Disagreement-based co-training

Tanha, J.; van Someren, M.; Afsarmanesh, H.

DOI
10.1109/ICTAI.2011.126

Publication date
2011

Document Version
Author accepted manuscript

Published in
2011 23rd IEEE International Conference on Tools with Artificial Intelligence: ICTAI 2011: proceedings

Citation for published version (APA):
Tanha, J., van Someren, M., & Afsarmanesh, H. (2011). Disagreement-based co-training. In T. M. Khoshgoftaar, & X. Zhu (Eds.), *2011 23rd IEEE International Conference on Tools with Artificial Intelligence: ICTAI 2011: proceedings* (pp. 803-810). IEEE. https://doi.org/10.1109/ICTAI.2011.126

Disagreement-Based Co-Training

Jafar Tanha, Maarten van Someren, and Hamideh Afsarmanesh
Informatics Institute
University of Amsterdam
Amsterdam, The Netherlands
j.tanha, M.W.vanSomeren, h.afsarmanesh@uva.nl

***Abstract*—Recently, Semi-Supervised learning algorithms such as co-training are used in many domains. In co-training, two classifiers based on different subsets of the features or on different learning algorithms are trained in parallel and unlabeled data that are classified differently by the classifiers but for which one classifier has large confidence are labeled and used as training data for the other. In this paper, a new form of co-training, called Ensemble-Co-Training, is proposed that uses an ensemble of different learning algorithms. Based on a theorem by Angluin and Laird that relates noise in the data to the error of hypotheses learned from these data, we propose a criterion for finding a subset of high-confidence predictions and error rate for a classifier in each iteration of the training process. Experiments show that the new method in almost all domains gives better results than the state-of-the-art methods.**

***Keywords*-Semi-Supervised Learning (SSL), Co-training, Self-training, Ensemble Learning, Disagreement learning.**

I. Introduction

In many practical learning domains, such as object detection, web-page categorization, or e-mail classification, there is often only a limited number of labeled data, which is often expensive or time consuming to obtain. Meanwhile, having access to a large pool of unlabeled data is readily possible in many domains. A Semi-Supervised approach tries to use both labeled and unlabeled instances as training data. The goal of Semi-Supervised Learning is to employ unlabeled instances for boosting the performance of supervised learning, which is trained with only a small amount of labeled instances.

There are several different methods for Semi-Supervised learning based on different assumptions. Expectation-Maximization (EM) [1] is one of the well-known Semi-Supervised methods. It uses a generative model, for example a mixture of Gaussians [2]. Other methods are the Transductive Support Vector Machine (TSVM) [3] and Semi-Supervised SVM method (S3VM) [4]. Recently, some new approaches have also been presented based on the graph-based models [5].

Here we consider co-training [6] as one of the widely used Semi-Supervised learning methods. Co-training involves two, preferably independent, views of data both of which are individually sufficient for training a classifier. In co-training, the “views” are subsets of the features that describe the data. Each classifier predicts labels for the unlabeled data and a degree of confidence. Unlabeled instances that are labeled with high confidence by one classifier are used as training data for the other. This is repeated until none of classifiers changes. In [7] different learning algorithms instead of feature subsets are used for co-training. This version uses statistical tests to decide which unlabeled data can be labeled with confidence.

In this paper, we propose two improvements for co-training. We call the resulting method Ensemble-Co-Training. First we consider co-training by an ensemble of N classifiers that are trained in parallel and second we derive a stop-criterion, using a theorem by Angluin and Laird [8] that describes the effect of adding uncertain data. Training an ensemble that votes on the labels for unlabeled data has the advantage that better estimates of confidence of predictions are obtained. This is useful because Semi-Supervised learning is used in settings where only a small amount of labeled data is available. The stop-criterion is useful because it does not require techniques like cross-validation or bagging, both of which are expensive in applications with a small amount of labeled data. It has two parameters, taken from PAC-learning (Probably Approximately Correct) [9], with an intuitive meaning: the probability and size of a prediction error. Our experiments on UCI datasets [10] show that Ensemble-Co-Training improves the performance of classification more than other methods.

The rest of this paper is organized as follows. Section II outlines the related work on Semi-Supervised learning. In section III we derive the Ensemble-Co-Training algorithm. In section IV we compare the results of the Ensemble-Co-Training on the UCI datasets. Finally, in section V, we present our conclusion and discussion.

II. Related Work

We distinguish three categories of Semi-Supervised learning methods in terms of the number of classifiers and views of data: learning with single-view and single classifier, learning with multiple views, and learning with single view and multiple classifiers.

Self-training, as a single-view Semi-Supervised learning method, has been used in many domains such as Natural Language Processing [11] and Image Classification [12]. In self-training, a classifier is first trained with a small amount of labeled data. The classifier is then used to predict the labels for the unlabeled instances (prediction step). In the next step, a subset S of the unlabeled instances, with their predicted labels, is selected to be added to the labeled instances (selection step). Typically, S consists of a few instances with high confidence predictions. The classifier is then re-trained on the new set of labeled instances and the training process is repeated for boosting the number of labeled instances, in order to improve the self-training performance (re-training step).

The research presented in [13], proposes a self-training Semi-Supervised support vector machine (SVM) algorithm that includes a model selection method, designed to train a classifier with small training data. Two examples show the validity of their algorithm with model selection. They apply the self-training algorithm to a data set collected from a P300-based brain computer interface (BCI) speller. In [11] a Semi-Supervised self-training approach with decision tree classifiers is used to classify sentences as subjective or objective. Instead of using the distribution in the leaves as confidence, they use the Naive Bayes trees algorithm, which builds a Naive Bayes Classifier at each leaf, and a version of decision tree learning without pruning which is known to work well with very small datasets.

As mentioned earlier, the co-training paradigm that was proposed by Blum and Mitchell [6] works well in domains that naturally have multiple views of data. Nigam and Ghani [14] analyzed the effectiveness and applicability of co-training when there are no two natural views of the data. They show that when independent and redundant views exist, co-training algorithms outperform other algorithms using unlabeled data, otherwise there is no difference. However in practice, many domains are not described by a large number of attributes that can naturally be split into two views. The result of randomly partitioned views is not always effective. They proposed a new multi-view co-training algorithm, which is called co-EM. In this algorithm they combined multi-view learning with the probabilistic EM model. The naive Bayes classifiers are used to estimate class labels. There are many application for the original co-training approach and co-EM, for example see [15].

Instead of co-training with classifiers that use different subsets of the features, Goldman and Zhou [7] proposed a method which trains two different classifiers with different learning algorithms. Their method uses time-consuming statistical tests to select unlabeled data for labeling. The rest of the co-training process in their method is similar to the standard co-training. Later, the same authors propose Democratic Co-Learning [16]. In democratic co-learning, a set of different learning algorithms is used to train a set of classifiers separately on the labeled data set in self-training manner. In this method they also use a statistical method for selecting unlabeled data in order to labeling them. Although, it does not rely on the existence of two views, their method still uses a time-consuming method for selecting unlabeled data.

Zhou and Li [17] propose the Tri-Training method, a form of co-training that uses a base learning algorithm to construct three classifiers from different sub-samples. The classifiers are first trained on data sets that are generated from the initial labeled data via bootstrap sampling [18]. Two classifiers then predict labels for the third classifier and vice versa. Predictions are made via majority voting by the three final classifiers. Tri-Training does not need multiple views of data nor statistical testing. Tri-Training can be done with more than three classifiers, which gives a method called Co-Forest [19]. A problem with this approach in the context of Semi-Supervised learning is that there is only a small amount of labeled data and therefore it is not possible to select subsamples that vary enough and are of a sufficient size. Furthermore, Ensemble learning methods need different classifiers [**?**] to be effective, but, as mentioned earlier, when there is only a small amount of labeled data then all initial training data that are generated by bagging methods will be roughly the same. Therefore, in this case the ensemble approach will not be effective.

In [20] an ensemble of decision trees is used for image classification. The basic idea of this algorithm is to start with a limited number of labeled images, and gradually propagate the labels to the unlabeled images. In each iteration a set of newly-labeled examples is added to the training set to improve the decision trees. The improved decision trees are used to propagate labels to more images, and the process is repeated when it reaches the stopping conditions. This algorithm directly uses the prediction of weak decision tree classifiers, which is somehow problematic. Adding mislabeled instances may lead to worse results.

Our approach in this paper is to use multiple base classifiers with different learning algorithms instead of using same base learner on the different subsamples of original labeled data. In this approach, the algorithm does not require independent and redundant attributes, but instead it employs N different hypotheses as in ensemble methods. This approach also tends to improve the accuracy of finding a subset of high-confidence predictions using different classifiers which is more challenging in self-training. We use a notion from PAC-learning for controlling the error rate, selecting a subset of high-confidence predictions by ensemble, and deciding when to stop the training process.

The method is related to "Disagreement Boosting" [21], which performs a form of boosting in which examples are re-weighted not by their predictability but by disagreement between multiple classifiers. Both methods use agreement between trained classifiers. In the Boosting approach the learning algorithm constructs a hypothesis in each iteration and incorporates these in a linear combination where in our approach each learning algorithm contributes a hypothesis. These are then combined to make a prediction. Furthermore, a key of disagreement-based methods also is to generate multiple learners which are crucial in ensemble approach.

III. Learning from noisy data in Ensemble-Co-Training algorithm

Two key issues in co-training are: (1) measuring the confidence in labels that are predicted for the unlabeled data, and (2) a criterion for stopping the training process [22]. Co-training aims at adding a subset of the high-confidence predictions, called newly-labeled examples. At some point labels will be noisy and cause the result of learning to become worse. This is a form of "overfitting". Problems (1) and (2) can be solved in an empirical way, by using a holdout set of labeled data to assess the effect of adding newly-labeled data. However, since Semi-Supervised learning is used for learning tasks where labeled data is scarce this is not a good solution. Instead, we propose an analytic solution for solving this problem. This can be summarized as follows. We use a theorem from PAC-learning that relates the number of training data to the probability that a consistent hypothesis has an error larger than some threshold for a setting with training data and with a certain error in the labels. We use an ensemble of learners for co-training instead of two and the agreement between the predictions of labels for the unlabeled data to obtain an estimate of the labeling error rate. Using this we can estimate the effect of learning on the error of the result of adding the newly-labeled data to the training set. This is used to decide which subset of high-confidence predictions should be added to the initial labeled data in order to improve the classification performance. Finally, the training process will be stopped when the estimated error rate in the initial labeled data is expected to increase. Figure 1 shows the general overview of Ensemble-Co-Training.

In this section we review the theorem that we use and show how it can be used to define a criterion for adding newly-labeled data. The entire algorithm is presented in section III-B.

A. Criterion for error rate and number of unlabeled data

In Semi-Supervised learning there is a small amount of labeled data and a large pool of unlabeled data. Data points can be divided into two parts: the points X_l=(x_1,x_2...,x_l), for which labels Y_l={+1,-1} are provided, and the points X_u= (x_{l+1},x_{l+2},...,x_{l+u}), the labels of which are not known. We assume that labeled and unlabeled data are drawn independently from the same data distribution. Meanwhile, we consider $l \ll u$, where l and u are the number of labeled data and unlabeled data respectively, which is more suitable for Semi-Supervised setting.

For selecting examples for labeling, and for deciding when to stop, we need an estimate of the effect of this on the error. As we observed, using a hold-out set for this is not attractive because we have little labeled data. We cannot expect that the effect gives substantial improvements on the labeled data either. This motivates the need for an estimate of the effect of labeling unlabeled data and adding them to the training data. Inspired by [7] and [17], we formulate a function that estimates the true classification error of a hypothesis from the size of the training set and the probability that a data point is mislabeled. This is based on a PAC-learning theorem by Angluin and Laird [8]. This theorem is as follows.

Theorem 1. If we draw a sequence σ of m data points where each data point has a probability η of being mislabeled, and we compute the set of hypotheses that are consistent with σ then if

$$m \geq \frac{2}{\epsilon^2(1-2\eta)^2}\ln(\frac{2N}{\delta}) \tag{1}$$

holds, where ϵ is the classification error of the worst remaining candidate hypothesis on σ, η (< 0.5) is an upper bound on the noise rate in the classifications of the data, N is the number of hypotheses, and δ is a probability that expresses confidence, then for a hypothesis H_i that minimizes disagreement with σ holds that:

$$Pr[d(H_i, H^*) \geq \epsilon] \leq \delta \tag{2}$$

where $d(,)$ is the sum over the probabilities of differences between classifications of the data according to hypothesis i and the actual data.

To construct an estimator for the error of a hypothesis, ϵ, we rewrite the above inequality as follows. First, we set δ to a fixed value, which means that we assume that the probability of an error is equal for all hypotheses, and second assume that N is (approximately) constant between two iterations.

Here, we introduce c such that $c = 2\lambda\ln(\frac{2N}{\delta})$, where λ is chosen so that equation (1) holds. Substituting c in (1) gives:

$$m = \frac{c}{\epsilon^2(1-2\eta)^2} \tag{3}$$

So, reformulating (3), then gives:

$$\frac{c}{\epsilon^2} = m(1-2\eta)^2 \tag{4}$$

Based on this corollary, the error rate ϵ can be controlled. In particular the change in the (bound on the) error rate can be estimated and used to select the newly-labeled examples and to decide when to stop.

For this, we need to estimate η. This is done using a set of hypotheses that are constructed by different learning algorithms. Suppose that there are k classifiers, denoted by H^*. All classifiers in H^* except h_j, called $H_p \; and \; p = 1, ..., k$ such that $p \neq j$, predict labels for the unlabeled data based on voting methods. Then the newly-labeled data is used for h_j, called component classifier, in the next iteration of the training process. In more details, let L, U, and $L_{i,j}$ denote the labeled data, unlabeled data, and the newly-labeled instances for jth classifier in the ith iteration of training process respectively. Moreover, in the ith iteration, a component classifier h_j has an initial labeled data with size $|L|$ and a number of newly-labeled data with size $|L_{i,j}|$, determined by the ensemble H_p. Assume that the error rate of H_p on $L_{i,j}$ is $\hat{e}_{i,j}$. Then the number of instances that are mislabeled by H_p in $L_{i,j}$ is estimated as $\hat{e}_{i,j}|L_{i,j}|$, where $\hat{e}_{i,j}$ is upper bound of classification error rate of the H_p. The same computation is also done for the initial labeled data such that $\eta_L|L|$, where η_L denotes the classification noise rate of L. As mentioned above, the training set in the ith iteration of training process is $L \cup L_{i,j}$ for a component classifier h_j. In this training set the number of noisy instances are instances in L and instances in $L_{i,j}$. Therefore, the noise rate in this training set can be estimated by:

$$\eta_{i,j} = \frac{\eta_L|L| + \hat{e}_{i,j}|L_{i,j}|}{|L| + |L_{i,j}|} \tag{5}$$

As shown, the error rate in the ith iteration for a component classifier h_j is estimated by (5). Since c in (4) is a constant, for simplicity assume that $c = 1$, substituting (5) in (4), when training is in the ith iteration, gives:

$$n_{i,j} \simeq \frac{1}{\epsilon_{i,j}^2} = (|L| + |L_{i,j}|)(1 - 2\eta_{i,j})^2. \tag{6}$$

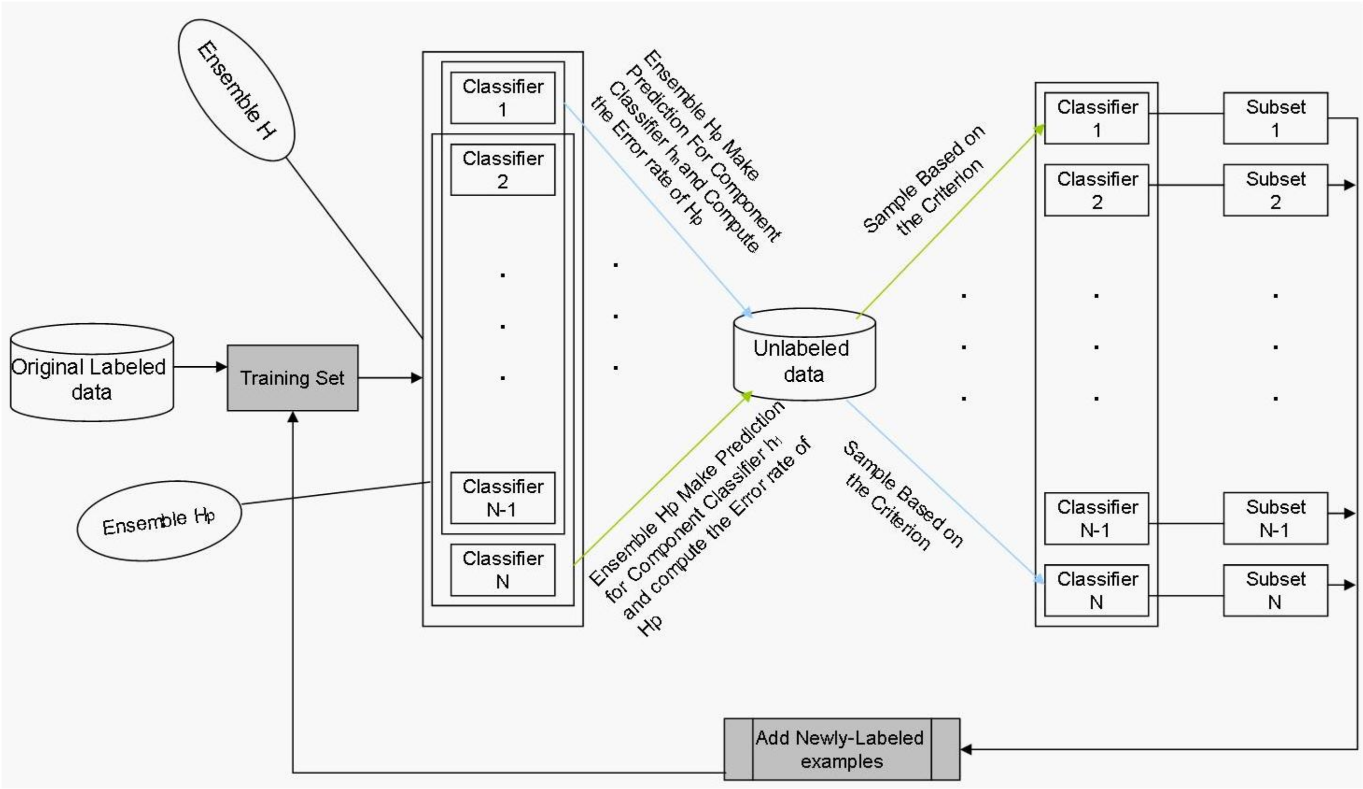
Ensemble H
Classifier 1
Classifier 2
Classifier N-1
Classifier N
Ensemble Hp Make Prediction For Component Classifier hi and Compute the Error rate of Hp
Sample Based on the Criterion
Unlabeled data
Original Labeled data
Training Set
Ensemble Hp
Ensemble Hp Make Prediction for Component Classifier hi and compute the Error rate of Hp
Sample Based on the Criterion
Classifier 1
Classifier 2
Classifier N-1
Classifier N
Subset 1
Subset 2
Subset N-1
Subset N
Add Newly-Labeled examples

Fig. 1. Block diagram of the proposed Ensemble-Co-Training

From this we derive a criterion for whether adding data reduces the error of the result of learning or not.

Theorem 2. If the following inequality satisfies in the ith and $(i-1)th$ iteration:

$$\frac{\hat{e}_{i,j}}{\hat{e}_{i-1,j}} < \frac{|L_{i-1,j}|}{|L_{i,j}|} < 1 \tag{7}$$

where $j = 1, 2, ..., k$, then in the ith and $(i-1)th$ iteration, the worst-case error rate for a component classifier h_j satisfies $\epsilon_{i,j} < \epsilon_{i-1,j}$.

Proof:

Given the inequalities in (7), then will have:

$$|L_{i,j}| > |L_{i-1,j}| \quad and \quad \hat{e}_{i,j}|L_{i,j}| < \hat{e}_{i-1,j}|L_{i-1,j}| \tag{8}$$

Thus, it can be easily shown that:

$$\begin{gathered} |L| + |L_{i,j}| > |L| + |L_{i-1,j}| \quad and\ then \\ \frac{\eta|L| + \hat{e}_{i,j}|L_{i,j}|}{|L| + |L_{i,j}|} < \frac{\eta|L| + \hat{e}_{i-1,j}|L_{i-1,j}|}{|L| + |L_{i-1,j}|} \end{gathered} \tag{9}$$

According to the (5) and (9) will have:

$$\eta_{i,j} = \frac{\eta|L| + \hat{e}_{i,j}|L_{i,j}|}{|L| + |L_{i,j}|} \quad and \quad \eta_{i-1,j} = \frac{\eta|L| + \hat{e}_{i-1,j}|L_{i-1,j}|}{|L| + |L_{i-1,j}|} \tag{10}$$

Hence, $\eta_{i,j} < \eta_{i-1,j}$ and then it can be easily written as:

$$\begin{aligned} n_{i-1,j} &= (|L| + |L_{i-1,j}|)(1 - 2\eta_{i-1,j})^2 \\ and\ n_{i,j} &= (|L| + |L_{i,j}|)(1 - 2\eta_{i,j})^2 \end{aligned} \tag{11}$$

So, according to $\eta_{i,j} < \eta_{i-1,j}$ and (9) will have; $n_{i,j} > n_{i-1,j}$, and since according to (4) $n \propto \frac{1}{\epsilon^2}$, then $\epsilon_{i,j} < \epsilon_{i-1,j}$.

Theorem 2 can be interpreted as saying that if the inequality (7) is satisfied, then the worst-case error rate of a component classifier h_j will be iteratively reduced, and the size of newly-labeled instances is iteratively increased in the training process.

As can be derived from the (7), $\hat{e}_{i,j} < \hat{e}_{i-1,j}$ and $|L_{i,j}| > |L_{i-1,j}|$ should be satisfied at the same time. However, in some cases $\hat{e}_{i,j}|L_{i,j}| < \hat{e}_{i-1,j}|L_{i-1,j}|$ may be violated because $|L_{i,j}|$ might be much larger than $|L_{i-1,j}|$. When this occurs, in order not to stop the training process, a subsample of $|L_{i,j}|$ are randomly selected such that new $|L_{i,j}|$ satisfies:

$$|L_{i,j}| < \frac{\hat{e}_{i-1,j}|L_{i-1,j}|}{\hat{e}_{i,j}}, \tag{12}$$

The condition in inequality (7) is used for stopping the training process and controlling the number of newly-labeled data as well as the error rate in Ensemble-Co-Training. In section III-B we present the Ensemble-Co-Training algorithm based on the analysis that we did in this section. This condition is based on several assumptions. The theorem holds for a ”candidate elimination” type of learning algorithm rather than a best hypothesis estimator that learns from examples that are

```
• Initialize: L,U, H
• At each iteration i:
  1. for j ∈ {1,2,...,k}
     - Find  $\hat{e}_{i,j}$ as error rate
       for component classifier  $h_j$
       based on disagreement among classifiers
     - Assign labels to the unlabeled examples
       based on agreement among ensemble $H_p$
     - Sample high-confidence examples for
       component classifier $h_j$
     - Build the component classifier $h_j$ based
       on newly-labeled and original labeled
           examples
  2. Control the error rate for each component
     classifier based on inequality (7)
     - Update ensemble H
• Generate final hypothesis
```

Fig. 2. An outline of the Ensemble-Co-Training

drawn at random. In our application the examples are selected and a single hypothesis is constructed in each iteration. At the end the multiple hypotheses are combined in an ensemble. Empirical evaluation must show if this the criterion gives useful results. An overview of our approach is presented in Figure 1.

B. The Ensemble-Co-Training Algorithm

In Ensemble-Co-Training each component classifier h_j is first trained on the original labeled data. Ensembles are then built by using all classifiers except one, i.e. H_p, for finding a subset of high-confidence unlabeled data with assigning confidence of prediction as weight for newly-labeled instances. These ensembles estimate the error rate for each component classifier from the agreement among the classifiers. After that, a subset of U is selected by ensemble H_p for a component classifier h_j. A pre-defined threshold is used for selecting high-confidence predictions. Data that have an improvement of error above a threshold are added to the labeled training data. Note that each classifier has its own set of training set through the training process. This avoids that classifiers converge too early and strengthens the effect of the ensemble. The data that is labeled for the classifier is not removed from the unlabeled data U to allow it to be labeled for other classifiers as well. This training process is repeated until there are no more data that can be labeled such that they improve the performance of any classifier. A brief outline of the Ensemble-Co-Training algorithm is shown in Figure 2.

In the Ensemble-Co-Training algorithm instead of computing the $\epsilon_{i,j}$, we use the disagreement among classifiers as error rate, called $\hat{e}_{i,j}$. Through the training process the algorithm attempts to decrease the disagreement among classifier for improving the performance.

Note that inequality (7) sometimes could not be satisfied, because the size of $L_{i,j}$ is much larger than the size of $L_{i-1,j}$. In this case the training process cannot reduce the error rate due to stopping the training process earlier. This is because of the fact that the error rate is a worse case error rather than an expectation. To solve this problem, a subsample of

```
Ensemble-Co-Training(L, U, H, $\theta$)
 Input: L: Initial labeled data,
        U: Unlabeled instances,
        H: The N Classifiers,
        $\theta$: The pre-defined confidence Threshold
        $\{P_l\}_{l=1}^{M}$: Prior probability
Begin
      for j  ∈ {1,2,..,k} do
           $h_j \leftarrow$  Learn( $Classifier_j$ ,L)
           $L_{0,j} \leftarrow$  0
           $\hat{e}_{0,j} \leftarrow 0.5$
                // Upper bound of error rate
      end //  for
       $i \leftarrow$ 0 // Number of iteration
      repeat
         $i \leftarrow$  i + 1
        for j  ∈ {1,2,...,k}
           $\hat{e}_{i,j} \leftarrow$ EstimateError($H_p$, L)
          if($\hat{e}_{i,j} < \hat{e}_{i-1,j}$)then
            for each x  ∈ U do
               if(Confidence(x, $H_p$ ) > $\theta$)
                  LabelingUnlabeled(x,$H_p$)
                  $L_{i,j} \leftarrow L_{i,j} \cup \{(x, H_p)\}$
            end// for
            if( $|L_{i-1,j}|$ = 0) then
               $L_{i-1,j} \leftarrow \lfloor \frac{\hat{e}_{i,j}}{(\hat{e}_{i-1,j}-\hat{e}_{i,j})} + 1 \rfloor$
            if(  $|L_{i-1,j}| < |L_{i,j}|$
                  and $\hat{e}_{i,j}|L_{i,j}| < \hat{e}_{i-1,j}|L_{i-1,j}|$) then
              $update_{i,j} \leftarrow$ TRUE
            else if(  $|L_{i-1,j}| > \frac{\hat{e}_{i,j}}{(\hat{e}_{i-1,j}-\hat{e}_{i,j})}$)then
              for each class $l$ , sample  $n_l \propto P_l$
                  as
               $L_{i,j} \leftarrow$
                     $Subsample(L_{i,j}, \lceil \frac{\hat{e}_{i-1,j}|L_{i-1,j}|}{\hat{e}_{i,j}} - 1 \rceil)$
               $update_{i,j} \leftarrow$ TRUE
            end//if
        end// for j
        for j ∈ {1,2,...,k}
           if(  $update_{i,j}$= TRUE ) then
               $h_j \leftarrow Learn(Classifier_j, L \cup L_{i,j})$
               $\hat{e}_{i-1,j} \leftarrow \hat{e}_{i,j}$
               $L_{i-1,j} \leftarrow |L_{i,j}|$
            end// if
        end// for j
        until ( none of  $h_i$  changes )
    end// repeat

    Output: Final H(x) ← argmax  $\Sigma_{i:h_i(x)=y} 1$
End
```

Fig. 3. Ensemble-Co-Training Algorithm

$L_{i,j}$ is randomly selected that does satisfy inequality (7), i.e, $\hat{e}_{i,j}|L_{i,j}| < \hat{e}_{i-1,j}|L_{i-1,j}|$, then will have:

$$|L_{i,j}| = [\frac{\hat{e}_{i-1,j}|L_{i-1,j}|}{\hat{e}_{i,j}}] - c'. \qquad (13)$$

In (12) the size of $L_{i-1,j}$ should be restricted by some criterion. Intuitively, it can be bounded by:

$$|L_{i-1,j}| = [\frac{\hat{e}_{i,j}}{\hat{e}_{i-1,j} - \hat{e}_{i,j}}] + d' \qquad (14)$$

To simplify (13) and (14) assume that $c' = d' = 1$.

Figure 3 presents the pseudo-code of the Ensemble-Co-Training algorithm in more details. The Ensemble-Co-Training algorithm uses inequality (7), (13), and (14) as well as three important functions: the $EstimateError$, the

$LabelingUnlabeled$, and the $finalHypothesis$ function. The $EstimateError(H_p, L)$ function estimates the classification error rate of hypothesis derived from combination of $h_1, ..., h_k$ as the disagreement among classifications such that $k \neq j$ for the component classifier h_j on the training data. Note that our assumption is that training data and newly-labeled data have the same data distribution. Since estimating the classification error rate on the unlabeled instances cannot be done easily, in Ensemble-Co-Training only the initial labeled data are used for estimating error rate.

We use two different error rate estimations in Ensemble-Co-Training. The first approach is the disagreement among predictions by hypotheses produced by different learning algorithms. In addition to this we check if a new example does not increase the error on the labeled data.

The $LabelingUnlabeled$ function labels a subset of high-confidence predictions by H_p for the component classifier h_j. After the last iteration the resulting hypotheses are combined into an ensemble classifier. We experimented with two methods for combining hypotheses: “Average of Probabilities” and “Majority Voting” [23]. In details, suppose that $Y = (y_1, y_2, ..., y_m)$ be the class labels and there are K classifiers. The “Average of Probabilities” voting method for prediction of the new example x is computed as follows:

$$\arg\max_m \left(\frac{1}{K}\sum_{i=1}^{K} p_i(y_m|x)\right), \tag{15}$$

For the “Majority Voting” method, the maximum number of classifiers is considered as a main rule. It means that the majority of the classifiers should be agreed with assigning label y_m to the instance x.

Furthermore, we compare the Ensemble-Co-Training approach without applying the criterion that we mentioned earlier, for showing the impact of the criterion on the performance. In particular we use a version of Democratic Co-Learning [16], which is another form of self-training with multiple classifiers, without the conservative statistical test.

IV. Experiments

Eight UCI datasets [10] are used in our experiments. We selected these datasets because: (i) they involve binary classification, and (ii) these are used in several other studies on Semi-Supervised learning, for example, [17] and [19]. Information about these datasets is in Table I. All sets have two classes and Perc. represents the percentage of the largest class.

Dataset	Attributes	Size	Perc.
Bupa	6	345	58
Colic	22	368	63
Diabetes	6	768	65
Heart	13	270	55
Hepatitis	19	155	79
Ionosphere	34	351	64
Tic-tac-toe	9	958	65
Vote	16	435	61

TABLE I
Overview of Datasets

For each dataset, about 30 percent of the data are kept as test set, and the rest is used as the pool of training instances, which are included a small amount of labeled and a large pool of unlabeled data. Training instances in each experiment are partitioned into 90 percent unlabeled data and 10 percent labeled data, and keeping the class proportions in all sets similar to the original data set. We use five Semi-Supervised learning methods in our experiments: self-training, Tri-training, Co-Forest, Democratic Co-Learning, and Ensemble-Co-Training. In each experiment, eight independent runs are performed with different random partitions of L and U. The average results are summarized in Tables II, III, IV, V, VI, and VII.

As the “base” learner C4.4grafted, which is a decision tree learner [24] with “grafting” and Laplacian correction, adaptations that often improve performance in domains with sparse data, is used in self-training and tri-training algorithms. The Random Forest [25] approach is employed in the Co-Forest learning method. We describe Ensemble-Co-Training for any number of supervised learning algorithms in our empirical work we only consider four learners: C4.4grafted, Naive Bayes, Random forest, and J48 with Laplacian Correction algorithm. To make performance comparable, in co-forest, self-training, and Ensemble-Co-Training we set the value of pre-defined threshold (θ) at 0.75 for all classifiers. We use WEKA [26] classifiers in Java for implementation.

In the first experiment we compare the learning algorithms in the supervised setting. Table II shows the results of classification accuracies. As it can be seen, in general ensemble methods have better accuracy than a single classifier. The best performance is boldfaced for each dataset.

Dataset	Grafted DT	Tri-training	Co-Forest	ECT
Bupa	**58.63**	56.96	55.97	58.01
Colic	80.74	71.30	73.98	**81.02**
Diabetes	65.35	**67.63**	66.49	66.40
Heart Statlog	73.54	**78.41**	72.73	76.14
Hepatitis	72.57	**80.29**	75.71	78.00
Ionosphere	79.13	77.76	82.48	**83.11**
Tic-tac-toe	65.00	65.60	**67.78**	67.66
Vote	94.41	87.53	90.79	**94.64**

TABLE II
Average Classification Accuracy of Supervised learning with Grafted decision tree (DT), Tri-training, Co-Forest, and Ensemble-Co-Training (ECT)

In the second experiment we compare Ensemble-Co-Training, Self-Training, Tri-Training, and Co-Forest in the case of Semi-Supervised data. Table III shows the classification accuracy of the methods. In four out of eight datasets Ensemble-Co-Training has the highest accuracy and in the other sets the differences are small.

In the third experiment we compare different ensemble types in terms of different combining hypotheses methods: “Averaging Probability” (where each prediction is weighed by the posterior probability assigned by the hypothesis) and “Majority voting” (where each hypothesis has one vote). The classification accuracies of these methods are shown in Table IV. Using the “Majority voting” method for estimating the error rate improves the classification accuracy of five data sets out of eight in our experiments.

Dataset	Grafted DT	Tri-training	Co-Forest	ECT
Bupa	57.01	56.97	56.64	**58.58**
Colic	78.91	72.69	76.48	**81.25**
Diabetes	65.88	67.72	**67.94**	66.45
Heart Statlog	71.78	78.25	74.84	**77.44**
Hepatitis	73.20	**82.00**	81.43	80.29
Ionosphere	79.76	79.38	**86.58**	83.48
Tic-tac-toe	67.35	65.83	**68.49**	67.96
Vote	93.94	87.06	92.42	**94.99**

TABLE III
Average Classification Accuracy of Self-training with Grafted decision tree (DT), Tri-training, Co-Forest, and Ensemble-Co-Training (ECT)

Dataset	ECT Averaging Probability	ECT Majority voting
Bupa	58.58	**59.59**
Colic	81.25	**81.39**
Diabetes	66.45	**67.02**
Heart Statlog	**77.44**	76.30
Hepatitis	**80.29**	80.00
Ionosphere	83.48	**83.98**
Tic-tac-toe	67.96	**68.02**
Vote	94.99	94.99

TABLE IV
Average Classification Accuracy of Ensemble-Co-Training (ECT) using Averaging Probability and Majority voting error rate for the $EstimateError$ function

Table V gives the classification accuracies of using different voting methods for the $LabelingUnlabeled$ function in Ensemble-Co-Training, which are again “Averaging Probability” and’“Majority Voting” methods. The best classification accuracies among the different settings in Ensemble-Co-Training are boldfaced in Table V. Using “Majority voting” for the $LabelingUnlabeled$ function in Ensemble-Co-Training archives the best results.

Dataset	ECT Averaging Probability	ECT Majority Voting
Bupa	58.58	**59.94**
Colic	81.25	**82.50**
Diabetes	66.45	67.11
Heart Statlog	77.44	**78.73**
Hepatitis	80.29	81.75
Ionosphere	83.48	**86.71**
Tic-tac-toe	67.96	**69.40**
Vote	94.99	94.59

TABLE V
Average Classification Accuracy of Ensemble-Co-Training(ECT) using Average voting and Majority voting for the $LabelingUnlabeled$ function

In the previous experiment (second) we used the default settings for the learning algorithms. In the next experiment we optimize the parameter settings. This is not possible in practical settings because the amount of data that is needed are normally not available. Yet it puts the non–optimized results in perspective. In particular compare the best current setting of Ensemble-Co-Training and Co-Forest which are almost always the two best methods. Table VI shows the result of comparison. On average the classification accuracy of Ensemble-Co-Training is 1.99% higher than Co-Forest method.

Dataset	Co-Forest	ECT Majority Voting	Improve%
Bupa	56.64(±1.89)	59.94(±2.11)	3.30
Colic	76.48(±3.42)	82.50(±0.84)	6.02
Diabetes	67.94(±1.65)	67.11(±1.11)	-0.83
Heart Statlog	74.84(±3.11)	78.73(±0.48)	3.90
Hepatitis	81.43(±1.5)	81.75(±2.11)	0.32
Ionosphere	86.58(±2.21)	86.71(±1.31)	0.12
Tic-tac-toe	68.49(±1.11)	69.40(±1.24)	0.91
Vote	92.42(±1.63)	94.59(±0.84)	2.17

TABLE VI
Average Classification Accuracy and standard deviation of Co-Forest and Ensemble-Co-training with the best setting

Here, we compare the results of using a version of Democratic co-learning, without the conservative statistical test, and Ensemble-Co-Training without the criterion that we discussed. The aim of this comparison is to show the advantages of using the criterion. As can be seen in Table VII still there is some improvements in the results, but it almost in all dataset is less than using Ensemble-Co-Training with the criterion. The third column (“ECT Majority Voting”) of the Table VI shows the results of using criterion and the second column (“ECT”) of the Table VII depicts the results without employing the criterion. The presented method not only improves the accuracy, but also it reduces the complexity of the training process. However, we believe that this criterion is not perfect method for evaluating the error rate, it still has at least two important effects that we mentioned the above. Another issue of this experiment is that the Ensemble-Co-Training without the criterion also works better than the Democratic co-learning, because it benefits from the co-training approach for the training process.

Dataset	ECT	Democratic Co-Learning
Bupa	**58.74**	57.05
Colic	**82.36**	80.60
Diabetes	**66.27**	66.05
Heart	**76.70**	76.30
Hepatitis	**80.37**	79.60
Ionosphere	82.99	**84.02**
Tic-tac-toe	**67.52**	67.40
Vote	**94.79**	94.48

TABLE VII
Classification Accuracy of Ensemble-Co-training without using the Criterion and Democratic Co-learning

Finally for fair comparison, we compare the performance of all methods that we discussed in this paper. Figure 4 shows the results of the comparison. We observe that the Ensemble-Co-Training with optimize setting gives the best results in all datasets in our experiments.

V. Conclusion and Discussion

We propose a method that uses an ensemble of classifiers for co-training rather than feature subsets. The ensemble is used to estimate the probability of incorrect labeling and this is used with a theorem by Angluin and Laird [8] to derive a measure for deciding if adding a set of unlabeled data will reduce the error of a component classifier or not. Our method does not require a time consuming test for selecting a subset of unlabeled data. Experiments show that in most cases our method outperforms similar methods.

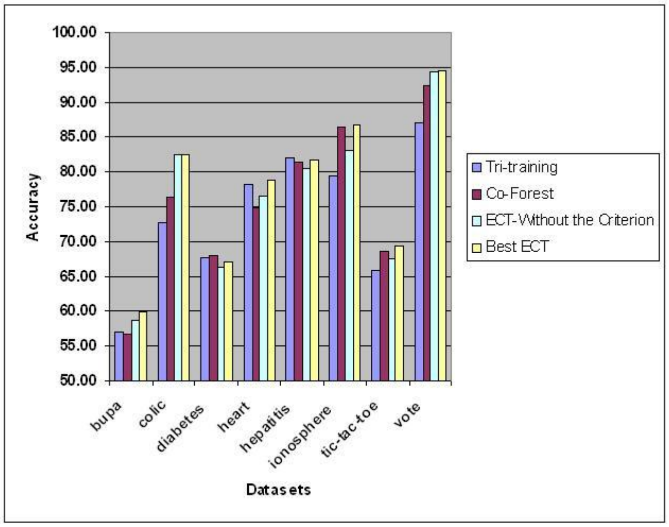

Fig. 4. The Classification Accuracy Comparison of all methods

Balcan and Blum [27] present a general analysis of Semi-Supervised learning with discriminative classifiers (that do not try to model the distribution of the data). They point out that an assumption is required on the relation between the distribution of the data and of the classes. Without such an assumption Semi-Supervised Learning is not possible. Our method is implicitly based on the assumption that the learning algorithms that construct the individual hypotheses bring the relevant learning biases to bear on the data and that their agreement is a good measure for the predictability of the data. The experiments show that the learning algorithms that were included together give good results in a range of application domains that vary substantially in many dimensions. For the future work, it would be interesting to make a further comparison with Disagreement Boosting methods.

References

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. pp. 1–38, 1977. [Online]. Available: http://www.jstor.org/stable/2984875

[2] B. Shahshahani and D. Landgrebe, "The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 32, no. 5, pp. 1087 –1095, Sep. 1994.

[3] V. Vapnik, *Statistical learning theory*. Berlin: Springer, 1998.

[4] K. P. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Advances in Neural Information Processing Systems*. MIT Press, 1998, pp. 368–374.

[5] X. Zhu, "Semi-supervised learning literature survey," 2006.

[6] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, ser. COLT' 98. New York, NY, USA: ACM, 1998, pp. 92–100. [Online]. Available: http://doi.acm.org/10.1145/279943.279962

[7] S. Goldman and Y. Zhou, "Enhancing supervised learning with unlabeled data," in *IN PROCEEDINGS OF THE 17TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING*. Morgan Kaufmann, 2000, pp. 327–334.

[8] D. Angluin and P. Laird, "Learning from noisy examples," *Machine Learning*, vol. 2, pp. 343–370, 1988, 10.1023/A:1022873112823. [Online]. Available: http://dx.doi.org/10.1023/A:1022873112823

[9] S. Dasgupta, D. McAllester, and M. Littman, "PAC Generalization Bounds for Co-Training," in *Proceedings of Neural Information Proccessing Systems*, 2001.

[10] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[11] B. Wang, B. Spencer, C. X. Ling, and H. Zhang, "Semi-supervised self-training for sentence subjectivity classification," in *Proceedings of the Canadian Society for computational studies of intelligence, 21st conference on Advances in artificial intelligence*, ser. Canadian AI'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 344–355. [Online]. Available: http://portal.acm.org/citation.cfm?id=1788714.1788746

[12] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *WACV/MOTION*. IEEE Computer Society, 2005, pp. 29–36.

[13] Y. Li, C. Guan, H. Li, and Z. Chin, "A self-training semi-supervised svm algorithm and its application in an eeg-based brain computer interface speller system," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1285 – 1294, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/B6V15-4RV7Y65-3/2/aa683a67e75ea176ecbbda80d160aea8

[14] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *CIKM*. ACM, 2000, pp. 86–93.

[15] U. Brefeld and T. Scheffer, "Co-em support vector learning," in *Proceedings of the twenty-first international conference on Machine learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 16–. [Online]. Available: http://doi.acm.org/10.1145/1015330.1015350

[16] Y. Zhou and S. Goldman, "Democratic co-learning," *Tools with Artificial Intelligence, IEEE International Conference on*, vol. 0, pp. 594–202, 2004.

[17] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 1529–1541, 2005.

[18] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996, 10.1007/BF00058655. [Online]. Available: http://dx.doi.org/10.1007/BF00058655

[19] M. Li and Z.-H. Zhou, "Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 37, no. 6, pp. 1088 –1098, 2007.

[20] A. Sharma, G. Hua, Z. Liu, and Z. Zhang, "Meta-tag propagation by co-training an ensemble classifier for improving image search relevance," *Computer Vision and Pattern Recognition Workshop*, vol. 0, pp. 1–6, 2008.

[21] B. Leskes and L. Torenvliet, "The value of agreement a new boosting algorithm," *J. Comput. Syst. Sci.*, vol. 74, no. 4, pp. 557–586, 2008.

[22] J. Tanha, M.V.Someren, and H. Afsarmanesh, "Ensemble-training: Ensemble based co-training," in *Benelearn*, 2011, p. 7.

[23] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, 1998.

[24] G. I. Webb, "Decision tree grafting from the all tests but one partition," in *IJCAI*, T. Dean, Ed. Morgan Kaufmann, 1999, pp. 702–707.

[25] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001, 10.1023/A:1010933404324. [Online]. Available: http://dx.doi.org/10.1023/A:1010933404324

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, November 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[27] M.-F. Balcan and A. Blum, "A discriminative model for semi-supervised learning," *J. ACM*, vol. 57, no. 3, 2010.