Ensemble based co-training

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

Publication date
2011

Document Version
Final published version

Published in
Proceedings of the 23rd Benelux Conference on Artificial Intelligence

Link to publication

Citation for published version (APA):
Tanha, J., van Someren, M., & Afsarmanesh, H. (2011). Ensemble based co-training. In P. De Causmaecker, J. Maervoet, T. Messelis, K. Verbeeck, & T. Vermeulen (Eds.), *Proceedings of the 23rd Benelux Conference on Artificial Intelligence: 3-4 November 2011, Ghent, Belgium* (pp. 223-231). (BNAIC; Vol. 23). BNAIC. http://allserv.kahosl.be/bnaic2011/digital_proceedings?page=6&order=field_papertype_value&sort=asc

Ensemble Based Co-Training

Jafar Tanha Maarten van Someren Hamideh Afsarmanesh

[a] *Informatics Institute, University of Amsterdam,*
Science Park 904, 1098 XH Amsterdam, The Netherlands

Abstract

Recently Semi-Supervised learning algorithms such as co-training are used in many application domains. In co-training, two classifiers based on different views of data or on different learning algorithms are trained in parallel and then 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 produces an approximately correct identification with high probability for reliable examples, 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 other methods.

1 Introduction

In many data mining domains, such as object detection, web-page categorization, or e-mail classification, there is only a limited number of labeled data. The labels are often expensive or time consuming to obtain. At the same time, there is often access to a large pool of unlabeled data. 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 improving the performance of supervised learning using only a small amount of labeled instances. There are several different methods for Semi-Supervised learning based on different assumptions. Examples are Expectation-Maximization, Transductive Support Vector Machine, and Graph-based models [16, 4]. Co-training is [2] as a widely used Semi-Supervised learning method. It 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.

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 criterion, using a theorem by Angluin and Laird [1] that describes the effect of learning from uncertain data. This criterion is used for estimates of confidence of predictions for unlabeled data. This is useful because Semi-Supervised learning is used in settings where only a small amount of labeled data is available. This approach does not require techniques like cross-validation or bagging, both of which are not efective in applications with a small amount of labeled data. It has two parameters, taken from PAC-learning (Probably Approximately Correct) [5], with an intuitive meaning: the probability and size of a prediction error. Our experiments on UCI datasets [6] show that Ensemble-Co-Training improves the performance of classification more than the other methods.

2 Related Work

The co-training paradigm that was proposed by Blum and Mitchell [2] works well in domains that naturally have multiple views of data. Nigam and Ghani [11] analyzed the effectiveness and applicability of co-training when there not two natural views of the data. They show that when independent and redundant views exist, co-training algorithms outperform the 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.

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 [14]. 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 also they use a statistical method for selecting unlabeled data for labeling. It does not rely on the existence of two views, but their method uses a time-consuming method for selecting unlabeled data for labeling.

Zhou and Li [15] 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. 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 needs 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 [10]. 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.

3 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 [12]. 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 among 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 3.2.

3.1 Criterion for error rate and number of unlabeled data

3.1.1 Basic Definitions

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. Also, 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.

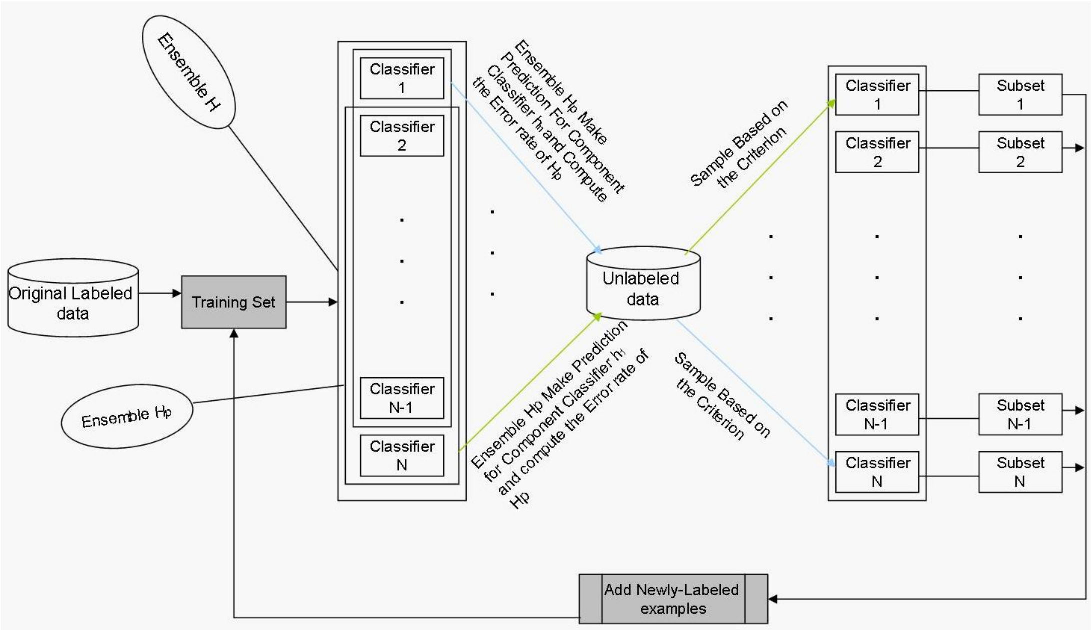
Ensemble H
Original Labeled data
Training Set
Ensemble Hp
Classifier 1
Classifier 2
Classifier N-1
Classifier N
Ensemble Hp Make Prediction For Component Classifier hm and Compute the Error rate of Hp
Ensemble Hp Make Prediction for Component Classifier hi and compute the Error rate of Hp
Unlabeled data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

Figure 1: Block diagram of the proposed Ensemble-Co-Training

3.1.2 Learning from noisy data

As we observed, using bagging for generating different hypothesis 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 [15], 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 [1]. 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 finding the error rate, we need to estimate η. This is done using a set of hypotheses that are constructed by different learning algorithms. 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, at 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 at the ith iteration of the 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 at 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:

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

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:

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

According to (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 $\epsilon_{i,j} < \epsilon_{i-1,j}$.

```
• 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 $h_j$ regarding
         inequality (7)
     - 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
Output:
• Generate final hypothesis
```

Figure 2: An outline of the Ensemble-Co-Training

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.

3.2 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 and the size of subset is controlled by (12). 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 the 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. We use different error rate estimations in Ensemble-Co-Training. One approach is the disagreement among predictions by hypotheses produced by different learning algorithms.

The $AssiningLabel$ function in Figure 2 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 and for finding error rate: “Average of Probabilities” and “Majority Voting”[9].

4 Experiments

Eight UCI datasets [6] 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, [15] and [10]. Information about these datasets is in Table 1. 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	21
Ionosphere	34	351	36
Tic-tac-toe	9	958	65
Vote	16	435	61

Table 1: 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 four Semi-Supervised learning methods in our experiments: Self-training [4], Tri-training, Co-Forest, 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 2, 3, and Figure 3.

As the "base" learner C4.4grafted, which is a decision tree learner [13] with "grafting" and Laplacian correction, adaptations that often improve the performance in domains with sparse data, is used in self-training and tri-training algorithms. The Random Forest [3] 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 [8] classifiers in Java for implementation.

In the first experiment we compare Ensemble-Co-Training, Self-Training, Tri-Training, and Co-Forest in the case of Semi-Supervised data. Table 2 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.

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

In the second 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). Using the "Majority voting" method for estimating the error rate also improves the classification accuracy in our experiments. The results of this settings, which is almost the best setting, for Ensemble-Co-Training are shown in Figure 3.

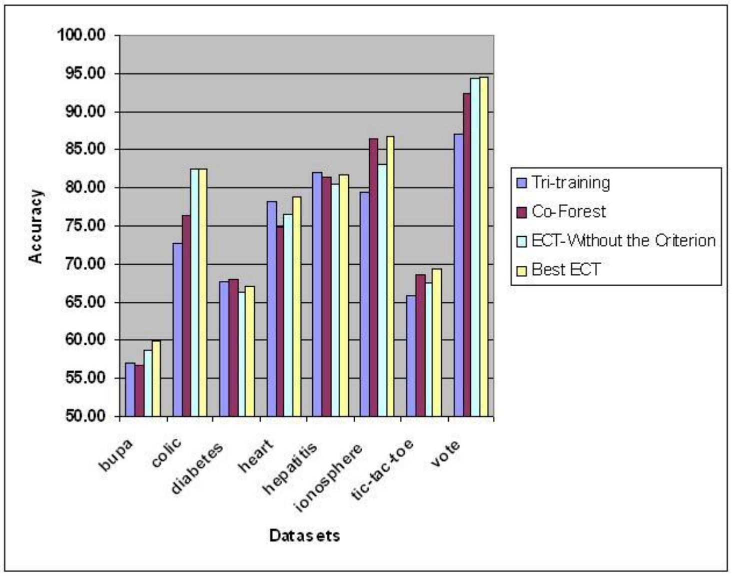

Figure 3: The Classification Accuracy Comparison of all methods

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 3 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 3: Average Classification Accuracy and standard deviation of Co-Forest and Ensemble-Co-training with the best setting

Finally we compare the performance of all methods that we discussed in this paper. We also report additional experiment on Ensemble-Co-training without using the criterion. The aim of this comparison is to show the advantages of using the criterion. As can be seen in Figure 3 still there is some improvements in the results, but it almost in all datasets is less than using Ensemble-Co-Training with the criterion. We observe that the Ensemble-Co-Training with optimize setting also gives the best results in all datasets in our experiments. 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.

5 Conclusion and Discussion

The co-training algorithms also need a solution for two problems: estimating confidence in predicted labels (to select unlabeled data for labeling) and a stopping criterion. 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 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. The classifiers must be sufficiently different in ensemble learning to improve over the individual classifiers. This is also true for our method. The experiments show that this approach works well in practice on most of the datasets on which it was evaluated.

References

[1] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2:343–370, 1988. 10.1023/A:1022873112823.

[2] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, COLT' 98, pages 92–100, New York, NY, USA, 1998. ACM.

[3] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. 10.1023/A:1010933404324.

[4] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. The MIT Press, September 2006.

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

[6] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[7] Sally Goldman and Yan Zhou. Enhancing supervised learning with unlabeled data. In *IN PROCEEDINGS OF THE 17TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING*, pages 327–334. Morgan Kaufmann, 2000.

[8] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18, November 2009.

[9] Josef Kittler, Mohamad Hatef, Robert P.W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:226–239, 1998.

[10] Ming Li and Zhi-Hua Zhou. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 37(6):1088 –1098, 2007.

[11] Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *CIKM*, pages 86–93. ACM, 2000.

[12] J. Tanha, M.V.Someren, and H. Afsarmanesh. Ensemble-training: Ensemble based co-training. In *Benelearn*, page 7, 2011.

[13] Geoffrey I. Webb. Decision tree grafting from the all tests but one partition. In Thomas Dean, editor, *IJCAI*, pages 702–707. Morgan Kaufmann, 1999.

[14] Yan Zhou and Sally Goldman. Democratic co-learning. *Tools with Artificial Intelligence, IEEE International Conference on*, 0:594–202, 2004.

[15] Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17:1529–1541, 2005.

[16] Xiaojin Zhu. Semi-supervised learning literature survey, 2006.