words, all partitions consistent with $\mathcal{D}$ are possible, and the algorithm is reduced to rote-learning. In particular, imagine that $|X_1| = |X_2| = N$, where N is a "medium-size" number in the sense that gathering $O(N)$ unlabeled examples is feasible but labeling them all is not. In this case, given just a single view (i.e., just the X_1 portion), the number of labeled examples needed to cover a substantial fraction of $\mathcal{D}$ could be as high as $O(N)$, and more generally is roughly the value m such that the set of examples whose individual probabilities are at least $1/m$ has substantial probability. (More precisely, the probability that the $(m+1)$st example has not yet been seen is $\sum_{x_1} \Pr_{\mathcal{D}}(x_1)(1 - \Pr_{\mathcal{D}}(x_1))^m$.)

On the other hand, the two views we have of the examples allow a tradeoff between the number of labeled and the number of unlabeled examples needed. In particular, suppose

[1] One could also reasonably define the extent to which a partition is not natural as the weight of the cut it induces in G.

our *unlabeled* sample is sufficient to cover the distribution in the sense that the probability that a random example from $\mathcal{D}$ does not "touch" any point in S (neither its first component nor its second component agrees with any example in S) is at most ϵ. In this case, from our observation that two examples in the same connected component must have the same label, it suffices to request the label of just one example per connected component of $G(S)$, since that label can then be propagated to the other examples in its component through co-training. (We are imagining now that the algorithm is allowed to select which unlabeled examples it wishes to have labeled, somewhat like the setting of the Query-by-Committee approach [10].)

Notice that as we see more unlabeled examples, we expect connected components in our sample S to merge, until eventually these are the same as the connected components of $\mathcal{D}$. Thus we reduce the number of labeled examples needed.

It turns out that we can use recent results in the study of random graph processes to quantify this tradeoff. For simplicity, let us now assume that our goal is to cover *all* of $\mathcal{D}$, not just a $1-\epsilon$ fraction. 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 [6] shows that this is nearly sufficient as well. Specifically, Theorem 2.1 of [6] 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, if $N/2$ points in X_1 are positive and $N/2$ are negative, and the same holds for X_2, and the distribution $\mathcal{D}$ is uniform subject to placing zero probability on illegal examples, then 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, we just need spanning trees. The minimum cut for each component has value $pN/2$, so $O(N \log N)$ unlabeled examples suffice. (This simple case can be analyzed easily from first principles as well.)

More generally, we can quantify the number of labeled examples needed in terms of the number of unlabeled examples m 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 $\alpha = c\log(N)/m$, where c is the constant from Karger's theorem, and if m 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 upper

[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 [7] handles this conversion formally.

bound on the expected number of labeled examples needed to cover all of $\mathcal{D}$. On the other hand, if we let $\alpha = 1/(2m)$, 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.

4 Conditional independence and classification noise

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 will 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 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,

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

and similarly,

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

In other words, in the context of our web-page example, if we see a link with the text "my advisor is" pointing to a given page, and let's say that we know that this means the page must be a positive example of a faculty home page, then the text on the page is that of a *random* positive example. 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. $\Pr_{\mathcal{D}}(h(x) = 1) \geq \epsilon$, and
2. $\Pr_{\mathcal{D}}(f(x) = 1 | h(x) = 1) \geq \Pr_{\mathcal{D}}(f(x) = 1) + \epsilon$,

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

[3]Using our bipartite graph view from Section 3, it is easy to see that for this distribution $\mathcal{D}$, the only "natural" target functions are the pair (f_1, f_2), their 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.

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.

Finally, for convenience, let us assume that $C_1 = C_2$, and we will call this concept class C.

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

Thus, for instance, any concept class learnable in the Statistical Query model [8] is learnable from unlabeled data and an initial weakly-useful predictor, assuming the conditional independence assumption.

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 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], though the model and results in [2] require that each noise rate itself be less than $1/2$. We will need to use the fact that we can handle one noise rate being larger than $1/2$, so long as the sum 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,

$$\Pr_{\mathcal{D}}(h(x_1) = 1 | f(x) = 1) \;=\; \frac{\Pr_{\mathcal{D}}(f(x) = 1 | h(x_1) = 1)\Pr_{\mathcal{D}}(h(x_1) = 1)}{\Pr_{\mathcal{D}}(f(x) = 1)} \;=\; \frac{qc}{p} \tag{1}$$

and

$$\Pr_{\mathcal{D}}(h(x_1) = 1 | f(x) = 0) \;=\; \frac{(1-q)c}{1-p}. \tag{2}$$

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 (1) and (2). 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. ■

5 Experiments

To get an empirical feel for the idea of co-training, and to provide some kind of sanity check, we performed the following simple experiment on the web page data of [1]. This data consists of web pages from Computer Science departments at four universities: Cornell, University of Washington, University of Wisconsin, and University of Texas. These pages have been classified by hand 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. For a given web page x, we considered x_1 to be the bag (multi-set) of words appearing on the page, and x_2 to be the bag of words underlined in links pointing into the page from other pages in the database. The learning algorithm used was the naive-Bayes algorithm, an algorithm that has been empirically observed to be successful for text-categorization tasks [9].

Our experiment used web pages from three of the universities as the training set and used those from the fourth (U. Texas) as a hold-out test set.[4] We began with a set S_ℓ of 4 labeled positive examples and 12 labeled negative examples selected randomly from the training set. This labeled sample was fed into two separate naive-Bayes algorithms (one considering only the x_1 portion of the examples and the other considering only the x_2 portion) producing two hypotheses h_1 and h_2. We then repeated the following procedure. From the remaining (unlabeled) training set we selected the three examples most believed to be negative by h_1

[4] It would better fit our assumptions to draw the hold-out set randomly from the total sample. It just happened to be easier to perform the experiment as stated.

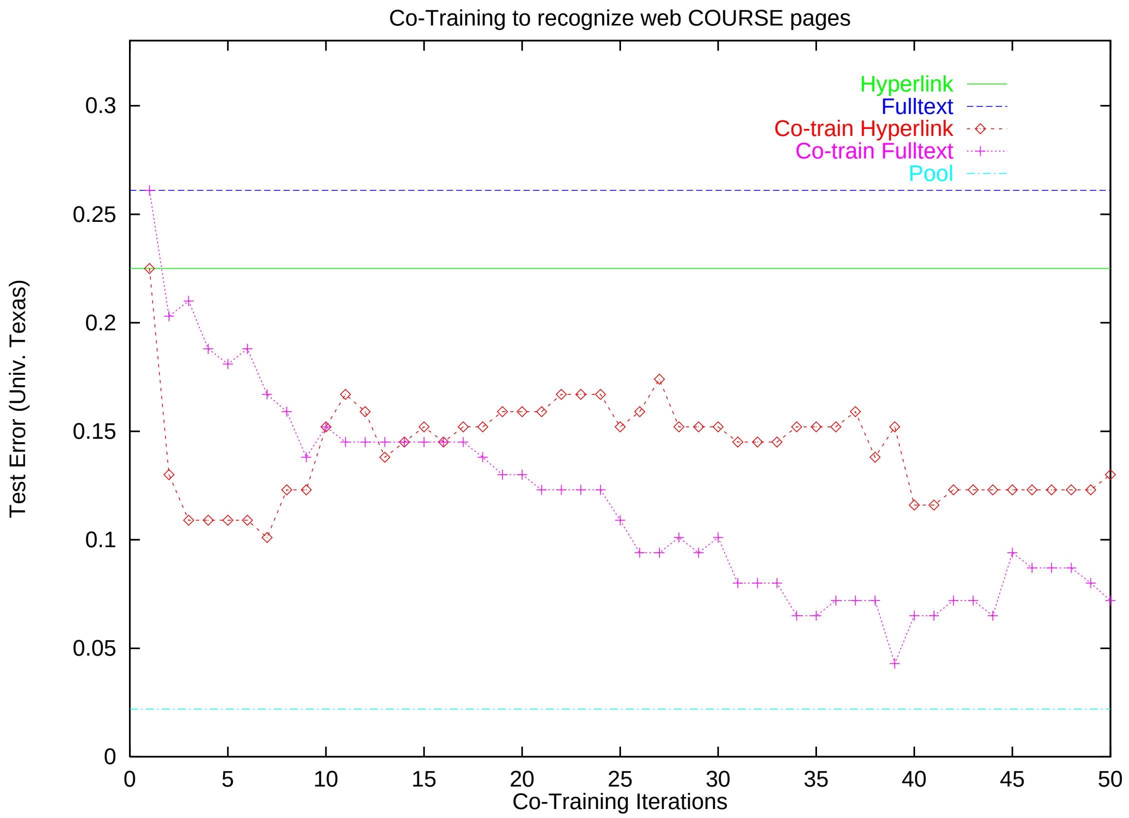

Figure 1: Results of co-training with test error measured according to the U. Texas data set.

and single example most believed to be positive by h_1, then "labeled" them accordingly, and added them to S_ℓ. Similarly we selected the top three "negative" examples and top single "positive" example according to h_2 and added them to S_ℓ. Then the learning algorithms were re-run on the new "labeled" sample, and so on. The reason for the 3:1 ratio was to maintain the negative-to-positive ratio in the sample S_ℓ; it turns out the algorithms were fairly sensitive to this issue.

Results of this experiment, showing the changing performance of the hypotheses as determined by the hold-out Texas data set are pictured in Figure 1. The top dashed line is the error of the initial hypothesis h_1 and the solid line just below it is the error of the initial hypothesis h_2. Notice that as the number of co-training iterations increases, the error of h_1 steadily drops, from 26% to about 7%, and the error of h_2 drops as well, though it appears to level off much more quickly. The lowest dashed line, labeled "pool," is the fraction of the final set S_ℓ that is mislabeled. That is, it represents the fraction of examples added to the "labeled" training set with incorrect labels. This fraction is about 2%.

Note this experiment involves just one data set and one target function. Further experiments are needed to determine the general behavior of the 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.

6 Conclusions and Open questions

Our theoretical model is clearly too simplistic for 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 "unnatural" target concepts to reduce the number of labeled examples needed to learn. One big open question is to what extent the consistency constraints in the model and the mutual independence assumption of Section 4 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 use of unlabeled data has a potential for significant benefits in practice, though further studies are clearly needed.

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] 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.

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

[5] J. Jackson and A. Tomkins. A computational model of teaching. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 319–326. Morgan Kaufmann, 1992.

[6] 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.

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

[8] 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.

[9] 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.

[10] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 287–294. Morgan Kaufmann, 1992.