Front. Comput. Sci., 2019, 13(1): 99–105
https://doi.org/10.1007/s11704-018-7138-5

RESEARCH ARTICLE

Towards making co-training suffer less from insufficient views

Xiangyu GUO, Wei WANG (✉)

National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

© Higher Education Press and Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract Co-training is a famous semi-supervised learning algorithm which can exploit unlabeled data to improve learning performance. Generally it works under a two-view setting (the input examples have two disjoint feature sets in nature), with the assumption that each view is sufficient to predict the label. However, in real-world applications due to feature corruption or feature noise, both views may be insufficient and co-training will suffer from these insufficient views. In this paper, we propose a novel algorithm named Weighted Co-training to deal with this problem. It identifies the newly labeled examples that are probably harmful for the other view, and decreases their weights in the training set to avoid the risk. The experimental results show that Weighted Co-training performs better than the state-of-art co-training algorithms on several benchmarks.

Keywords semi-supervised learning, co-training, insufficient views

1 Introduction

Traditional machine learning needs a large amount of labeled data to learn a good model. But it is expensive to assign labels to the training data since it requires human efforts. In real-world applications, unlabeled data are generally easy to collect (e.g., with several lines of code you can implement a crawler to crawl thousands of images in a few hours). During the past decade, researchers have shown great interest in exploiting these abundant unlabeled data to help the learning process. One famous paradigm is semi-supervised learning, which automatically exploits unlabeled data in addition to labeled data to improve learning performance. Roughly speaking, semi-supervised learning algorithms can be classified into four categories, i.e., generative methods [1, 2], S3VM [3, 4], graph-based methods [5–7], and disagreement-based methods [8, 9]. Disagreement-based methods [10] usually train multiple classifiers for the same task and exploit the disagreement among them to improve their performance, where unlabeled data serve as a platform for exchanging information between classifiers.

Co-training [8] is a representative of the disagreement-based method. It trains two classifiers separately on two sufficient and redundant *views* (i.e., two disjoint attribute sets) and lets them label some unlabeled data for each other. Sometimes the data don't have two views, and researchers have attempted to make co-training work on single-view data as well. Nigam and Ghani [11] showed that when there is one redundant attribute set, it can be randomly divided into two views to apply co-training. However, if the attribute set isn't redundant enough, this method may perform poorly. Some variants of co-training have been developed which can work well on single-view data: Goldman and Zhou [12] used two different decision tree algorithms to train two different models from a single view, and let them label data for each other; Zhou and Li [9] proposed the *tri-training* algorithm which trains three classifiers, and assigns labels to unlabeled data by majority voting. Co-training and its variants have achieved great success in many applications, such as natural language processing [13–15] and computer vision [16].

There are also some studies on the theoretical supports of co-training. Blum and Mitchell [8] showed that *conditional independence* between the two views is sufficient for co-training to succeed. Later, some researchers showed that weaker assumptions can also guarantee the success of co-

Received April 20, 2017; accepted August 7, 2017

E-mail: wangw@lamda.nju.edu.cn

training, e.g., the *weak dependence* assumption proposed by Abney [17] and the *ϵ-expansion* proposed by Balcan et al. [18]. Finally Wang and Zhou [19] found a sufficient and necessary condition for co-training to succeed. As for the single-view setting, Wang and Zhou [20] proved that so long as the two classifiers have large diversity, co-training style algorithm can improve the learning performance by exploiting unlabeled data. However, all these works assume that the two views are sufficient to predict the label. In practice, both views may be insufficient due to feature noise or corruption. Wang and Zhou [21] found that this view insufficiency can lead to performance degradation of co-training. Nevertheless, they don't give any algorithm to deal with insufficient views. This inspired us to design a better co-training style algorithm that can work with insufficient views.

In this paper, we propose a novel algorithm *Weighted Co-training* to address the view insufficiency problem. Under insufficient views the two optimal classifiers in the two views are no longer compatible, some data labeled by one view may be harmful for the other to learn the optimal classifier. We attempt to detect the potentially harmful data and decrease their weights in the training set, which can avoid the risk of degrading the performance. The experimental results show that the proposed algorithm performs better than the state-of-art co-training algorithms on several benchmarks.

The rest of this paper is organized as follows: in Section 2 we give some notations and definitions, then in Section 3 we present our algorithm; we conduct the experiments in Section 4 and make a conclusion in Section 5.

2 Preliminaries

Let $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ denote the instance space, where $\mathcal{X}_1$ and $\mathcal{X}_2$ represent the two views. We focus on binary classification, i.e., the label belongs to $\mathcal{Y} = \{-1, +1\}$. Let $L \cup U$ denote the data set, where $L = \{(\langle x_{1,1}, x_{2,1}\rangle, y_1), \ldots, (\langle x_{1,l}, x_{2,l}\rangle, y_l)\} \subset \mathcal{X} \times \mathcal{Y}$ is the set of labeled data, and $U = \{\langle x_{1,l+1}, x_{2,l+1}\rangle, \ldots, \langle x_{1,l+u}, x_{2,l+u}\rangle\} \subset \mathcal{X}$ is the set of unlabeled data. Let $\mathcal{F}_v : \mathcal{X}_v \mapsto [-1, +1]$ denote the hypothesis space for $\mathcal{X}_v$ ($v = 1, 2$). For some hypothesis f_v, the corresponding label it predicted on x_v can be denoted as

$$\widehat{y_v} = \begin{cases} +1, & \text{if } f_v(x_v) > 0, \\ -1, & \text{otherwise.} \end{cases}$$

The optimal classifier in $\mathcal{F}_v$ is denoted as $f_v^* = \arg\min_{f \in \mathcal{F}_v} P(\mathrm{sign}(f(x_v)) \neq y)$. We follow the definition of insufficiency given by Wang and Zhou [21]:

Definition 1 (Insufficiency) Let $\mathbb{D}$ denote the unknown data distribution over $\mathcal{X}_v \times \mathcal{Y}$. The insufficiency $\Upsilon(\mathcal{X}_v, \mathcal{Y}, \mathbb{D})$ on the learning task with respect to the example space $\mathcal{X}_v \times \mathcal{Y}$ and distribution $\mathbb{D}$ is defined as

$$\Upsilon(\mathcal{X}_v, \mathcal{Y}, \mathbb{D}) = 1 - \int_{x \in \mathbb{D}_{\mathcal{X}_v}} |2P(y = 1|x) - 1|\, P(x)\mathrm{d}x.$$

The view insufficiency can be interpreted as averaging the quantity $1 - |2P(y = 1|x_v) - 1|$ over the distribution $\mathbb{D}_{\mathcal{X}_v}$. Usually it is hard to calculate the view insufficiency directly, but Wang and Zhou [21] gave the following proposition to state that the view insufficiency is related with the error rate of the optimal classifier:

Proposition 1

$$\Upsilon(\mathcal{X}_v, \mathcal{Y}, \mathbb{D}) = 2P(\mathrm{sign}(f_v^*(x_v)) \neq y)).$$

From Proposition 1 we can see that if the view is very insufficient, the optimal classifier f^* has high error rate.

3 Weighted co-training

In original co-training [8], two classifiers are first trained with the initial labeled data. Then, each classifier selects and labels some high-confident unlabeled data for its peer. After that, each classifier is updated with the newly labeled data provided by its peer. The whole process repeats until no classifier changes or reaching a pre-set number of rounds. The detailed description is in Algorithm 1.

Algorithm 1 Co-training

```
  Input: Labeled data $L$, Unlabeled data $U$, number of instances to be
  labeled for positive/negative class during each iteration $(p, n)$, number
  of iterations $T$
  Output: Trained classifiers $(f_1^T, f_2^T)$
1 Create a pool $U'$ by randomly sample some examples from $U$;
2 for $t = 0$ to $T$ do
3     Train $f_1^t, f_2^t$ on $L$;
4     Let $f_1^t$ label its most confident $p + n$ instances from $U'$;
5     Let $f_2^t$ label its most confident $p + n$ instances from $U'$;
6     Add these newly labeled instance to $L$;
7     Replenish $U'$ with another $2p+2n$ instances randomly selected from
      $U$;
8 end
```

When both views are sufficient, there exist theoretical supports guaranteeing co-training's performance. But in practice the views are usually insufficient, as stated by Wang and Zhou [21], those performance guarantees are no longer satisfied. They gave an error bound for the output classifier, which is presented in Proposition 2:

Proposition 2 (Theorem 9 in [21]) When the two views are diverse, for $(\langle x_1, x_2\rangle, y) \sim \mathbb{D}$, let $\eta_v = P(f_v^*(x_v) \neq y)$, $(v = 1, 2)$ denote the error rate of the optimal classifier f_v^*; Then for any $\epsilon \in (0, 1/2)$ and $\delta \in (0, 1)$, with probability $1 - \delta$ the output classifiers f_1^T and f_2^T in Algorithm 1 satisfy

$$R(f_v^T) \leqslant \frac{\eta_1 + \eta_2 + P(\mathrm{sign}(f_1^*(x_1)) \neq \mathrm{sign}(f_2^*(x_2)))}{2} + \epsilon.$$

When the two views are insufficient, their optimal classifiers are not compatible, i.e., $P(\mathrm{sign}(f_1^*(X_1)) \neq \mathrm{sign}(f_2^*(X_2))) > 0$. Proposition 2 indicates that some labeled examples provided by one view may be harmful for learning the optimal classifier in the other view. In fact, these harmful examples are exactly the ones *inconsistent* with f_v^* (here "inconsistent" means their labels are different from the predictions of f_v^*). Hence, if we can reduce the effect of these harmful examples, the performance of the algorithm will be improved.

To address the problem, we propose the *Weighted Co-training* algorithm. In each iteration, the algorithm adjusts the weights of the examples in the training set based on the outputs of the updated classifier, so that the potential inconsistent examples' weights will be decreased. In the following part, we focus on one of the two views. For the sake of convenience, we use f to denote the classifier and use $(x_i, \widehat{y_i})$ to denote the i-th example with its pseudo label provided by the other view. Let N^t denote the newly labeled data provided by the other view in iteration t, and $N = |N^t| = p + n$ denote the size of N^t. In iteration t, the updated classifier f is obtained by minimizing the empirical risk:

$$f = \mathop{\arg\min}_{f \in \mathcal{F}} \sum_{i=1}^{l+N\cdot(t+1)} \mathbb{I}(\mathrm{sign}(f(x_i)) \neq \widehat{y_i}).$$

As discussed above, some newly labeled examples are inconsistent with the optimal classifier. To reduce the effect in the learning process, we will decrease their weights in the training set. Ideally, the weight vector $\mathbf{w}^{t+1}$ should be in the form of

$$w_i^{t+1} = \begin{cases} 0, & \text{if } \mathrm{sign}(f^*(x_i)) \neq \widehat{y_i}, \\ \text{constant}, & \text{otherwise}. \end{cases}.$$

However, it is hard to know $\mathbf{w}^{t+1}$ since the optimal classifier f^* is difficult to achieve. Generally, the optimal $\mathbf{w}^{t+1}$ should satisfy the optimization in Eq. (1):

$$\mathbf{w}^{t+1} = \mathop{\arg\min}_{\substack{\mathbf{w}^{t+1} \in \mathbb{R}^{l+N\cdot(t+1)} \\ \|\mathbf{w}^{t+1}\|=1}} \sum_{i=1}^{l+N\cdot(t+1)} w_i^{t+1} \mathbb{I}(\mathrm{sign}(f^*(x_i)) \neq \widehat{y_i}). \tag{1}$$

Since the optimal classifier f^* is unknown, we use f^{t+1} to approximate f^* and get Eq. (2) from Eq. (1):

$$\mathbf{w}^{t+1} = \mathop{\arg\min}_{\substack{\mathbf{w}^{t+1} \in \mathbb{R}^{l+N\cdot(t+1)} \\ \|\mathbf{w}^{t+1}\|=1}} \sum_{i=1}^{l+N\cdot(t+1)} w_i^{t+1} \mathbb{I}(\mathrm{sign}(f^{t+1}(x_i)) \neq \widehat{y_i}). \tag{2}$$

Solving Eq. (2) directly would make the weight concentrate on a few examples with small loss while assigning zero weight to the rest. To avoid such risk and make the weight spread on more examples, we require that the entropy of $\mathbf{w}^{t+1}$ should be lower-bounded by some constant:

$$-\sum_{i=1}^{l+N\cdot(t+1)} w_i^{t+1} \log(w_i^{t+1}) > \xi, \text{ for some constant } \xi > 0.$$

In the learning process, the loss should keep decreasing, i.e.,

$$\sum_{i=1}^{l+N\cdot(t+1)} w_i^{t+1} \mathbb{I}(\mathrm{sign}(f^{t+1}(x_i)) \neq \widehat{y_i}) \leqslant \sum_{i=1}^{l+N\cdot t} w_i^t \mathbb{I}(\mathrm{sign}(f^t(x_i)) \neq \widehat{y_i}).$$

To summarize the whole process: in iteration t, train f^{t+1} with $(L^t \cup N^t, \mathbf{w}^t)$ and solve the optimization in Eq. (3) to obtain $\mathbf{w}^{t+1}$:

$$\begin{aligned} \mathbf{w}^{t+1} = & \mathop{\arg\min}_{\mathbf{w}^{t+1} \in \mathbb{R}^{l+N\cdot(t+1)}} \sum_{i=1}^{l+N\cdot(t+1)} w_i^{t+1} \mathcal{L}(f^{t+1}(x_i), \widehat{y_i}), \\ \text{s.t.} \quad & \sum_{i=1}^{l+N\cdot(t+1)} w_i^{t+1} \mathcal{L}(f^{t+1}(x_i), \widehat{y_i}) \leqslant \sum_{i=1}^{l+Nt} w_i^t \mathcal{L}(f^t(x_i), \widehat{y_i}), \\ & -\sum_{i=1}^{l+N\cdot(t+1)} w_i^{t+1} \log(w_i^{t+1}) \geqslant \xi \quad \text{for some } \xi > 0, \\ & \|\mathbf{w}^{t+1}\| = \sum_{i=1}^{l+N\cdot(t+1)} w_i^{t+1} = 1, \ \mathbf{w}^{t+1} \geqslant 0. \end{aligned} \tag{3}$$

Here $\mathcal{L}(\cdot, \cdot)$ represents the surrogate loss function. When we choose $\mathcal{L}$ to be hinge loss (i.e., $\mathcal{L}(f(x), y) = \max(0, -f(x)y)$), the optimization in Eq. (3) can be solved by standard Lagrange multipliers method and the optimal $\mathbf{w}^{t+1}$ has the form of:

$$w_i^{t+1} = \begin{cases} C_i \exp(-|f^{t+1}(x_i)|) & \text{if } \mathrm{sign}(f^{t+1}(x_i)) \neq \widehat{y_i}, \\ C_0 & \text{otherwise}. \end{cases} \tag{4}$$

where C_i, C_0 are some constants relevant to the parameter ξ. For the examples that are inconsistent with f^{t+1}, their weight are reduced by multiplying a factor of $\exp(-|f^{t+1}(x_i)|)$, hence an example's weight will shrink more if it is wrongly classified with higher confidence. The algorithm is depicted in Algorithm 2.

Algorithm 2 Weighted Co-training

```
Input: Labeled data $L = (L_1, L_2)$, Unlabeled data $U = (U_1, U_2)$, number of instances to be labeled for positive/negative class during each iteration $(p, n)$, weight shrinkage parameter $\alpha$
Output: Trained classifiers $(f_1^T, f_2^T)$
1  $\left(L_1^0, L_2^0\right) \leftarrow (L_1, L_2)$;
2  Initialize classifiers $f_1^0, f_2^0$ on $L_1^0, L_2^0$ respectively;
3  Create a pool $U'$ by randomly sample some examples from $U$;
4  foreach $(x_1, x_2)$ in $(L_1^0, L_2^0)$ do
5      $(\mathbf{w}(x_1), \mathbf{w}(x_2)) \leftarrow (1.0, 1.0)$
6  end
7  for $t = 0$ to $T$ do
8      for $v \in \{1, 2\}$ do
9          Let $f_v^t$ label its most confident $p + n$ instances from $U'$ as $N_v^t$;
10         Replenish $U'$ with another $p+n$ instances randomly selected from $U$;
11         $L_{3-v}^{t+1} \leftarrow L_{3-v}^t \cup N_v^t$;
12         Retrain the classifiers on $L_{3-v}^{t+1}$ with current data weight to get $f_{3-v}^{t+1}$;
           // Deal with inconsistent data
13         $D_{3-v}^{t+1} \leftarrow \left\{(x, y) | (x, \hat{y}) \in L_{3-v}^{t+1},\ \mathrm{sign}(f_{3-v}^{t+1}(x)) \neq \hat{y}\right\}$;
14         foreach $x \in D_{3-v}^{t+1}$ do
15             $\mathbf{w}(x) = \mathbf{w}(x) * \exp(-\alpha * |f_{3-v}^{t+1}(x)|)$
16         end
17         foreach $x \in L_{3-v}^{t+1} \setminus D_{3-v}^{t+1}$ do
18             $\mathbf{w}(x) = 1.0$
19         end
20     end
21 end
```

4 Experiments

In order to study whether our algorithm could have better performance, we conduct empirically studies on several real-world data sets. In the experiments we use several classifiers as base learners, including naive bayes (NB, based on multinomial distribution), support vector machine (SVM, with linear kernel), logistic regression (LR), and decision tree (TREE). We use the original co-training in [8] and DCPE co-training in [22] as the baselines. In each iteration of DCPE co-training, it encourages the two classifiers to label the examples on which they make the same prediction but have diverse confidence.

4.1 Results on two-view data sets

For two-view setting, we use data sets *courses* [8], *ads* [23], and *citeseer* [24]. These data sets are often used for multi-view learning in previous works [8, 19, 20]. The *courses* data set has two views: a *pages* view (the Web page's text content) and a *links* view (the hyper-link pointing to the page). The data set contains 1,051 instances, of which 230 are positive and 821 are negative. The *ads* data set has five views and we follow the setting in [20] to use the 1st and the 3rd views. It consists of 138 positive and 845 negative instances. The *citeseer* data set has four views: *content, inbound, outbound*, and *cites*. Following the setting in [25] we use the *content* and *cites* views in our experiment. The *citeseer* contains six different classes, in this paper we focus on binary classification, and select classes 3 and 4 to serve as positive and negative classes, which have 681 positive and 666 negative instances.

We first estimate the view *insufficiency* of the data sets. By proposition 1, the insufficiency Υ can be calculated with respect to the error rate of the optimal classifier:

$$\Upsilon = 2 \times \mathrm{Err}(f^*).$$

The optimal classifier is difficult to achieve, we approximate it by using the classifier trained on all labeled data. We gradually increase the number of labeled data until the trained classifier's error rate converges, and the converged error rate is used as $\mathrm{Err}(f^*)$. The insufficiency is listed in Table 1, from which we can see that all three data sets are insufficient.

Table 1 Empirical view insufficiency on two-view data

Configuration	Insufficiency Υ	
	View 1	View 2
courses-NB-NB	0.150	0.120
ads-SVM-SVM	0.146	0.108
citeseer-NB-NB	0.274	0.343

Note: Configuration "*data-A-B*" means "on data set *data*, train classifier *A* on view 1 and *B* on view 2".

For each data set, we randomly select 25% data as the test set, and the remaining 75% as the training set. A small fraction of the training set is selected as the initial labeled data (the concrete size will be described later) while the rest are the unlabeled data. Each experiment is repeated for 40 times and the average result is reported.

The average test error rates are shown in Fig. 1. Each image is split vertically into two parts, depicting the results in the two views respectively. It can be observed that our algorithm outperforms the original co-training and DCPE co-training on data sets *courses* and *ads*. On the first view of the *citeseer* data set, the performance of our algorithm is a little worse than DCPE co-training. Nevertheless, DCPE co-training performs badly on the *ads* data set and stops early since it cannot find the required examples in the iterative process.

4.2 Results on single-view data sets

As mentioned in Section 1, co-training also works on single-view data. We still use "view 1" and "view 2" to refer to the two classifiers that are actually trained on the same feature

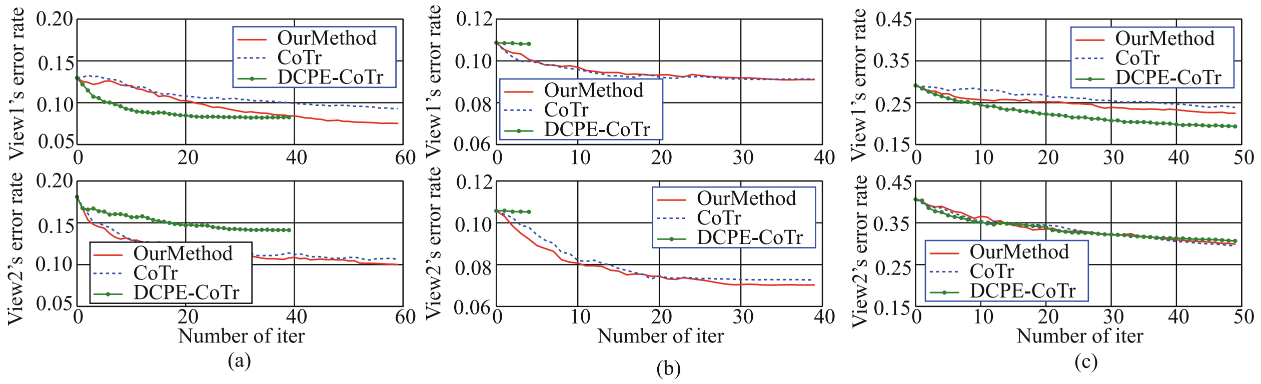

Fig. 1 Test error vs. iteration numbers on two-view data sets. Figure name "*data-A-B-c-d-p-n*" means "on data set *data*, train classifier *A* on view 1 and *B* on view 2 with *c* positive and *d* negative initial labeled data instances, then label *p* positive data and *n* negative ones on each view for every iteration". For example, "*courses*-NB-NB-3-9-1-3" shows result on data set *courses*, on which NB (Naive Bayes classifier) is used on both view 1 and view 2; initial labeled data consist of 3 positive ones and 9 negative ones; and the algorithm labels 1 positive and 3 negative unlabeled data every iteration for each view. (a) *courses*-NB-NB-3-9-1-3; (b) *ads*-SVM-SVM-4-24-1-6; (c)*citeseer*-NB-NB-8-8-1-1

set. We select several data sets from the UCI data repository [26], including *pendigits, nursery, mushroom, tic-tac-toe, colic* and *vehicle*. Among them, the data sets *mushroom*, *tic-tac-toe*, and *colic* consist of binary classes: *mushroom*, 418 positive and 818 negative instances; *tic-tac-toe*, 626 positive and 332 negative instances; and *colic*, 232 positive and 136 negative instances. The remaining three data sets have more than 2 classes, we choose the two largest classes for binary classification: *nursery*, class 1 and 0, with size 4266 and 4320; *pendigits*, class 3 and 5, with size 1055 and 1055. The size of each class in *vehicle* is too small, we group classes 0 and 3 as negative class while the rest as positive class.

We examine the view insufficiency of these data sets in Table 2 and present the test error rates in Fig. 2. On all but the *nursery* data set, our algorithm outperforms the original co-training. On the *nursery* data set, our algorithm has similar performance as the original co-training, since the view insufficiency of this data set is zero. Our algorithm also significantly outperforms DCPE co-training on 4 data sets, i.e., *pendigits*, *nursery*, *mushroom* and *vehicle*. On *tic-tac-toe* and *colic* data sets, Our algorithm also has comparable performance with DCPE co-training.

Table 2 Empirical view insufficiency on single-view data

Configuration	Insufficiency Υ	
	View 1	View 2
pendigits-NB-TREE	0.020	0.000
nursery-SVM-LR	0.000	0.000
mushroom-NB-SVM	0.158	0.031
tic-tac-toe-TREE-SVM	0.369	0.033
colic-LR-TREE	0.321	0.516
vehicle-LR-TREE	0.106	0.541

Note: configuration "*data-A-B*" means "on data set *data*, train classifier *A* on view 1 and *B* on view 2".

4.3 Further experiments

In this section, we will examine whether our algorithm could identify the inconsistent examples in the learning process. We use the classifier trained on all examples to approximate the optimal classifier, and compare its prediction with the *pseudo labels* in the iterative process. Recall that in Algorithm 2, D^{t+1} denotes the set of potential inconsistent examples returned by our algorithm (line 13 in Algorithm 2) in the *t*th iteration, we denote the set of all potential inconsistent examples as $D = \bigcup_{t=1}^{T} D^t$, and the *inconsistency* $\mathcal{I}$ between the optimal classifier f^* and the *pseudo labels* of these examples can be measured as:

$$\mathcal{I}(D) = \frac{1}{|D|} \sum_{x_i \in D} \mathbb{I}(f^*(x_i) \neq \widehat{y_i}).$$

The inconsistency $\mathcal{I}$ on the identified examples of our algorithm is shown in Table 3 and Table 4 for the two-view and the single-view data, respectively. From Table 3 and Table 4 we can see that on all data sets, the optimal classifiers have high inconsistency with the pseudo labels on the identified examples, i.e., these examples are harmful for learning the optimal classifiers, and their weights should be decreased dramatically in the training set. On View 1 of *tic-tac-toe* and View 2 of *nursery, mushroom, pendigits, colic*, our algorithm

Table 3 The inconsistency between the optimal classifier and the pseudo labels on the identified instances, for two-view data sets

Configuration	Inconsistency $\mathcal{I}$	
	View 1	View 2
courses-NB-NB	1.0	0.923
ads-SVM-SVM	1.0	0.978
citeseer-NB-NB	1.0	0.833

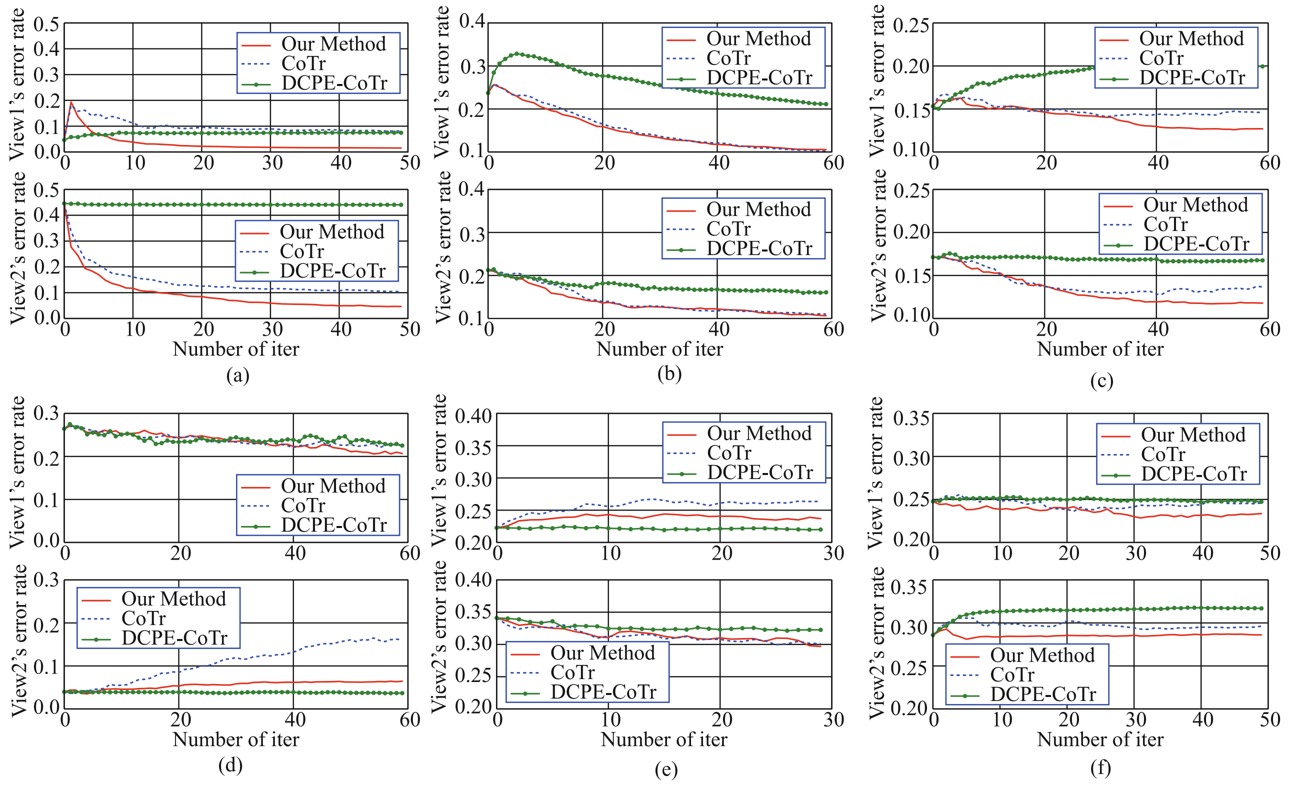

Fig. 2 Test error vs. iteration numbers on single-view data sets. (a) *pendigits*-NB-TREE-1-1-1-1; (b) *nursery*-NB-SVM-3-3-1-1; (c) *mushroom*-NB-SVM-4-4-1-1; (d) *tic-tac-toe*-TREE-SVM-55-55-1-1; (e) *colic*-LR-TREE-20-30-1-2; (f) *vehicle*-SVM-NB-20-30-1-2

Table 4 The inconsistency between the optimal classifier and the pseudo labels on the identified instances, for single-view data sets

Configuration	Inconsistency $\mathcal{I}$	
	View 1	View 2
pendigits-NB-TREE	1.0	-
nursery-NB-SVM	0.8	-
mushroom-NB-SVM	0.667	-
tic-tac-toe-TREE-SVM	-	0.975
colic-LR-TREE	0.84	-
vehicle-SVM-NB	1.0	0.92

Note: "-" means no inconsistent data are detected

does not identify any inconsistent examples, the corresponding classifier's performance on these data sets is similar as the original co-training (Fig. 2).

5 Conclusion

In this paper, we propose a novel algorithm named Weighted Co-training to make co-training suffer less from insufficient views. Under insufficient views the newly labeled data may be inconsistent with the optimal classifiers. Our algorithm achieves better accuracy and robustness by identifying probably inconsistent examples and decreasing their weights in the training set. We regard this paper as a preliminary work towards safe co-training which can utilize unlabeled data to help the learning process without risk of performance degradation, and we expect that more researchers will follow this direction.

Acknowledgements This work was supported by the NSFC (61673202, 61305067), the Fundamental Research Funds for the Central Universities, and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

1. Miller D J, Uyar H S. A mixture of experts classifier with learning based on both labelled and unlabelled data. Advances in Neural Information Processing Systems, 1997, 571–577
2. Nigam K, McCallum A, Thrun S, Mitchell T. Text classification from labeled and unlabeled documents using EM. Machine Learning, 2000, 39(2/3): 103–134
3. Bennett K P, Demiriz A. Semi-supervised support vector machines. Advances in Neural Information Processing Systems, 1998, 368–374
4. Joachims T. Transductive inference for text classification using support vector machines. In: Proceedings of the 16th International Conference on Machine Learning. 1999, 200–209
5. Blum A, Chawla S. Learning from labeled and unlabeled data using graph mincuts. In: Proceedings of the 18th International Conference on Machine Learning. 2001, 19–26
6. Zhu X, Ghahramani Z, Lafferty J. Semi-supervised learning using

gaussian fields and harmonic functions. In: Proceedings of the 20th International Conference on Machine Learning. 2003, 912–919

7. Zhou D, Bousquet O, Lal T N, Weston J, Schölkopf B. Learning with local and global consistency. Advances in Neural Information Processing Systems, 2003, 321–328
8. Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory. 1998, 92–100
9. Zhou Z H, Li M. Tri-training: exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(11): 1529–1541
10. Zhou Z H, Li M. Semi-supervised learning by disagreement. Knowledge and Information System, 2010, 24(3): 415–439
11. Nigam K, Ghani R. Analyzing the effectiveness and applicability of co-training. In: Proceedings of the 10th International Conference on Information and Knowledge Management. 2000, 86–93
12. Goldman S A, Zhou Y. Enhancing supervised learning with unlabeled data. In: Proceedings of the 17th International Conference on Machine Learning. 2000, 327–334
13. Kiritchenko S, Matwin S. Email classification with co-training. In: Proceedings of the 2001 Conference of the Centre for Advanced Studies on Collaborative Research. 2001, 301–312
14. Maeireizo B, Litman D, Hwa R. Co-training for predicting emotions with spoken dialogue data. In: Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions. 2004, 28
15. Wan X. Co-training for cross-lingual sentiment classification. In: Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP. 2009, 235–243
16. Liu R, Cheng J, Lu H. A robust boosting tracker with minimum error bound in a co-training framework. In: Proceedings of the 12th IEEE International Conference on Computer Vision. 2009, 1459–1466
17. Abney S P. Bootstrapping. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. 2002, 360–367
18. Balcan M F, Blum A, Yang K. Co-training and expansion: towards bridging theory and practice. Advances in Neural Information Processing Systems, 2004, 89–96
19. Wang W, Zhou Z H. A new analysis of co-training. In: Proceedings of the 27th International Conference on Machine Learning. 2010, 1135–1142
20. Wang W, Zhou Z H. Analyzing co-training style algorithms. In: Proceedings of the 18th European Conference on Machine Learning. 2007, 454–465
21. Wang W, Zhou Z H. Co-training with insufficient views. In: Proceedings of the 5th Asian Conference on Machine Learning. 2013, 467–482
22. Xu J, He H, Man H. DCPE co-training for classification. Neurocomputing, 2012, 86: 75–85
23. Kushmerick N. Learning to remove internet advertisements. In: Proceedings of the 3rd Annual Conference on Autonomous Agents. 1999, 175–181
24. Giles C L, Bollacker K D, Lawrence S. Citeseer: an automatic citation indexing system. In: Proceedings of the 3rd ACM International Conference on Digital Libraries. 1998, 89–98
25. Bisson G, Grimal C. Co-clustering of multi-view datasets: a parallelizable approach. In: Proceedings of the 12th IEEE International Conference on Data Mining. 2012, 828–833
26. Lichman M. UCI machine learning repository. 2013

Xiangyu Guo received his BS in Electronic Engineering from Xidian University, China in 2014. He received the National Scholarship in 2011. Currently he is a master student at the Department of Computer Science and Technology, Nanjing University, China. His research interests include machine learning and data mining.

Wei Wang is an associate professor at Department of Computer Science and Technology, Nanjing University, China. He received his PhD degree from Department of Computer Science and Technology, Nanjing University, China in 2012. His research interest mainly includes computational learning theory, especially in semi-supervised learning and active learning.