ICIC Express Letters
Volume **10**, Number **11**, November **2016**

 ISSN 1881-803X
pp. **2533–2538**

IMPROVING CO-TRAINING ALGORITHM WITH ACTIVE LEARNING

Xitao Zou[1], Jiang Xiong[1] and Xianchun Zou[2]

[1]College of Computer Science and Engineering
Chongqing Three Gorges University
No. 666, Tianxing Road, Wanzhou Dist., Chongqing 404000, P. R. China
xiaotao1009@sina.cn

[2]College of Computer and Information Science
Southwest University
No. 2, Tiansheng Road, Beibei Dist., Chongqing 400715, P. R. China
zouxc@swu.edu.cn

Received April 2016; accepted July 2016

Abstract. *Co-training is a seminal semi-supervised learning algorithm. However, in co-training, especially at the initial stage, there are too few labeled samples to train strong classifiers, which may lead to the introduction of noisy labels in the following steps. Aiming at this problem, in this paper, two measures are taken to improve co-training. Firstly, to increase the number of labeled samples, we define a strategy of uncertainty sampling and then cooperate active learning into co-training. Secondly, to alleviate the introduction of noisy labels, we put forward a simple but efficient method to evaluate the labeling confidence. Experiments on several UCI datasets show that our algorithm performs better than several relative co-training algorithms on classification accuracy.*
Keywords: Co-training, Active learning, Semi-supervised classification, Noisy labels

1. **Introduction.** Co-training is one of the most prominent semi-supervised learning algorithms. In 1998, Blum and Mitchell proposed the original co-training algorithm [1]. Co-training builds two classifiers on two sufficient and redundant views of the initial labeled samples, and in each iteration, each classifier labels samples for the other classifiers. Then the two classifiers retrain themselves with the expanded labeled samples. After original co-training algorithm, large numbers of researchers paid a great deal of attention to co-training. Goldman and Zhou presented a new co-training algorithm which does not need two sufficient and redundant views [2]. Instead, they train two classifiers with two different decision trees on the same attributes of labeled samples. Furthermore, to decrease constraints of co-training, Zhou and Li proposed tri-training [3]. In tri-training, neither sufficient views nor redundant views are necessary. Three classifiers are built on the subset of labeled samples which are chosen from the initial labeled samples by bootstrap [4].

Co-training algorithms above can often achieve very good classification accuracy. However, because the small number of labeled samples in co-training, especially in the initial stage, classification ability of classifiers built on these labeled samples is extremely weak. As a result, these weak classifiers may label the unlabeled samples incorrectly and bring in noisy labels for the later training process. To relieve this problem, a host of measures are taken, while cooperating active learning into co-training is one of the most outstanding ones.

Active learning is another subfield of semi-supervised algorithm [5]. Active learning is made up of two components: sampling and human annotation [6]. The goal of sampling is to select some unlabeled samples based on certain criteria, while human annotation

aims at labeling samples by annotators. The key point of active learning is how to select unlabled samples [6]. Generally speaking, there are several well-known measures for sampling: uncertainty sampling [7], query by committee [8], expected gradient length [9] and so on.

In the framework of uncertainty sampling, an active learner selects and labels the samples which are most uncertain to label. While active learning with query by committee maintains a committee such as classifiers trained on the current set of labeled samples. Every committee member is allowed to vote on the labels of unlabeled samples. The samples with the most disagreement on their labels are selected and taken to label by annotators in active learning. In expected model change framework, active learning selects and labels unlabeled samples that would impart the greatest change to the current model.

In order to enhance the classification accuracy of co-training, many researchers have worked out their solutions by combining co-training with active learning. Zhan and Chen [10] introduced human-computer interactions into co-training to avoid the rejective judgment when the classifiers do not agree with each other and the inaccurate judgment when the base classifiers all agree with each other. Xie and Liu [11] applied active learning and ensemble learning to co-training. Zhang et al. [12] put forward a semi-supervised learning algorithm by combining the benefits of co-training and active learning. This algorithm applies two different techniques, selecting high confidence and nearest neighbor to label, exploiting the most informative instances with human annotation, to boost the classification accuracy. Algorithms in [10, 11, 12] can always achieve higher classification accuracy, but few of them take the classification boundary into consideration, which limits the increment of performance of these algorithms.

To combine co-training with active learning efficiently, decrease the introduction of noisy labels and improve the efficiency of co-training, in this paper, we combine co-training with active learning and come up with a new semi-supervised learning algorithm (Improve-CoTrain in short). In Improve-CoTrain, two criteria are induced. On one hand, we define a novel method of uncertainty sampling to select unlabeled samples in active learning and then cooperate it into co-training. Specifically, in each iteration of Improve-CoTrain, we build semi-supervised linear classifiers from labeled samples and then calculate the labeling uncertainty of unlabeled samples, and the unlabeled samples with the most labeling uncertainty will be labeled by annotators and used to train classifiers. On the other hand, to evaluate the labeling confidence, we calculate the differences of labeling vectors from the two auxiliary classifiers when labeling the same unlabeled samples, and the unlabeled samples with the least differences are labeled and taken to the subset of labeled samples of the main classifier.

The rest of this paper is organized as follows. Section 2 defines the measures of uncertainty sampling and labeling confidence. Section 3 presents the details of Improve-CoTrain. Section 4 is the relative experiments and Section 5 draws conclusions on this paper.

2. **Research Methods.**

2.1. **Uncertainty sampling.**

In uncertainty sampling methods, if an unlabeled sample belongs to a classification with a probability which is very close to 0.5, then the label of the sample is with the most uncertainty. Based on this, we define a new method to evaluate the labeling uncertainty of samples. The formula to calculate the labeling uncertainty is as follows.

$$Uncertainty(x_i) = \frac{4}{C}\sum_{c=1}^{C} p_{ic}(1 - p_{ic}) \tag{1}$$

where C represents the total number of categories in classification. p_{ic} is the value in row i and column c of matrix Y_U. The bigger the $Uncertainty(x_i)$ is, the more labeling uncertainty the x_i has.

To calculate the sample labeling uncertainty, we build semi-supervised linear classifier (SLC) on labeled samples to acquire the classification matrix Y_U ($Y_U \in R^{|U|\times C}$) of unlabeled samples set U. The values in row i of Y_U represent the labeling vector y_i of sample x_i, $y_i = [p_{i1}, p_{i2}, \ldots, p_{iC}]$, and p_{ic} is the value in column c of y_i.

$$p_{ic} = \begin{cases} 1, & \text{if } label(x) = c \\ 0, & \text{if } label(x) \neq c \end{cases} \tag{2}$$

The semi-supervised linear classifier is specified as:

$$f(x) = P^T x + b \tag{3}$$

where $x \in R^{D\times 1}$ is an unlabeled sample, and D is the dimension of samples x. $P \in^{D\times C}, b \in R^{C\times 1}$. C represents the total number of categories in classification. $f(x) \in R^{C\times 1}$ is the classification vector of sample x on SLC. For each $x_i \in U$, we can work out $f(x_i)$ by using Equation (3), and $f^T(x_i)$ is regarded as the labeling vector of x_i and added to the row i of Y_U.

To solve the linear classification, we define the deviation function of the linear classifier as follows.

$$\Psi(x, P, b) = \sum_{i=1}^{|L|} \|Px_i + b - y_i\|^2 + \frac{\alpha}{2} \sum_{i,j=1}^{N} \|f(x_i) - f(x_j)\|^2 w_{ij} \tag{4}$$

where $|L|$ is the number of labeled samples and N is the number of labeled and unlabeled samples. w_{ij} is the weight between samples x_i and x_j. α balances the importance of two parts.

We take partial derivative of $\Psi(x, P, b)$ with respect to P and b, and let $\frac{\partial \Psi}{\partial P} = 0$, $\frac{\partial \Psi}{\partial_b} = 0$, and then we can acquire the value of P and b. For the detailed solution of the semi-supervised linear classifier, you can read [13].

2.2. **Labeling confidence.** In co-training, when labeling an unlabeled sample, if the similarity of the label vectors generated by two auxiliary classifiers is quite high, then it demonstrates that this unlabeled sample is easy to be labeled correctly. Based on this, we define the strategy of labeling confidence as follows.

$$h_t({x_i}^t) = \sum_{C_a \in A_t, C_b \in A_t} \left\| f_a({x_i}^a) - f_b({x_i}^b) \right\| \tag{5}$$

where $h_t({x_i}^t)$ is the predictive confidence of unlabeled sample x_i when the main classifier is C_t, and $f_a({x_i}^a)$ is the predicted label vector of unlabeled sample x_i with auxiliary classifier C_a. $A_t = \{\{C_1, C_2, \ldots, C_T\} - \{C_t\}\}$. The smaller $h_t({x_i}^t)$ is, the higher the labeling confidence is.

3. **Improve-CoTrain.** Based on the uncertainty sampling method and labeling confidence measure above, we put forward a semi-supervised learning algorithm, Improve-CoTrain in short. The detailed description of Improve-CoTrain is depicted in Algorithm 1.

4. **Experiments.** In this section, to verify the performance of Improve-CoTrain, we experiment on four datasets from UCI [14]. The name, number of features, number of samples, positive rate and negative rate of each dataset are depicted in Table 1.

In experiments, for each experimental dataset, we randomly take 80% samples as training data while the rest as test data. In training data, 20% samples are selected as initial labeled data and the rest as unlabeled data.

Algorithm 1 Improve-CoTrain: Improving Co-Training Algorithm with Active Learning

Input:

L: Original labeled sample set;
U: Unlabeled sample set;
$Learner$: Learning algorithm;
$iter\_num$: the iteration time of algorithm.

Output:

for $t = 1$ *to* 3 **do**
  $S_t \leftarrow BootstrapSample(L)$//random sampling
  $classifier_t = Learner(S_t)$//train classifier $classifier_t$ using labeled sample set S_t
end for
for $iter = 1$ *to* $iter\_num$ **do**
  Building SLC on L by using Equations (2), (3) and (4)
  Calculating Y_U of U by SLC
  Calculating $Uncertaint y(x_i)$ ($x_i \in U$) by using Y_U and Equation (1)
  Labeling the N_1 unlabeled samples with the highest uncertainty by annotator
  Adding the N_1 newly labeled samples to L and S_t ($t = 1, 2, 3$), then deleting them from U
  for $t = 1$ *to* 3 **do**
    $U_{xyh} = \emptyset$
    for $i = 1$ *to* $|U|$ **do**
      Labeling x_i by $classifier_{t1}$, $classifier_{t2}$ ($t1 \neq t2 \neq t$, $t1, t2 = 1, 2, 3$)
      if label(x_i^{t1}) = label(x_i^{t2}) **then**
      Calculating the labeling confidence of x_i by Equation (5)
      Adding (x_i, label(x_i^{t1}), $h_t({x_i}^t)$) into U_{xyh}
      end if
    end for
    Taking N_2 samples (with their labels) who have the highest labeling confidence from U_{xyh} to U_{xy}
    $S_t \leftarrow S_t \cup U_{xy}$
    $classifier_t = Learner(S_t)$
  end for
end for
$classifier \leftarrow$ Ensemble ($classifier_1$&$classifier_2$&$classifier_3$)//Integrating base classifiers for ensemble prediction
return $classifier$

Table 1. Experimental datasets

dataset	#features	#samples	#pos/#neg
australian	14	690	55.5%/44.5%
diabetes	8	768	65.1%/34.9%
german	20	1000	70.0%/30.0%
wdbc	30	569	37.3%/62.7%

In the first experiment, to demonstrate the validation of Improve-CoTrain, we compare it with tri-training in [3], random sampling (the sample selection method in Improve-CoTrain is placed by random sampling), and NF-CT-SSAL in [10]. What is more, we train 20 times on each dataset to overcome the random results. The learning algorithm to train classifiers in experiment is BP neural network. We set $iter\_num = 6$ and record the error rate of each comparing algorithm in each iteration. The results on each of four datasets are presented in Figure 1.

From Figure 1, we can observe that Improve-CoTrain usually outperforms its comparing algorithms on classification accuracy. So we draw a conclusion that Improve-CoTrain can effectively avoid the introduction of noisy data and improve the efficiency of co-training algorithm.

In another experiment, we try to investigate the performance of Improve-CoTrain under different ratios of labeled samples. In the process of experiment, the ratio of labeled samples is varied from 0.2 to 0.8 with step-size 0.2. The algorithm to train classifiers is also BP neural network. After setting $iter\_num = 6$, we experiment on the four datasets

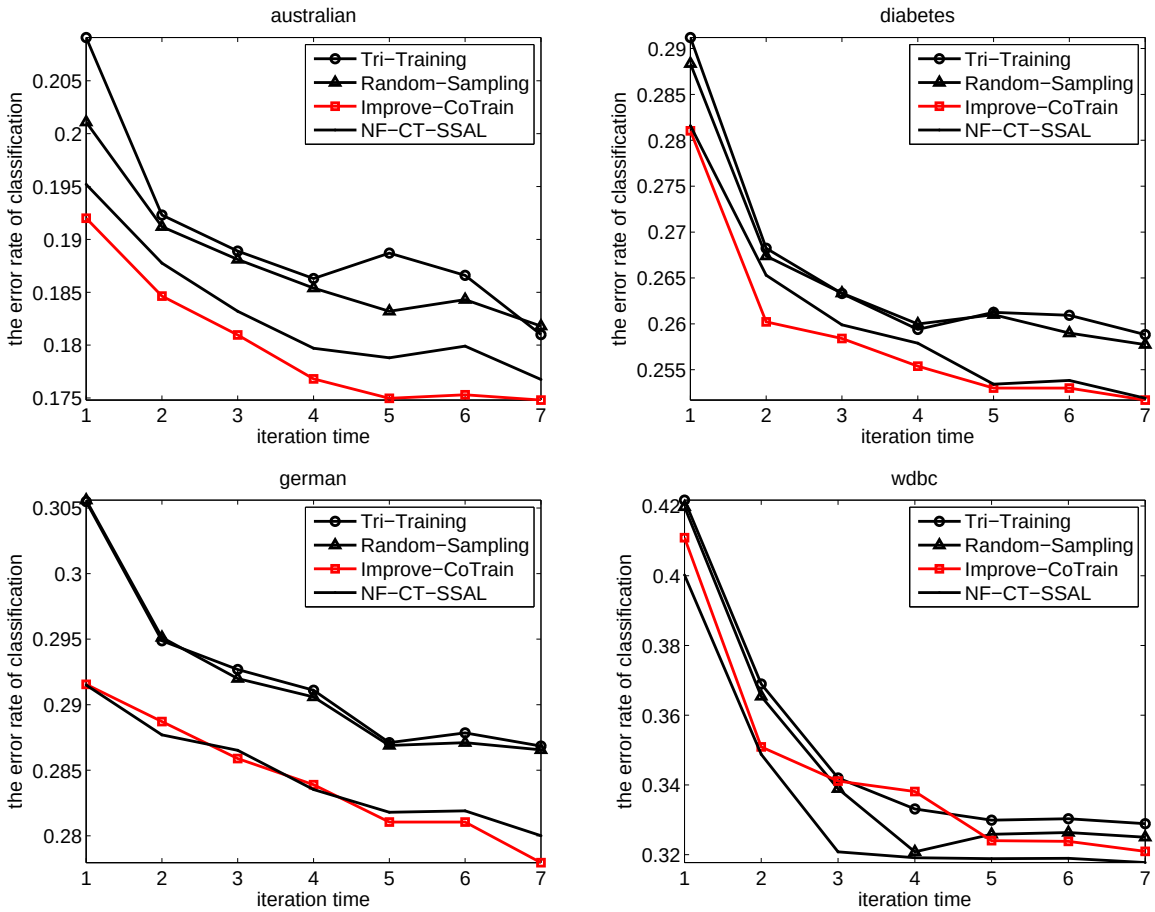

FIGURE 1. Error rate of tri-training, random sampling, Improve-CoTrain, and NF-CT-SSAL on different experimental datasets

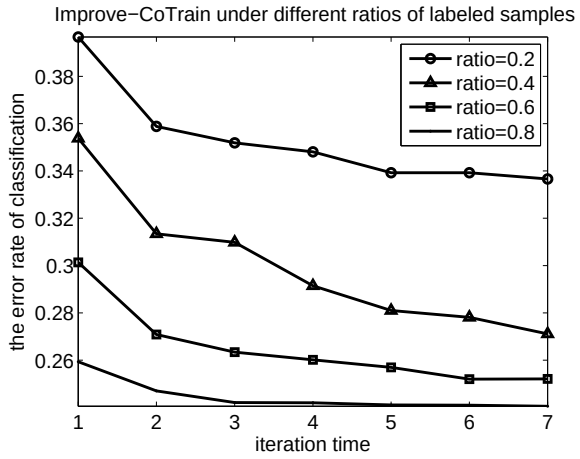

FIGURE 2. Performance of Improve-CoTrain under different ratios of labeled samples

separately, and acquire the error rate of classification in each iteration. Towards each ratio of labeled samples, the average error rate of classification in each iteration is calculated and exhibited in Figure 2.

In Figure 2, we can know that under different ratios of labeled samples, Improve-CoTrain always performs very good on classification. However, it is obvious that Improve-CoTrain can acquire better performance when the ratio of labeled samples is 40% or 60%.

From the point of my view, this is partly because when the ratio of labeled samples is 20%, the basic classifiers in Improve-CoTrain are very weak, and while the ratio of labeled samples is 80%, active learning in Improve-CoTrain takes litter effect.

5. **Conclusions.** In this paper, to investigate the performance of co-training and prevent the influence of noisy labels in co-training, we define a measure of sample selection in active learning, and the selected unlabeled samples with a high sampling uncertainty are labeled by annotator and used to retrain classifiers in co-training. Besides, we put forward a formula to calculate the labeling confidence of each unlabeled samples. Finally, we conduct experiments on several datasets of UCI to find out the performance of our algorithm. The experimental results demonstrate that Improve-CoTrain works very well on classification. As the method of calculating labeling confidence in this paper is only suitable for co-training algorithms which have two auxiliary classifiers, our goal in the future is to work out better measures of calculating labeling confidence.

Acknowledgment. This paper is partially supported by grants from the National Natural Science Foundation of China (Project No. 61101234). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] A. Blum and T. Mitchell, Combining labeled and unlabeled data with co-training, *Proc. of the 11th Annual Conference on Computational Learning Theory*, pp.92-100, 1998.
[2] S. Goldman and Y. Zhou, Enhancing supervised learning with unlabeled data, *Proc. of the 17th International Conference on Machine Learning*, San Francisco, CA, pp.327-334, 2000.
[3] Z. Zhou and M. Li, Tri-training: Exploiting unlabeled data using three classifiers, *IEEE Trans. Knowledge and Data Engineering*, vol.17, no.11, pp.1529-1541, 2005.
[4] S. Abney, Bootstrapping, *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, pp.360-367, 2002.
[5] B. Settles, Active learning literature survey, *Technical Report 1648*, University of Wisconsin-Madison, 2009.
[6] J. Long, P. Yin, E. Zhu and W. Zhao, A survey of active learning, *Journal of Computer Research and Development*, vol.45, pp.300-304, 2008.
[7] D. Lewis and W. Gale, A sequential algorithm for training text classifiers, *Proc. of the ACM SIGIR Conference on Research and Development in Infromation Retrieval*, pp.3-12, 1994.
[8] H. S. Seung, M. Opper and H. Sompolinsky, Query by committee, *Proc. of the ACM Workshop on Computational Learning Theory*, pp.287-294, 1992.
[9] B. Settles, M. Craven and S. Ray, Multiple-instance active learning, *Advances in Neural Information Processing Systems*, vol.20, pp.1289-1296, 2008.
[10] Y. Zhan and Y. Chen, Co-training semi-supervised active learning algorithm with noise filter, *Pattern Recognition and Artificial Intelligence*, vol.22, no.5, pp.750-755, 2009.
[11] H. Xie and M. Liu, An ensemble co-training algorithm based on active learning, *Journal of Shandong University (Science Edition)*, vol.42, no.3, pp.1-5, 2012.
[12] Y. Zhang, J. Wen, X. Wang and Z. Jiang, Semi-supervised learning combining co-training with active learning, *Expert Systems with Applications*, vol.41, no.5, pp.2372-2378, 2014.
[13] G. Yu, G. Zhang, Z. Zhang, Z. Yu and L. Deng, Semi-supervised classification based on subspace sparse representation, *Knowledge and Information Systems*, vol.43, no.1, pp.81-101, 2015.
[14] A. Asuncion and D. J. Newman, *UCI Repository of Machine Learning Databases*, School of Information and Computer Science, University of California, Irvine, 2007.