Enhancing Supervised Learning with Unlabeled Data

Sally Goldman SG@CS.WUSTL.EDU
Yan Zhou ZY@CS.WUSTL.EDU
Department of Computer Science, Washington University, St. Louis, MO 63130 USA

Abstract

In many practical learning scenarios, there is a small amount of labeled data along with a large pool of unlabeled data. Many supervised learning algorithms have been developed and extensively studied. We present a new "co-training" strategy for using unlabeled data to improve the performance of standard supervised learning algorithms. Unlike much of the prior work, such as the co-training procedure of Blum and Mitchell (1998), we do *not* assume there are two redundant views both of which are sufficient for classification. The only requirement our co-training strategy places on each supervised learning algorithm is that its hypothesis partitions the example space into a set of equivalence classes (e.g. for a decision tree each leaf defines an equivalence class). We evaluate our co-training strategy via experiments using data from the UCI repository.

1. Introduction

In many practical learning scenarios, there is a small amount of labeled data along with a large pool of unlabeled data. Many supervised learning algorithms (e.g. ID3 and HOODG) have been developed and extensively studied. Can unlabeled data be used to improve their accuracy? Intuitively, one would expect two different supervised learning algorithms to complement each other since they use different representations for their hypotheses and use the provided labeled data in different ways. Hence, you would expect the two algorithms to "notice" different patterns in the data and thus be able to label some unlabeled data for the other. This is the motivation behind our *co-training* strategy.

In this paper, we present a new co-training strategy for using unlabeled data to improve the performance of standard supervised learning algorithms. Unlike much of the prior work, such as the co-training procedure of Blum and Mitchell (1998), we do *not* assume there are two redundant views both of which are sufficient for classification. In our co-training algorithm, there are two different supervised learning algorithms which are each originally trained on the provided labeled data. Then using statistical techniques, each learner can select some of the unlabeled data to label for the other learner. We repeat this process as long as more data is selected to be labeled. Finally, we combine the two resulting hypotheses to obtain our final hypothesis. The only requirement our co-training strategy places on each supervised learning algorithm is that its hypothesis partitions the example space into a set of equivalence classes (e.g. for a decision tree each leaf defines an equivalence class).

There are two key issues we had to resolve in designing our co-training algorithm. First we present, a simple technique based on statistical confidence intervals, for combining the hypothesis from each of the supervised learning algorithms to obtain our final hypothesis. The aim is for this hypothesis to correctly classify the portions of the instance space that are correctly classified by *either* of the two hypotheses that are being combined. Hence, its error rate can be better than that of both hypotheses that were combined. Second, we present a methodology for deciding when one supervised learning algorithm should label data for the other. That is, how confident should a learning algorithm be about its prediction when it labels data for the other? Since neither of the two learning algorithms are 100% sure about their predictions (in fact, their hypotheses may both have significant error rates), additional care has to be taken to ensure a sufficiently low classification noise rate (i.e. noise in the labels) while the training data sets are expanded.

We evaluate our co-training strategy via experimental results using data from the UCI repository using ID3 (Quinlan, 1986) and HOODG (Kohavi, 1994) as our two supervised learning algorithms. Both ID3 and HOODG are trained using only the labeled data whereas our co-training strategy also makes use of a provided pool of unlabeled data. The results are very promising. Over 24 different runs (across 8 different data sets), our co-training strategy yielded a 27.4% improvement over ID3, a 37.0% improvement over HOODG, and a 15.6% improvement over an algorithm that omnisciently picks the better of ID3 and HOODG for each individual run.

The remaining of this paper is organized as follows. Section 2 outlines the related work of using unlabeled data to improve the performance of the supervised learning algorithms. In Section 3, we present our co-training algorithm and describe the theoretical foundations behind our techniques. Section 4 presents our empirical results. Finally, in Section 5, we present our conclusions and discuss directions for future work.

2. Related Work

Many different approaches have been studied for using unlabeled data for improving the performance of supervised learning algorithms. Most classic methods of learning with unlabeled data use a generative model for the classifier and use Expectation-Maximization (EM) of Dempster, Laird, and Rubin (1977) or other approaches to model the label estimation or parameter estimation for the general model (Nigam, McCallum, Thrun, Mitchell, 2000; Wu & Huang, 1999).

Some work uses the distribution of unlabeled data to define a metric or a kernel function which is used further to perform a sanity check or used for the Support Vector Machine trained from the labeled data (Schuurmans, 1997; Hofmann, 1999; Zhang, 1999). The idea of large margin classification and transductive inference also inspires the use of unlabeled data (Shawe-Taylor, 1999). Large margin classification algorithms favor the decision rules that achieve large classification margins which creates dependencies between the unlabeled data and the parameters of the function class (Jaakkola, Meila, & Jebara, 1999).

Other approaches make use of unlabeled data for probability estimation which, together with the target weight estimated from the labeled data, is substituted for the estimates of the probabilities in a Stastical Query (Kearns, 1993) learning algorithm as done by De Comitè, Denis, and Letouzey (1999).

Another related area of research that has very different goals is that of active learning (Dagan & Engelson, 1995; Liere & Tadepalli, 1997; Lewis, 1995) where the algorithm repeatedly selects an unlabeled example, asks an expert (e.g. a human) to provide the correct label, and then rebuilds its hypothesis. In the field of query learning, unlabeled data are used for label querying (Campbell, Cristianini, & Smola, 2000). An important component of active/query learning is in the selection of the unlabeled example. In our co-training technique, we assume there is no expert available to provide labels. However, some of the techniques we introduce could be useful for active learning.

Our study is built on the initial work performed by Blum and Mitchell (1998). They show that unlabeled data can be used to augment labeled data provided that an instance space can be represented using two different views (i.e. two independent and redundant sets of attributes). For example, if you are learning to classify web pages as high or low quality you could look at the links from the page or the links into the page. They make the strong assumption that either view of the example would be sufficient if there were enough labeled data. As in our work, their goal is to use a large set of unlabeled data to augment a much smaller set of labeled examples. They present a very different co-training strategy for this situation and gave both empirical and theoretical results showing that such a strategy can work under this setting. (See also, Craven et. al (1998) for additional results on using co-training for classifying web pages.) The idea of co-training with the assumption of natural redundancy in the data is also used by Collins and Singer (1999). There has been significant work studying applications to the area of text classification. For example, Riloff and Jones (1999) consider the task of learning to classify a noun phrase as a positive or negative example of a "location." Here, the two redundant sufficient features come from looking at the noun phrase itself and the linguistic context in which the noun phrase appears. Yarowsky (1995) uses a similar co-training approach to disambiguate word sense (e.g. to determine if the word "plant" refers to a manufacturing plant or a botanical plant).

While there are settings such as these in which there are two independent (and sufficiently redundant views), there are also many settings in which such redundant views are not available. In this paper, we present a co-training procedure can be used in general situations where there are not such redundant views.

3. Our Co-training Method

We now describe our co-training algorithm and provide some theoretical basis for our design. We assume that we have two different supervised learning algorithms A and B which both output a hypothesis that defines a partition of the instance space. For example, a decision tree partitions the instance space with one equivalence class defined per leaf. We maintain a set

U of unlabeled data, L of the original labeled data, a set L_A which is data that B labeled for A (initially empty), and a set L_B which is data that A labeled for B (initially empty). We also keep an estimate w_A (respectively, w_B) of the number of examples in L_A (respectively, L_B) that are mislabeled. In making these estimates, we bias them towards overestimating the errors causing our co-training to be conservative in labeling data. Hence, we often refer to our estimates as *conservative estimates*.

Our co-training algorithm repeats the following steps until both L_A and L_B do not change during an iteration. At the start of each iteration, we train algorithm A on the labeled examples $L \cup L_A$ to obtain the hypothesis H_A. Similarly, we train B on $L \cup L_B$ to obtain H_B. Each algorithm considers each of its equivalence classes and decides which ones to use to label data from U for the other algorithm. There are two tests that must be satisfied before labeling data. The first ensures that the equivalence class used to label data has an accuracy that is at least as good as the accuracy of the other hypothesis. The second test is to help prevent a degradation in performance due to the increased noise in the labels. For all data from U that is in any equivalence class from H_A that passes both tests, A labels the data and places it in L_B. B labels data for A in the same manner. This completes one round of our procedure.

Detailed pseudo-code for our combining method is given in Table 1 and detailed pseudo-code for our co-training algorithm is given in Table 2. We now discuss the two key aspects of our co-training algorithm: combining H_A and H_B to get an overall hypothesis and selecting which data each algorithm should label for the other in each round.

3.1 Combining

In this section, we describe how the two hypotheses H_A and H_B are combined. To estimate the accuracy of H_A and H_B we use a 95%-confidence interval for a binomial parameter (e.g. See Larson and Marx (1986)). For each hypothesis and for each equivalence class within the two hypotheses we use 10-fold cross validation to compute how many correct predictions are made. Then we compute the 95%-confidence interval which we denote here by $[\ell, h]$. Then to make a prediction for example x, we compare the $(\ell + h)/2$ of the confidence intervals for A, B, the equivalence class of A that contains x, and the equivalence class of B that contains x. We predict according to the hypothesis that corresponds to the maximum of these four quantities. While sometimes, for example, the mean of the confidence interval for the equivalence class of H_A containing x may be lower than the mean of the confidence interval for the equivalence class of H_B containing x, we still predict according to H_A if H_A has the highest overall confidence since we want to be cautious about not predicting with the better of H_A and H_B. It is only when the confidence interval mean for the equivalence class of H_B containing x is higher than that of H_A's overall hypothesis, that we instead predict according to H_B.

When performing the cross validation, we could either use the originally labeled data or all the labeled data (i.e. including that labeled by the other algorithm). Based on earlier tests, we use the originally labeled data. The advantage of using only the originally labeled data is that the confidence interval will be more accurate since the data has less labeling errors. However, when computing the confidence interval estimate for the equivalence classes, by just using the original labeled data there are often some equivalence classes with no data. If an equivalence class of A (for example) has no data and the mean of A's confidence interval is larger than the maximum of B's overall and leaf confidence intervals by more than 0.1, then we predict according to A regardless of the confidence of B's leaf.

3.2 Choosing What Examples to Label

When should algorithm A take an unlabeled example from U and place it in L_B, labeled according to H_A? Intuitively, A should only consider placing example x in L_B if A's confidence in the validity of its label for x is better than B's confidence in the validity of its label for x AND the amount of data labeled is sufficient to compensate for the increased classification noise it will cause for training in future rounds.

We first address the question of how A will decide which examples it has sufficient confidence in its prediction to label them for B (and vice versa). As with our combining method, we use a 95%-confidence interval for a binomial parameter. First, for the overall hypothesis of algorithm B, H_B, we use L to compute the 95%-confidence interval denoted by $[\ell_B, h_B]$. Next, for each equivalence class z defined by H_A we use L to compute the 95%-confidence interval $[\ell_z, h_z]$. If the high end, h_z of the confidence interval for the equivalence class z of H_A is higher than the low end, ℓ_B of H_B then all examples from U which are in equivalence class z pass our first test for being added to L_B.

We now describe the second test which is designed to control the classification noise rate when labeling examples. This test is based on the following relationship between the hypothesis worst-case accuracy $(1 - \epsilon)$,

Table 1. Our technique to combine H_A and H_B to create our hypothesis.

```
Combine($H_A, H_B$)                    \\$H_A$ and $H_B$ are the hypotheses being combined
    Use $L$ for 10-fold cross validation to compute the 95%-confidence interval $[\ell_A, h_A]$ for $H_A$
    Use $L$ for 10-fold cross validation to compute the 95%-confidence interval $[\ell_B, h_B]$ for $H_B$
    For each example $x$ in the instance space
        Let $z$ be the equivalence class of $H_A$ containing $x$
        Use $L$ for 10-fold cross validation to compute the 95%-confidence interval $[\ell_z, h_z]$ for exs in $z$
        Let $z'$ be the equivalence class of $H_B$ containing $x$
        Use $L$ for 10-fold cross validation to compute the 95%-confidence interval $[\ell_{z'}, h_{z'}]$ for exs in $z'$
        If $(\ell_A + h_A)/2 > (\ell_B + h_B)/2$              \\ $H_A$ is more accurate overall than $H_B$
            If $z$ is empty
                If $((\ell_A + h_A)/2 - \max\{(\ell_B + h_B)/2, (\ell_{z'} + h_{z'})/2\} > 0.1)$ then predict with $H_A(x)$
                Else predict with $H_B(x)$
            Else If $((\ell_A + h_A)/2 > (\ell_{z'} + h_{z'})/2) \vee ((\ell_z + h_z)/2 > (\ell_{z'} + h_{z'})/2)$ then predict with $H_A(x)$
            Else predict with $H_B(x)$
        Else                                         \\ $H_B$ is more accurate overall than $H_A$
            If $z'$ is empty
                If $((\ell_B + h_B)/2 - \max\{(\ell_A + h_A)/2, (\ell_z + h_z)/2\} > 0.1)$ then predict with $H_B(x)$
                Else predict with $H_A(x)$
            Else If $((\ell_B + h_B)/2 > (\ell_z + h_z)/2) \vee ((\ell_{z'} + h_{z'})/2 > (\ell_z + h_z)/2)$ then predict with $H_B(x)$
            Else predict with $H_A(x)$
```

sample size (m) and classification noise rate (η) where we assume that the other parameters are held constant (e.g. Angluin and Laird (1988)):

$$m = \frac{c}{\epsilon^2(1-2\eta)^2}\ \text{, or equivalently}\ \ \epsilon = \sqrt{\frac{c}{m(1-2\eta)^2}}$$

for c a constant. We use this relationship to decide if the amount of additional data labeled is sufficient to compensate for the increase in the classification noise rate. To simplify our computation, we let c in the above formula be 1 and compute the square of the inverse of the error. More specifically, in each co-training round, algorithm A chooses which data to label for algorithm B as follows. For B's current hypothesis we have the following values for m, the number of (labeled) training examples and η, the classification noise rate: $m = |L \cup L_B|$ and our conservative estimate for η is $w_B/|L \cup L_B|$. Hence our estimate for $1/\epsilon_B^2$ is $q_B = |L \cup L_B| \left(1 - 2\left(\frac{2w_B}{|L \cup L_B|}\right)\right)^2$. If our first test (i.e. $h_z > \ell_B$) was passed, then we compute U_z, the set of examples from U that h_A maps to equivalence class z. Also using the low-end of the 95%-confidence interval, ℓ_z, we conservatively estimate the number of examples from U_z that are mislabeled, $w_z = (1-\ell_z)|U_z|$. We can then compute our estimate for what the square of the inverse of the error would be if the examples in U_z were labeled for B:

$$q_z = |L \cup L_B \cup U_z| \left(1 - \frac{2(w_B + w_z)}{|L \cup L_B \cup U_z|}\right)^2.$$

Finally, if $q_z > q_B$, indicating a belief that H_B will be improved if the examples in U_z (as labeled by H_A) are added to L_B, we update L_B and w_B accordingly. The corresponding method is used for B to select data to label for A. Since our co-training procedure enables each algorithm to label a significant amount of data in each round, it tends to require very few iterations.

4. Evaluation

We now describe our experimental results. For our two supervised algorithms (referred to as A and B in our general technique), we used ID3 (Quinlan, 1986), a decision tree algorithm and HOODG (Kohavi, 1994), a decision graph algorithm. These algorithms were selected simply because they were available as part of MLC++. Thus we view them as two arbitrary algorithms which satisfy the property that their hypotheses form a partition of the data.

4.1 ID3 and HOODG

ID3 (Quinlan, 1986) is a top-down induction decision tree algorithm. The criteria for splitting the tree is

Table 2. Our co-training method.

```
\\ ———— Initialization ————
$L$ = given labeled data set
$U$ = given unlabeled data set
$L_A = \emptyset$        \\ data labeled by $B$ for use by $A$
$w_A = 0$        \\ conservative estimate for # of mislabeled examples in $L_A$
$L_B = \emptyset$        \\ data labeled by $A$ for use by $B$
$w_B = 0$        \\ conservative estimate for # of mislabeled examples in $L_B$

\\ ———— Main Co-Training Loop ————
Use $L$ for 10-fold cross validation to compute the 95%-confidence interval $[\ell_A, h_A]$ for $H_A$
Use $L$ for 10-fold cross validation to compute the 95%-confidence interval $[\ell_B, h_B]$ for $H_B$
Repeat until $L_A$ and $L_B$ do not change        \\each iteration is a co-training round
    Run algorithm $A$ on labeled data $L \cup L_A$ to obtain hypothesis $H_A$
    $q_A = |L \cup L_A| \left(1 - 2\left(\frac{2w_A}{|L \cup L_A|}\right)\right)^2$        \\conservative estimate for $(1/\epsilon_A)^2$
    Run algorithm $B$ on labeled data $L \cup L_B$ to obtain hypothesis $H_B$
    $q_B = |L \cup L_B| \left(1 - 2\left(\frac{2w_B}{|L \cup L_B|}\right)\right)^2$        \\conservative estimate for $(1/\epsilon_B)^2$
    \\——— Choose data for $A$ to label for $B$ ———
    For each equivalent class $z$ defined by $H_A$
    let $U_z$ be the examples from $U$ that map to $z$
        Use $L$ for 10-fold cross validation to compute the 95%-confidence interval $[\ell_z, h_z]$ for $z$
        if $h_z > \ell_B$        \\First test passed for labeling examples in $U_z$ for $B$
            $w_z = (1 - \ell_z)|U_z|$        \\conservative estimate for # exs in $U_z$ mislabeled by $H_A$
            $q_z = |L \cup L_B \cup U_z| \left(1 - \frac{2(w_B + w_z)}{|L \cup L_B \cup U_z|}\right)^2$
            if $q_z > q_B$        \\ estimated error rate for $B$ would decrease if $A$ labeled exs in $U_z$
                let $L_z$ be examples in $U_z$ as labeled by $H_A$
                $L_B = L_B \cup L_z$
                $w_B = w_B + w_z$
    \\——— Choose data for $B$ to label for $A$ ———
    For each equivalent class $z'$ defined by $H_B$
    let $U_{z'}$ be the examples from $U$ that map to $z'$
        Use $L$ for 10-fold cross validation to compute the 95%-confidence interval $[\ell_{z'}, h_{z'}]$ for $z'$
        if $h_{z'} > \ell_A$        \\First test passed for labeling examples in $U_{z'}$ for $A$
            $w_{z'} = (1 - \ell_{z'})|U_{z'}|$        \\conservative estimate for # exs in $U_{z'}$ mislabeled by $H_B$
            $q_{z'} = |L \cup L_A \cup U_{z'}| \left(1 - \frac{2(w_A + w_{z'})}{|L \cup L_A \cup U_{z'}|}\right)^2$
            if $q_{z'} > q_A$        \\ estimated error rate for $A$ would decrease if $B$ labeled exs in $U_{z'}$
                let $L_{z'}$ be examples in $U_{z'}$ as labeled by $H_B$
                $L_A = L_A \cup L_{z'}$
                $w_A = w_A + w_{z'}$

\\ ———— Final Hypothesis ————
Predict according to Combine$(H_A, H_B)$
```

Table 3. Test data characteristics and a summary of our results. For each data set we show the number of test points that are misclassified by ID3 (DT), HOODG (DG), and our co-training algorithm (COM) for the one of the three runs that had the median overall performance. The average improvement is the improvement in the error rate over the best of DT, DG, and COM at round 0.

Data	Number of		Errors After Round #						Avg.	Run		
Set	Attributes	Alg.	0	1	2	3	4	5	Improv.	#	$\lvert L\rvert$	$\lvert U\rvert$
		DT	12	12	6					1	214	298
threeof9	9	DG	51	6	6				40.3%	2	253	259
(run 1 shown)		COM	12	5	6					3	282	230
		DT	64	49	47	45				1	100	152
breast-cancer	9	DG	64	66	56	56			17.0%	2	100	152
(run 2 shown)		COM	64	56	51	51				3	100	152
		DT	29	29						1	18	142
corral	6	DG	30	31					0.0%	2	18	142
(run 2 shown)		COM	29	29						3	18	142
		DT	168	151	145	147	147	145		1	199	867
flare	10	DG	153	147	147	147	143	143	6.1%	2	238	828
(run 3 shown)		COM	153	147	147	146	143	143		3	283	783
		DT	136	108	108	103				1	126	432
monk2	6	DG	80	80	53	53			27.4%	2	144	432
(run 1 shown)		COM	77	83	59	59				3	145	432
		DT	18	18	8					1	46	82
mux6	6	DG	8	8	8				0.0%	2	48	80
(run 2 shown)		COM	18	8	8					3	52	76
		DT	37	37	29					1	27	408
vote	16	DG	87	23	23				24.7%	2	28	407
(run 3 shown)		COM	37	31	22					3	29	406
		DT	5	2						1	384	589
xd6	9	DG	2	0					100.0%	2	384	589
(run 1 shown)		COM	2	0						3	384	589

based on the information gain of the attributes. An attribute with the lowest average entropy (highest information gain) is made the root of the current subtree. The rest of the tree is built recursively.

HOODG (Kohavi, 1994) stands for hill-climbing oblivious decision graph. In MLC++, HOODG is an inducer for building oblivious decision graphs bottom-up. Unlike ID3, HOODG does not have replication (duplication of subtrees in disjunctive concepts) and fragmentation (partitioning of data into fragments) problems.

4.2 Experimental Results

We now present our experimental results. The eight data sets we used were from the *UCI Machine Learning Database* (Merz & Murphy, 1998). For all eight data sets we choose to reduce the amount of labeled data provided, using the rest as unlabeled data. We picked the size of the labeled data set L so that the supervised learning algorithms had mediocre performance since this is the setting for which our co-training algorithm is designed. Namely, it is intended for a setting where there is not enough labeled data to make highly accurate predictions but for which there is enough labeled data for the bootstrapping process of our co-training procedure.

To increase the size of U, the provided labeled data that was not put in L is placed (without labels) along with the test data to form U. For each data set, we performed three independent runs, each using a different random selection of labeled data to use for L. Table 3 shows some relevant characteristics of the data sets we used for our empirical tests as well as a summary of our results.

Figure 1 shows the results from one of our runs using the Flare data in graphical form. For this data set HOODG performed better than ID3 when using just the initial labeled data (i.e. round 0). Our co-training procedure helped both algorithms to improve their performance.

Figure 2 shows the results from one of our runs using the breast cancer data set. In this data set ID3 had the better performance. Again (as we generally see), both hypotheses were improved by the co-training.

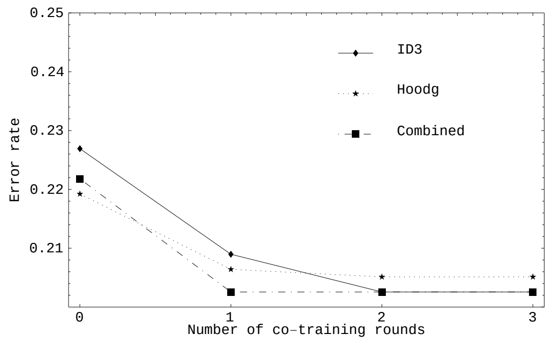

Figure 1. Results for the FLARE data set (run 2).

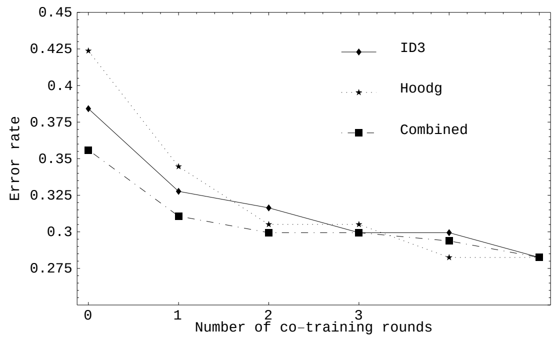

Figure 2. Results for the BREAST-CANCER data set (run 3).

While the eight different data sets (with three independent runs used for each) showed different characteristics in terms of how ID3 and HOODG interact, in 17 of the 24 runs, our co-training method reduced the error rate from both that of ID3 and HOODG. In 6 of the 24 runs, we obtained the result of the better of ID3 and HOODG. In only 1 run did the performance go down by a small amount (in that case the error went from 8/261 to 9/261). In all three runs on xd6, our co-training procedure was able to reduce the error rate to 0. The summary of the reduction of error rates we obtained with co-training are shown in Table 4. The average number of co-training rounds (where round 0 is included as a round) was 3.33.

5. Conclusions and Future Work

In this paper, we presented a new co-training strategy for learning with both labeled data and unlabeled data for settings in which there are not two redundant views of the data. Our empirical results demonstrate that two standard supervised learning algorithms can be used to successfully label data for each other.

There are many interesting directions for future research. Currently we are using a standard 95%-confidence interval for a binomial parameter. In our tests we also tried other (e.g. 99%) confidence intervals. Too high of a confidence interval does not allow data to be labeled, whereas using too low of a confidence label allows too much data to be labeled. We also tried using the confidence-rated boosting procedure from Schapire and Singer (1998). However, in general, A and B are trained on different data sets (of different sizes). This technique did not work well since the confidence rating is only relative to hypotheses obtained from training on the same data. Exploring ways to improve the estimation of the confidence interval is an important direction for future work.

Table 4. Summary of Results.

Algorithm	Avg. Error Rate
ID3	16.4%
HOODG	18.9%
Better of ID3 and HOODG	14.1%
Our Algorithm (COM)	11.9%

One variation of our co-training procedure that we studied is as follows. In each iteration, one could have each algorithm label only the examples from an equivalence class with the highest confidence level. However, along with significantly increasing the number of co-training rounds, the overall error rates were not as good using this approach. We also considered increasing the number of equivalence classes for HOODG (which is currently two) by breaking the decision graph at one level higher (i.e. uses as the leaves the internal nodes which are direct ancestors of the 0 and 1 leaf). However, this did not improve our performance.

Clearly, there are many additional variations to our specific co-training technique that would be interesting to explore and we believe that further improvements are possible. We also plan to perform additional empirical studies using real data from several application areas in which there are not two redundant set of features. In addition, we plan on running test in which we use different supervised learning algorithms (besides ID3 and HOODG). Another interesting research direction is to explore how our techniques could be used in active learning to help decide which examples would be most valuable to be labeled.

Along with performing more empirical studies, we hope to develop a theory about our co-training procedure so we can better understand when it will be appropriate and how to improve it.

Acknowledgments

The authors gratefully acknowledge the support of the National Science Foundation under grant CCR-9734940 . We also thank Dan Dooly, Dan Blandford and Bob Amar for useful discussions.

References

Angluin, D. & Laird, P. (1988). Learning from noisy examples. *Machine Learning*, *2*,343–370.

Blum, A. & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the Eleventh Annual Conference on Computational Learning Theory* (pp. 92–100).

Campbell, C., Cristianini, N., & Smola, A. (2000). Query learning with large margin classifiers. Unpublished manuscript, Department of Engineering Mathematics, University of Bristol, Bristol BS8 1TR, United Kingdom.

Collins, M. & Singer, Y. (1999). Unsupervised models for named entity classification. *Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.*

Craven, M., DiPasquo, D., Freitag, D.,McCallum, A., Mitchell, T., Nigam, K., & Slattery, S. (1998). Learning to extract symbolic knowledge from the world wide web. *Proceedings of Fifteenth National Conference on Artificial Intelligence* (pp. 509–516).

Dagan, I., & Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 150–157). San Francisco: Morgan Kaufmann.

De Comitè, F., Denis, F., Gilleron R., & Letouzey, F. (1999). Positive and unlabeled examples help learning. *Proceedings of the Tenth International Conference on Algorithmic Learning Theory* (pp. 219–230).

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, *39*,1–38.

Kohavi, R. (1994). Bottom-up induction of oblivious, read-once decision graphs. *Proceedings of the European Conference on Machine Learning.*

Hofmann, T. (1999). Text categorization with labeled and unlabeled data: A generative model approach. *Working Notes for NIPS 99 Workshop on Using Unlabeled Data for Supervised Learning.*

Jaakkola, T. & Meila, M., & Jebara, T. (1999). Maximum entropy discrimination. Technical Report MIT AITR-1668. MIT, AI Lab, Cambridge, MA.

Kearns, M. (1993). Efficient noise-tolerant learning from statistical queries. *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing* (pp. 392–401). New York: ACM Press.

Larsen, R. J. & Max, M. L. (1986). *An Introduction to Mathematical Statistics and Its Applications.* Prentice Hall.

Lewis, D. (1995). Evaluating and optimizing autonomous text classification systems. *Proceedings of the Eighteenth Annual International ACM Special Interest Group on Information Retrieval* (pp. 246–254). New York: ACM Press.

Liere, R. & Tadepalli, P. (1997). Active learning with committees for text categorization. *Proceedings of the Fourteenth National Conference on Artificial Intelligence* (pp. 591–596).

Merz, C. J. & Murphy, P. M. (1998). UCI Repository of Machine Learning Databases.

Nigam, K.,McCallum, A.K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*,103–134.

Quinlan, R. (1986). Induction of decision trees. *Machine Learning*, *1*,81–106.

Riloff E. & Jones, R. (1999). Learning dictionaries for information extraction by multi-level bootstrapping. *Proceedings of the Sixteenth National Conference on Artificial Intelligence* (pp. 474–479).

Schapire, R. E. & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, *37*,297–336.

Schuurmans, D. (1997). A new metric-based approach to model selection. *Proceedings of the Fourteenth National Conference on Artificial Intelligence* (pp. 552–558).

Shawe-Taylor, J. (1999). The use of the margin in transduction. *Working Notes for NIPS 99 Workshop on Using Unlabeled Data for Supervised Learning.*

Wu, Y. & Huang, T. S. (1999). Using unlabeled data in supervised learning by discriminant-EM algorithm. *Working Notes for NIPS 99 Workshop on Using Unlabeled Data for Supervised Learning.*

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivalling supervised methods. *Proceedings of the Thirty-third Annual Meeting of the Association for Computational Linguistics* (pp. 189–196).

Zhang, T. (1999). Some asymptotic results concerning the value of unlabeled data. *Working Notes for NIPS 99 Workshop on Using Unlabeled Data for Supervised Learning.*