Democratic Co-Learning

Yan Zhou
School of Computer and Information Sciences
University of South Alabama
Mobile, AL 36688
zhou@cis.usouthal.edu

Sally Goldman
Department of Computer Science and Engineering
Washington University
St. Louis, MO 63130-4899
sg@cse.wustl.edu

Abstract

For many machine learning applications it is important to develop algorithms that use both labeled and unlabeled data. We present democratic co-learning *in which multiple algorithms instead of multiple views enable learners to label data for each other. Our technique leverages off the fact that different learning algorithms have different inductive biases and that better predictions can be made by the voted majority. We also present* democratic priority sampling*, a new example selection method for active learning.*

1. Introduction

In many practical learning scenarios there is only a small amount of labeled data (which is often costly to obtain) along with a large pool of unlabeled data. One of many example applications is content-based image retrieval in which a user (via relevance feedback) labels a small number of images as desirable or undesirable. However, there is an extremely large pool of unlabeled images available. The goal of the content-based image retrieval system is to determine which images the user finds desirable. We use *semi-supervised learning* to refer to settings in which unlabeled data is used to augment labeled data when the size of the labeled data is insufficient.

In a *single-view* semi-supervised method the learner receives a single set of attributes to use for learning. In a *multi-view approach* (such as co-training [1]), the learner receives two or more independent and redundant sets of attributes where each view individually is adequate for learning. While there are applications with two such views, there are also many settings in which there are not. Nigam and Ghani [11] showed that co-training has strong dependence on its assumption of independent and redundant feature split.

The question we address in this paper is how unlabeled data can be used to improve the accuracy of supervised learning algorithms in situations when:

- Only a small amount of labeled data is available,
- there is a large pool of unlabeled data, and
- there are not two independent and redundant sets of attributes

Our work replaces the need for two attribute sets by leveraging from the fact that different learning algorithms have different inductive biases even when seeing the same data.

Our work is motivated, in part, by the empirical success of ensemble methods (e.g. boosting [8] or bagging [9]) in which individual classifiers are trained from different training sets using re-sampling techniques on the labeled data. There are two important questions we must address:

1. How can one create the set of hypotheses to combine to obtain better accuracy given that there is not enough labeled data to apply re-sampling techniques?
2. How can one make use of the large unlabeled pool of data?

In our work, we use an ensemble-style approach but rather than creating the classifiers with a single algorithm run on different subsets of the labeled data (which is not an option because of the limited amount of labeled data), we instead run different algorithms using the same set of data. Also ensemble methods do not use unlabeled data as an additional source of knowledge but rather are designed when

there is a sufficient source of labeled data but only weak learning algorithms. Our early work [7] demonstrates that two different algorithms can successfully label data from the unlabeled pool for each other. More recently, such an approach has been successfully applied to content-based image retrieval [19].

We present *democratic co-learning*, a new single-view semi-supervised technique, that can be used for applications without two independent and redundant feature sets and which is applicable with a small pool of labeled data. In democratic co-learning, a set of different learning algorithms are employed to train a set of classifiers separately on the labeled data set. The output concepts are combined using weighted voting to predict labels for an unlabeled examples. The newly labeled examples are added to the training set of the classifiers that predict differently than the majority. The process is repeated until no more data can be added to the training set of the classifiers. We also present *democratic priority sampling* to select examples for which to request labels for active learning. Finally, we obtain *active democratic co-learning* which uses democratic priority sampling to select examples to be actively labeled and uses democratic co-learning to label additional examples.

2. Related Work

Like ensemble methods (e.g. boosting [8] or bagging [9]), democratic co-learning integrates a group of learners to boost the overall accuracy and exploits differences in the bias between methods or methods that allow locally different models. However, there are fundamental differences and motivations. An ensemble method improves itself by creating random subsets or purposely biased distributions from the training data, which is inapplicable when the amount of training data is small.

In general the semi-supervised learning problem has been studied in two settings: multiple-view and single view. In a *single-view* semi-supervised method the learner receives a single set of attributes to use for learning. In a *multi-view approach* (such as the co-training procedure of Blum and Mitchell [1]), the learner receives two or more independent and redundant sets of attributes where each view individually is adequate for learning. Democratic co-learning is a new single-view approach.

The Expectation-Maximization *(EM)* [6] can be viewed as a single-view semi-supervised learning algorithm by treating the unlabeled examples as having a hidden variable (the label). Used in this way, EM begins with an initial classifier trained on the labeled examples. It then repeatedly uses the current classifier to temporarily label the unlabeled examples and then trains a new classifier on all labeled examples (the original and the newly labeled) until it converges. While the EM algorithm works well when the assumed model of data holds, violations of these assumptions often result in poor performance [10]. Democratic co-learning is different from other single-view algorithms such as EM [6] in that like the statistical co-learning algorithm introduced in our early work [7], it uses multiple learning algorithms to serve a similar role that multiple views provide in co-training.

Blum and Mitchell [1] introduced the multi-view semi-supervised learning approach. They make the strong assumption that the instance space can be represented using two different views (i.e. two independent and redundant sets of attributes) and that either view by itself is sufficient for perfect classification if there were enough labeled data. They presented a *co-training* algorithm for this situation and gave both empirical and theoretical results evaluating it. 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. Nigam and Ghani [11] have shown that co-training has strong dependence on its assumption of independent and redundant feature split. In this paper, we present a new single-view technique, *democratic co-learning* that is applicable to settings that violate the assumption of independent and redundant feature sets. Our technique leverages off the fact that different learning algorithms have different inductive biases and that better predictions can be made by the voted majority.

Co-EM [11] integrates co-learning and EM by using the hypothesis learned in one view to probabilistically label the examples in the other view. The primary difference between co-EM and co-training is that like EM, co-EM assigns a temporary label to each unlabeled example from scratch at each iteration whereas co-training selects a subset of the unlabeled examples to permanently label. In both cases, the hypothesis obtained from one view is used to perform labeling for the other view.

Two-view EM (*2v-EM*) [12] aims to demonstrate that the strength from co-training and co-EM does not come merely from combining classifiers learned from different views. 2v-EM performs EM on each view in isolation and then combines the prediction of the hypotheses learned in each view. Using text-categorization benchmarks they showed that when the requirement of two independent and redundant views is severely violated 2v-EM can outperform co-training and co-EM.

While democratic co-learning has similarities with statistical co-learning from our earlier work [7], there are major differences. First, statistical co-learning uses two learning algorithms and requires them each to output a hypothesis that partitions the domain into equivalence classes. For example, the decision tree output by C4.5 defines one equivalence class per leaf. This assumption limits the applicability of that approach. Also, we used statistical tests to de-

	single-view single learner	multi-view single learner	single-view multiple learners
Non-active	EM	Co-Training Co-EM 2v-EM	Statistical Co-Learning **Democratic Co-Learning**
Active	QBC (+EM) Uncertainty Sampling (+EM)	Co-Testing Co-Test(co-EM)	**Active Democratic Co-Learning**

Table 1. A framework for classifying semi-supervised algorithms.

cide when one algorithm should label data for the other. Yet, the amount of labeled data available was insufficient for applying those tests. Democratic co-learning resolves both of these problems by using an ensemble-like method to reduce the need for statistical tests and enable it to be applied to any three or more standard supervised learning algorithms.

Some useful insights for our work come from meta-learning. In theory, there is no single learning algorithm that will be superior on all problems [2]. It has also been shown that classifiers with uncorrelated errors may reduce the error rate when using a combined model [5]. Chan and Stolfo [3] considered learning in a distributed setting in which the labeled data is distributed over many locations and thus each learning algorithm only sees a subset of the labeled data. While the setting for their research is quite different than ours, their research showed that since different learning algorithms use different representations for their hypotheses and have different inductive biases, the underlying strategies embodied by different learning algorithms may complement each other by effectively reducing the space of incorrect classifications of a learned concept [3]. In their multi-algorithm meta-learning strategy [4], Chan et al. provided only a fraction of the labeled data to each base classifier yet the resulting combined classifier obtained a better overall accuracy than a classifier trained from all the available data. One key difference from our work is that they assume each learner only sees a small amount of labeled data because it is distributed. As in their work, we expect different algorithms to infer different patterns in the data. Another difference with our work is that we use the classifiers not only to boost the performance but also to label data in U to increase the pool of labeled data for other learning algorithms that did not infer the same patterns.

We briefly review work on active learning. Uncertainty sampling [13, 14] repeatedly selects an unlabeled example with the most "uncertain" membership and asks the oracle to provide the correct label. The learning algorithm then rebuilds its hypothesis based on the new training set. Query-by-committee (QBC) [13, 8] measures the degree to which a group of classifiers disagree rather than using a single classifier to measure the certainty of its classification. In QBC, committee members can be generated on different subsets of the training data, or randomly chosen according to the posterior distribution of possible models given the training data. Instead of basing priorities on the number of disagreements, we consider a variant [15] of QBC where the priority of example x is computed using the entropy of the classifications voted by each member where $Entropy(x) = -\sum_{i=1}^{r} \frac{V(i)}{k} \log_2 \frac{V(i)}{k}$ for k the number of committee members, r the total number of labels, and $V(i)$ the number of votes for label i. Examples with the highest entropy are selected for labeling.

Co-testing [12] is an active multi-view learning that repeatedly trains one hypothesis for each view and selects as a query an unlabeled example where the two hypotheses predict differently (a *contention point*). The contention points on which the combined prediction of the two classifiers is least confident is selected. *Co-Test (Co-Em)* [16] combines co-testing and co-EM to get an active multi-view semi-supervised learning algorithm. Their experiments show that co-Test (co-EM) outperforms other non-active multi-view algorithms without using more labeled data and is better able to overcome violations in the assumptions of two independent and redundant views.

Table 1 classifies semi-supervised techniques based on whether they use a single-view or multi-view approach and on whether active learning is used. Our new contributions are shown in bold.

3. Democratic Co-Learning

We now present democratic co-learning. Let L be the set of labeled data, U the set of unlabeled data, and $A_1, \ldots, A_n$ (for $n \geq 3$) the provided supervised learning algorithms[1]. Democratic co-learning begins by training all n learners on the original labeled data set L. For every example x in the unlabeled data set U, each learner predicts a label $c_i \in \mathcal{C} = \{c_1, c_2, \ldots, c_r\}$ for x. Let c_k be the majority prediction. In Section 3.1, we introduce several labeling criteria that must be satisfied before example x will be labeled

1 While we describe democratic co-learning for any number of supervised learning algorithms in our empirical work we only consider $n = 3$.

with c_k for the learners that did not predict c_k for x. All n learners are then re-trained using the updated training data and this process is repeated until no more data is selected for labeling. The final hypothesis makes predictions using a variant of a weighted majority vote among the n learners (see Section 3.2). The detailed democratic co-learning procedure is shown in Figure 1.

L is the labeled data, U is the unlabeled data
$A_1, \ldots, A_n$ are the n different learning algorithms
For $i = 1, \ldots, n$
  $L_i = L$ /⋆ labeled data for A_i ⋆/
  $e_i = 0$ /⋆ estimate for # mislabeled exs in L_i ⋆/
Repeat until none of $L_1, \ldots, L_n$ change
  For $i = 1, \ldots, n$
    Run learner A_i with data L_i to compute hyp H_i
  For each unlabeled example $x \in U$
    For possible labels $j = 1, \ldots, r$
      $c_j = |\{H_i \mid H_i(x) = j\}|$
    $k = \arg\max_j \{c_j\}$
/⋆— Choose which exs to propose for labeling —⋆/
  For $i = 1, \ldots, n$
    Use L to compute 95%-conf. int. $[\ell_i, h_i]$ for H_i
    $w_i = (\ell_i + h_i)/2$
    For $i = 1, \ldots, n$
      $L'_i = \emptyset$ /⋆ data proposed for adding to L_i ⋆/
    If $\sum_{H_j(x)=c_k} w_j > \max_{c'_k \neq c_k} \sum_{H_j(x)=c'_k} w_j$
      $L'_i = L'_i \cup \{(x, c_k)\}$, $\forall i$ such that $H_i(x) \neq c_k$
/⋆– Estimate if adding L'_i to L_i improves accuracy–⋆/
  For $i = 1, \ldots, n$
    Use L_i to compute 95%-conf. int. $[\ell_i, h_i]$ for H_i
    $q_i = |L_i| \left(1\text{–}2\left(\frac{e_i}{|L_i|}\right)\right)^2$ /⋆ est. of error rate ⋆/
    $e'_i = \left(1\text{–}\frac{\sum_{i=1}^{d} \ell_i}{d}\right) |L'_i|$ /⋆ est. of new error rate ⋆/
    $q'_i = |L_i \cup L'_i| \left(1\text{–}\frac{2\,(e_i + e'_i)}{|L_i \cup L'_i|}\right)^2$ /⋆ if L'_i added ⋆/
    If $q'_i > q_i$
      $L_i = L_i \cup L'_i$
      $e_i = e_i + e'_i$
  Return *Combine*$(H_1, H_2, ..., H_n)$

Figure 1. Democratic co-learning.

3.1. Labeling Criteria

No unlabeled example is labeled by one learner for another unless a majority of the learners agree on the label. In addition to this majority vote requirement, we also require that the sum of the mean confidence values of the learners in the majority group is greater than the sum of the mean confidence values of the learners in the minority groups where the *mean confidence* of a learner is $(\ell + h)/2$ for ℓ and h defined by the 95%-confidence interval $[\ell, h]$. We have performed experiments with 90% and 99% confidence intervals and the results were very similar. Using a vote weighted by a measure of confidence eliminates the possibility that a majority of learners make the same wrong predictions each with very low confidence. For example, suppose there are three co-learners in a binary classification problem. One learner predicts "positive" for unlabeled example x with 99% confidence and the other two predict x is "negative" each with a confidence of 30%. In this case, we would not want to let the two learners that predict x is negative, label x for the learner predicting x is positive.

In order to balance the benefits of adding more labeled examples to the training data with the increase in noise rate that may occur in the labels, we use the same tests as those in our earlier work to estimate if the increase in the labeled data is sufficient to compensate for the increase in the number of mislabeled examples. The details can be seen in Figure 1.

3.2. Combining

The easiest way to create the final hypotheses is using standard majority vote among the possible class labels. In order to combine better, in addition to the number of votes for each label, we also consider each individual classifier's confidence value (as measured by the mean of the 95%-confidence interval[2]) in its prediction. We partition classifiers into r groups, one for each possible label. We use an *m-estimator* to adjust the average of the mean confidence value of each group such that the average mean confidence value of a group is discounted more if it has smaller size. Let n be the size of a group. Based on some preliminary experiments not reported here, we use a Laplace correction of $s = \frac{n+0.5}{n+1}$ to avoid zero frequency of votes and bias towards a voting power of 0.5 when the group size is too small. The group of classifiers with the highest discounted confidence value is used to predict for the example. When the confidence value of classifiers within a group has a large variance, the adjustment made above may not be effective. Hence, we ignore any classifier whose confidence value is less than 50%. See Figure 2.

4. Democratic Priority Sampling

In this section, we present *democratic priority sampling*. As in democratic co-learning we begin by using the labeled

2 Again, empirically we found little difference when using either a 90% or 99%-confidence interval.

Combine$(H_1, H_2, ..., H_n)$

For $i = 1, \ldots, n$

Use L to compute 95%-conf. int. $[\ell_i, h_i]$ for H_i

$w_i = (\ell_i + h_i)/2$

For each example x in the instance space

For $i = 1, \ldots, n$

If $H_i(x)$ predicts c_j and $w_i > 0.5$

Allocate H_i to group G_j

For $j = 1, \ldots, r$

/$\star$ compute group average mean confidence $\star$/

$$\overline{C}_{G_j} = \frac{|G_j|+0.5}{|G_j|+1} \times \frac{\sum_{H_i \in G_j} {}^{w_i}}{|G_j|}$$

H predicts with G_k for $k = \arg\max_j(\overline{C}_{G_j})$

Return H

Figure 2. Combine procedure.

data L to train the k different learners to obtain the k classifiers $H_1, \ldots, H_k$. One possible way to then select the example to actively label would be to use the vote entropy as in QBC. However, we also want to incorporate the confidence of each individual classifier in the priority estimate. Hence we define a *confidence-weighted vote entropy* to incorporate the confidence of each individual classifier in the priority estimate by computing the vote entropy weighted by the mean confidence of the classifiers. We did test using an unweighted majority but obtain better results using a weighted majority vote.

More formally, let r be the number of different labels and let $G_i(x)$ contain the set of classifiers among $H_1, \ldots, H_k$ that predict a label of i for x. We define the priority of unlabeled example x as

$$Priority(x) = -\sum_{i=1}^{r} \frac{W_i(x)}{W} \log \frac{W_i(x)}{W}$$

where $W_i(x) = \sum_{H_j \in G_i(x)} w_j$ for w_j the mean of the 95%-confidence interval of H_j and $W = \sum_{j=1}^{k} w_j$. The example with the highest priority label is given to an expert for labeling. Then the hypotheses are recomputed using the larger pool of labeled data and the process is repeated.

While there are many similarities between democratic priority sampling and QBC, there are two key differences. First, the committee members are obtained by using different learning algorithms versus the same learning algorithm trained on different data. Secondly, we use a weighted variant of vote entropy to incorporate the confidence estimates into the priorities.

5. Empirical Results

In this section we present our empirical results. As the three base learning algorithms we use: naive Bayes (NB), C4.5 [17] and 3-nearest neighbor (3-NN) [18]. In all of our experiments, we compute the reported accuracy using a test set that is roughly the same size as the unlabeled pool.

5.1. Non-active Co-learning

We present results for the non-active setting. In the left plot in Figure 3, we compare democratic co-learning with naive Bayes, C4.5, 3-NN, and the results of using combining alone on the DNA data set. Democratic co-learning outperforms the three individual algorithms and the gain was not achieved by simply combining the prediction made by the three learners. In the right plot in Figure 3, we compare democratic co-learning with naive Bayes, C4.5 and 3-NN when each is combined with EM to use the unlabeled data. In each of these plots $|L|$ varies between 35 and 100 with an independent run of each algorithm performed for each integer in this range. The purpose of these experiments is to evaluate how the performance of these methods is affected by varying the size of the pool of labeled data. Across all values for $|L|$ we tested, democratic co-learning outperforms the three individual algorithms when they are combined with EM to make use of the labeled data. Notice that EM may have negative impact on poor classifiers trained over insufficient labeled data.

We now consider a single value of $|L|$ over a variety of data sets. We show the performance of each base algorithms, as well as the performance when we just use the combining method of democratic co-learning to demonstrate that we are making use of the unlabeled data as opposed to having our gains come from the ensemble of the three base algorithms. We also compared our work to other semi-supervised learning algorithms. For statistical co-learning we use naive Bayes and C4.5 since they generally perform better than 3-NN. To create a hypothesis from naive Bayes that partitions the input domain as required by statistical co-learning, we take all of the data in U and label it according to the naive Bayes hypothesis and then use C4.5 to create the equivalence classes (one per leaf). We use eight of the UCI[3] benchmark data sets. For all data sets $|L| = 40$ except for the adult data set where $|L| = 60$. Table 2 shows other statistics about the data sets. We created 20 different data sets by randomly partitioning the data into L, U, and the test data. In addition, we picked random partitions in which democratic co-learning labeled at least one example in U.

3 http://www.ics.uci.edu/∼mlearn/MLRepository.html

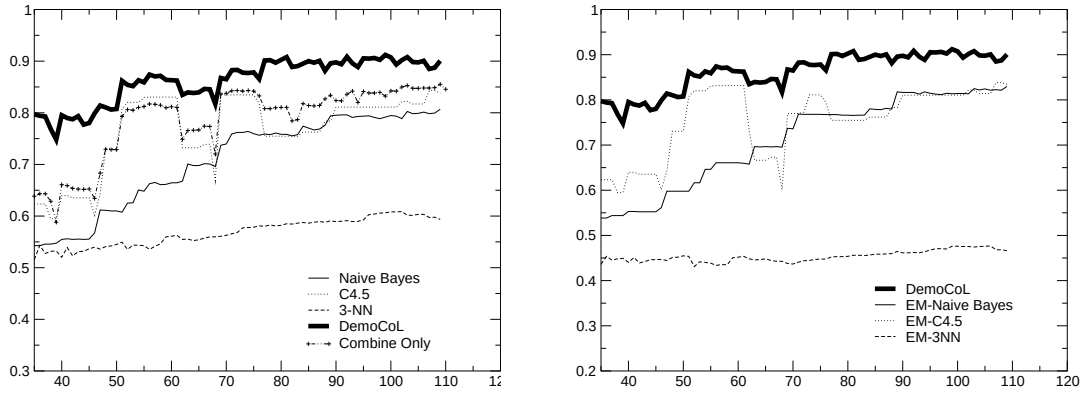

Figure 3. Results on DNA data. The x-axis is $|L|$ and the y-axis is the accuracy.

data set	# atts	exs from U labeled for NB	C4.5	3-NN	$\|L\|$	$\|U\|$	avg. # rounds
flare	10	108	151	80	40	515	2.7
monk2	6	40	84	40	40	193	2.3
vote	16	66	40	40	40	200	2.2
DNA	180	367	289	432	40	1588	2.8
cancer	9	59	40	45	40	124	2.1
adult	14	413	130	353	60	1691	2.6
3-of-9	9	40	91	40	40	234	2.3
xd6	9	86	115	40	40	463	2.7

Table 2. A summary of amount of data labeled by democratic co-learning.

A key contribution of our work is a semi-supervised learning technique that can be applied when there are not such independent and redundant set of attributes. Since the work on two-view approaches generally only reports results on data sets that naturally have two appropriate feature sets, comparing our work to those approaches requires that we re-implement their work. We have selected to do this for the Blum and Mitchell co-training procedure [1] which we refer to as *two-view co-training*. In order to create two views, we randomly partition the features into two sets and then treat these as our two views as done by Nigam and Ghani [11]. We also tested how sensitive the performance of two-view co-training was to the random choice of the partition of the features. For each of the UCI data sets we fixed the choice of which examples to place in L, U, and the test set and then randomly picked 20 different random partitions of the features into two sets. For these we found a standard deviation of anywhere from 0.03 to 0.06. Finally, we present results obtained by using EM with each of the three base algorithms. To create a measure of the best performance one could expect for the given data sets, the column labeled by "data in U labeled" shows the *best* result obtained among any of the base algorithms (naive Bayes, C4.5, and 3-NN) when all examples in U are correctly labeled and placed in L. Due to the small size of L and therefore considerable variation in performance, a paired t-test is used to determine the statistical significance of the difference made by democratic co-learning. Our results are shown in Table 3. The value in parenthesis is the value of the paired t-test values between democratic co-learning and that method. A positive value indicates that democratic co-learning performed better. Any value that is statistically significant at the 95% confidence level or higher (i.e. ≥ 2.093) is in bold. All values greater than 2.861 are also statistically significant at the 99% level and all values greater than 3.8834 are statistically significant at the 99.9% level. Due to space constraints the standard deviation is only shown for democratic co-learning.

As compared to combining alone, the performance of democratic co-learning performs better at the 95% confidence level for 6 of the 8 data sets and at the 90% confidence level for the other two data sets. So democratic co-learning

Algorithm	flare	monk2	vote	DNA
Demo. Co-Learning	.833±.013	.656±.035	.944±.012	.747±.045
Combining Only	.808 (1.997)	.641 (**4.798**)	.938 (2.042)	.652 (**7.025**)
Data in U Labeled	.832	.654	.948	.882
Statistical Co-Learning	.813 (**2.112**)	.646 (0.993)	.864 (**2.879**)	.623 (**3.566**)
Two-View Co-Training	.737 (**5.841**)	.607 (**4.854**)	.859 (**9.959**)	.768 (-1.966)
EM-NB	.740 (**4.930**)	.647 (**6.210**)	.846 (**8.285**)	.567 (**13.852**)
EM-C4.5	.822 (**2.612**)	.554 (**10.846**)	.942 (1.630)	.676 (**6.740**)
EM-3NN	.814 (**5.303**)	.655 (1.000)	.893 (**12.157**)	.588 (**8.693**)
NB	.753 (**5.714**)	.643 (**6.425**)	.861 (**7.546**)	.573 (**13.350**)
C4.5	.822 (**2.612**)	.554 (**10.846**)	.942 (1.630)	.676 (**6.740**)
3NN	.812 (**5.333**)	.637 (**2.836**)	.902 (**9.259**)	.626 (**8.584**)

Algorithm	cancer	adult	3-of-9	xd6
Demo. Co-Learning	.703±.026	.784±.021	.774±.031	.774±.021
Combining Only	.692 (**2.697**)	.771 (**4.154**)	.759 (**5.181**)	.760 (**2.667**)
Data in U Labeled	.744	.820	.956	.978
Statistical Co-Learning	.698 (0.386)	.757 (**2.781**)	.711 (**5.221**)	.731 (**3.484**)
Two-View Co-Training	.653 (**2.752**)	.787 (-0.280)	.636 (**9.931**)	.650 (**7.175**)
EM-NB	.670 (**5.351**)	.725 (**4.358**)	.608 (**14.386**)	.672 (**22.433**)
EM-C4.5	.695 (1.462)	.750 (**7.452**)	.669 (**7.808**)	.721 (**6.580**)
EM-3NN	.704 (-1.40)	.770 (**2.630**)	.725 (**4.944**)	.717 (**13.638**)
NB	.652 (**8.923**)	.732 (**3.927**)	.755 (**4.816**)	.750 (**4.680**)
C4.5	.695 (1.462)	.750 (**7.398**)	.667 (**7.881**)	.723 (**6.111**)
3NN	.695 (**3.206**)	.762 (**4.091**)	.758 (**4.116**)	.756 (**3.239**)

Table 3. Our non-active learning results.

is making use of the unlabeled data and not just benefiting from the use of an ensemble method of combining. As compared to the other 5 semi-supervised methods, democratic co-learning performs statistically significantly at the 95% level in 32 of the 40 tests we performed. (In fact in 27 of the 40 tests, our improvements are statistically significant at the 99% confidence level.) Of the 8 tests in which the difference in performance was not statistically significant democratic co-learning performed better in all but two of them.

5.2. Active Co-learning

Table 4 shows our active learning results. For uncertainty sampling we use naive Bayes where the normalized probability measure of naive Bayes is used to give an uncertainty value. For QBC we use $k = 5$ different committee members each trained with naive Bayes on a random subset (without replacement) of $|L|/k$ examples from L. The active learning is used to select 40 additional examples to have labeled.

We first show the best result obtained among the base algorithms when all data in U is properly labeled. Next we compare democratic priority sampling (with no use of the unlabeled data except in serving as a pool of data for which labels may be requested) with QBC and uncertainty sampling. For QBC and uncertainty sampling, we show the paired t-test value with respect to democratic priority sampling. Finally, we compare the following active and semi-supervised algorithms: active democratic co-learning, co-testing, and co-test(co-EM) showing the paired t-test values with respect to active democratic co-learning.

For the active approaches in which the unlabeled data is only used as a pool for the active learner, democratic priority sampling performed better in 5 of the 8 data sets than each of QBC and uncertainty sampling but only 2 of these 5 cases (for each data set) was statistically significant at the 95% level. We are currently repeating these experiments using 20 different random choices for L, U, and the test data and we believe that we will find statistically significant improvements in more cases. For the active semi-supervised algorithms, democratic co-learning performed better than each of co-testing and co-test (Co-EM) in 5 of the 8 data sets with 4 of the 5 (for each data set) being statistically significant at the 95% level.

We also ran a paired t-test between democratic priority sampling and active democratic learning. For the 3-of-9 and DNA data sets the improvement of active democratic co-learning was statistically significant at the 95% level, and

Algorithm	flare	monk2	vote	DNA
Data in U Labeled	$.832 \pm .012$	$.654 \pm .030$	$.948 \pm .021$	$.882 \pm .179$
Demo. Priority Samp.	$.819 \pm .011$	$.644 \pm .011$	$.934 \pm .028$	$.714 \pm .060$
Query By Committee	.768 (**4.654**)	.632 (1.165)	.880 (2.190)	.813 (-2.631)
Uncertainty Sampling	.815 (0.246)	.637 (0.631)	.918 (2.120)	.701 (0.703)
Active Demo. Co-Learn.	$.821 \pm .013$	$.644 \pm .011$	$.941 \pm .023$	$.849 \pm .049$
CoTesting	.824 (-0.714)	.644 (0.000)	.914 (2.581)	.740 (**5.160**)
Co-Test(co-EM)	.826 (-0.805)	.646 (-0.690)	.898 (**4.493**)	.744 (2.189)

Algorithm	cancer	adult	3-of-9	xd6
Data in U Labeled	$.744 \pm .031$	$.820 \pm .179$	$.956 \pm .051$	$.978 \pm .012$
Demo. Priority Samp.	$.696 \pm .028$	$.772 \pm .061$	$.809 \pm .046$	$.842 \pm .048$
Query By Committee	.679 (0.753)	.793 (-.708)	.773 (1.616)	.762 (**3.105**)
Uncertainty Sampling	.710 (-3.903)	.792 (-0.717)	.769 (2.175)	.776 (**2.933**)
Active Demo. Co-Learn.	$.709 \pm .032$	$.812 \pm .012$	$.849 \pm .048$	$.893 \pm .031$
CoTesting	.710 (-0.051)	.788 (**5.086**)	.734 (**4.563**)	.762 (**8.610**)
Co-Test(Co-EM)	.718 (-0.941)	.776 (**4.695**)	.602 (**7.260**)	.676 (**14.529**)

Table 4. Our active learning results.

for the vote and XD6 data sets, the improvement of active democratic co-learning was statistically significant at the 90% level. For the flare and monk2 data sets there really is not much room for improvement. Similarly, in comparing the performance between democratic co-learning and active democratic co-learning the use of active learning generally improved the performance in data sets in which the performance of democratic co-learning was not already close to that obtained when all data in U is given the proper label.

6. Concluding Remarks

We have demonstrated that democratic co-learning, a single-view multiple algorithm semi-supervised learning technique is statistically superior to many semi-supervised learning approaches when there are not two sufficiently independent and redundant set of attributes. Using data from the UCI repository, we have compared the performance of democratic co-learning to combining alone (without using the unlabeled data) and to other single-view and multi-view semi-supervised learning algorithms. Democratic co-learning performed better at the 95% confidence level in 38 of the 48 tests that we performed in the non-active learning setting. For the other 10 tests there was no significant difference in performance between democratic co-learning and the other approaches studied.

In general, co-learning works well if the estimated mean confidence reflects which learner is better and when the multiple classifiers are good in different regions enabling them to classify data for each other. Finally, there needs to be room for improvement by at least one of the supervised learning algorithm if it received more correctly labeled data. Democratic co-learning also outperformed each of the three individual algorithms when combined with EM and by picking learners that work in very different ways, we can increase the diversity needed for them to be able to label data for each other.

References

[1] Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proc. of the 11th Annual Conf. on Computational Learning Theory. (1998) 92–100.

[2] Schaffer, C.: A conservation law for generalization performance. In: Proc. of the 11th Int. Conf. on Machine Learning, San Mateo: Morgan Kaufmann (1994) 259–265.

[3] Chan Philip, K., Stolfo, S.: On the accuracy of meta-learning for scalable data mining. Journal of Intelligent Integration of Information **8(1)** (1998) 5–28.

[4] Chan Philip, K., Stolfo, S.: Scaling learning by meta-learning over disjoint and partially replicated data. In: Proc. of the 9th Florida AI Research Symposium. (1996) 151–155.

[5] Ali, K., Pazzani, M.: Error reduction through learning multiple descriptions. Machine Learning **24** (1996) 173–202

[6] Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society, Series B **39** (1977) 1–38

[7] Goldman, S., Zhou, Y.: Enhancing supervised learning with unlabeled data. In: Proc. of the 17th Int. Conf. on Machine Learning, San Francisco: Morgan Kaufmann (2000) 327–334.

[8] Freund, Y., Seung, H., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. Machine Learning **28** (1997) 133–168.

[9] Breiman, L.: Bagging predictors. Machine Learning **24(2)** (1996) 123–140.

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

[11] Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: The 9th Int. Conf. on Information and Knowledge Management. (2000) 86–93.

[12] Muslea, I., Minton, S., Knoblock, C.: Selective sampling with redundant views. In: Proc. of AAAI-2000. (2000) 621–626.

[13] Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Proc. of the ACM Workshop on Computational Learning Theory. (1992) 287–294.

[14] Lewis, D.D., Gale, A.W.: A sequential algorithm for training text classifiers. In: Proc. of the Special Interest Group on Info. Retrieval, AAAI Press and MIT Press (1994) 3–12.

[15] Dagan, I., Engelson, S.: Committee-based sampling for training probabilistic classifiers. In: Proc. of the 12th Int. Conf on Machine Learning, San Francisco: Morgan Kaufmann (1995) 150–157.

[16] Muslea, I., Minton, S., Knoblock, C.: Selective sampling + semi-supervised learning = robust multi-view learning. In: IJCAI-01 Workshop on Text Learning: Beyond Supervision. (2001)

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

[18] Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory **13** (1967) 21–27.

[19] Zhou, Z., Chen, K., Jiang, Y.: Exploiting unlabeled data in content-based image retrieval. In: Proc of the 15th European Conf. on Machine Learning. (2004)