Analyzing Co-Training Style Algorithms

Wei Wang and Zhi-Hua Zhou

National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, China
{wangw, zhouzh}@lamda.nju.edu.cn

Abstract. Co-training is a semi-supervised learning paradigm which trains two learners respectively from two different views and lets the learners label some unlabeled examples for each other. In this paper, we present a new PAC analysis on co-training style algorithms. We show that the co-training process can succeed even without two views, given that the two learners have large difference, which explains the success of some co-training style algorithms that do not require two views. Moreover, we theoretically explain that why the co-training process could not improve the performance further after a number of rounds, and present a rough estimation on the appropriate round to terminate co-training to avoid some wasteful learning rounds.

1 Introduction

Unlabeled training data are usually much easier than labeled training data to be obtained in many practical machine learning applications, so semi-supervised learning which attempts to exploit unlabeled data to help improve the performance of learning with limited labeled training data has attracted much attention during the past few years [5, 9, 13, 6, 12, 10, 17, 3]. *Co-training* is a well-known semi-supervised learning paradigm. In its initial form [5], co-training trains two classifiers separately on two *sufficient and redundant views*, i.e., two attribute sets each of which is sufficient for learning and conditionally independent to the other given the class label, and lets the two classifiers label some unlabeled instances for each other. Since in most real-world scenarios sufficient and redundant views do not exist, variants of co-training that do not require two views have been developed. For example, rather than using two views, Goldman and Zhou [8] used two different supervised learning algorithms, Zhou and Li [16] used two different parameter configurations of the same base learner, etc.

There are several theoretical studies on co-training. Dasgupta et al. [7] proved that when the requirement of sufficient and redundant views is met, the co-trained classifiers could make fewer generalization errors by maximizing their agreement over the unlabeled data. Balcan et al. [2] showed that given appropriately strong PAC-learners on each view, an assumption of *expansion* on the underlying data distribution, which is weaker than the assumption of sufficient and redundant views, is sufficient for *iterative co-training* to succeed. This tells that the *conditional independence* [5] or even the *weak dependence* [1] between

the two views is not needed, at least, for iterative co-training which is the popular routine taken by many variants of co-training.

Previous theoretical studies mainly investigate co-training with two views. Although the applicability of co-training style algorithms that do not require two views is better in practice and empirical studies have shown the effectiveness of those algorithms, there is no theoretical analysis that can explain that why co-training without two views can succeed. On the other hand, in experiments we have observed that the co-training process could not improve the learning performance further after a number of rounds, which has not been analyzed by previous theoretical studies.

In this paper, we present a new PAC analysis which addresses the above issues. In detail, we derive a theorem on why co-training can work without two views, and a theorem on why co-training could not improve the performance after a number of learning rounds. The second theorem provides a rough estimation on the appropriate round to terminate the co-training process to avoid some wasteful learning rounds, which is validated by an empirical study in this paper.

The rest of this paper is organized as follows. After stating some preliminaries in Section 2, we present our theoretical results in Section 3, then report on our empirical study on determining the appropriate round to terminate co-training in Section 4, and finally conclude the paper in Section 5.

2 Preliminaries

Given data set $\mathcal{S} = \mathcal{L} \cup \mathcal{U}$, where $\mathcal{L} = \{(x_1, y_1), \cdots, (x_l, y_l)\} \subset \mathcal{X} \times \mathcal{Y}$ is the labeled data set and $\mathcal{U} = \{(x_{l+1}, x_{l+2}, \cdots, x_n)\} \subset \mathcal{X}$ is the unlabeled data set. $\mathcal{Y} = \{-1, +1\}$; $\mathcal{X}$ is with distribution $\mathcal{D}$. Let $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ denote the hypothesis space. Assume that $|\mathcal{H}|$ is finite, and $\mathcal{D}$ is generated by the ground truth $h^* \in \mathcal{H}$. It is obvious that the generalization error of h^* is zero. Since we have only finite sample, it is hard to achieve h^* over $\mathcal{S}$. Suppose we obtain a classifier $h^i \in \mathcal{H}$ from $\mathcal{S}$, which is somewhat different from h^*. Let $d(h^i, h^*)$ denote the difference between the two classifiers h^i and h^*, then

$$d(h^i, h^*) = Pr_{x \in \mathcal{D}}[h^i(x) \neq h^*(x)]. \tag{1}$$

Let ϵ bound the generalization error of the classifiers what we wish to achieve finally. That is, if $d(h^i, h^*) = Pr_{x \in \mathcal{D}}[h^i(x) \neq h^*(x)] < \epsilon$, we say that we have obtained a desired classifier since the difference between this classifier and the ground truth h^* is very small; otherwise we say that the classifier h^i is 'bad'. Of course we wish to have a high probability to achieve a good classifier. The confidence parameter δ can play this role. The learning process is said to do probably approximately correct learning of h^* if and only if $Pr[d(h^i, h^*) \geq \epsilon] \leq \delta$, where the probability is taken over all possible training data. Formally, the requirement is that the difference between the ground truth h^* and the hypothesis h^i be small (less than ϵ) with high probability (more than $1 - \delta$).

3 Main Results

Given the initial labeled data $\mathcal{L}$ and unlabeled data $\mathcal{U}$, consider the following co-training learning process:

Co-Training Process: *At first, two initial classifiers h_1^0 and h_2^0 are trained using $\mathcal{L}$ which contains l labeled examples. Then, h_1^0 selects u number of unlabeled instances from $\mathcal{U}$ to label, and puts these newly labeled examples into the data set σ_2 which contains all the examples in $\mathcal{L}$; at the same time, h_2^0 selects u number of unlabeled instances from $\mathcal{U}$ to label, and puts these newly labeled examples into the data set σ_1 which contains all the examples in $\mathcal{L}$. h_1^1 and h_2^1 are trained from σ_1 and σ_2, respectively. After that, h_1^1 selects u number of unlabeled instances to label, and uses these newly labeled examples to update σ_2; while h_2^1 also selects u number of unlabeled instances to label, and uses these newly labeled examples to update σ_1. Such a process is repeated for a pre-set number of learning rounds.*

Different learners have different biases, which is an intuitive explanation to that why co-training style algorithms can succeed. The two classifiers that have different biases will label some instances with different labels. The difference $d(h^i, h^j)$ between the two classifiers h^i and h^j implies the different biases between them. If the examples labeled by the classifier h^i is useful for the classifier h^j, h^i should know some information that h^j does not know. In other words, h^i and h^j should have large difference. As the Co-Training Process proceeds, the two classifiers will become more and more similar and the difference between them will become smaller and smaller since the two classifiers label more and more unlabeled instances for each other. The difference can be helpful for analyzing the co-training style algorithms.

In the i-th learning round, let a_i and b_i denote the upper bound of the generalization error of h_1^i and h_2^i, respectively; let $d(h_1^{i-1}, h_2^i)$ denote the difference between h_1^{i-1} and h_2^i, and let $d(h_1^i, h_2^{i-1})$ denote the difference between h_1^i and h_2^{i-1}. It is feasible to estimate the difference when there are a large mount of unlabeled instances. Now we present our main result.

Theorem 1. *Given the initial labeled data set $\mathcal{L}$, assuming that the size of $\mathcal{L}$ is sufficient to learn two classifiers h_1^0 and h_2^0 whose upper bound of the generalization error is $a_0 < 0.5$ and $b_0 < 0.5$ respectively with high probability (more than $1-\delta$) in the PAC model, i.e., $l \geq max\{\frac{1}{a_0} ln \frac{|\mathcal{H}|}{\delta}, \frac{1}{b_0} ln \frac{|\mathcal{H}|}{\delta}\}$. Then h_1^0 selects u number of unlabeled instances from $\mathcal{U}$ to label and puts them into σ_2 which contains all the examples in $\mathcal{L}$, and then h_2^1 is trained from σ_2 by minimizing the empirical risk. If $lb_0 \leq e \sqrt[M]{M!} - M$, then*

$$Pr[d(h_2^1, h_*) \geq b_1] \leq \delta. \tag{2}$$

where $M = ua_0$ and $b_1 = \max\{\frac{lb_0 + ua_0 - ud(h_1^0, h_2^1)}{l}, 0\}$.

Proof. Firstly, we analyze the expected rate of disagreement between the classifier h^i and the sample sequence σ_2 which consists of u number of newly labeled examples and $\mathcal{L}$. By minimizing the empirical risk, the classifier which

has the lowest observed rate of disagreement with the sample sequence σ_2 will be generated. Let $d(h^i, \sigma_2)$ denote the expected rate of disagreement between h^i and σ_2. Then

$$d(h^*, \sigma_2) = \frac{u \times d(h_1^0, h^*)}{l+u} \tag{3}$$

$$d(h_2^1, \sigma_2) = \frac{l \times d(h_2^1, h^*) + u \times d(h_1^0, h_2^1)}{l+u} \tag{4}$$

In order to achieve 'good' classifiers whose generalization errors are less than b_1 by minimizing the empirical risk, the sample sequence σ_2 must be sufficient to guarantee that no classifier whose generalization error is no smaller than b_1 has a lower observed rate of disagreement with σ_2 than h^* with a probability bigger than $1-\delta$.

Since the upper bound of the generalization error of the classifier h_1^0 is a_0, $d(h^*, \sigma_2)$ is no more than $\frac{ua_0}{l+u}$. Let $M = ua_0$, the probability that the classifier h_2^1 has a lower observed rate of disagreement with σ_2 than h^* is less than

$$C_{l+u}^{M} d(h_2^1, \sigma_2)^M [1 - d(h_2^1, \sigma_2)]^{l+u-M}. \tag{5}$$

Let $b_1 = \max\{\frac{lb_0 + ua_0 - ud(h_1^0, h_2^1)}{l}, 0\}$, if $d(h_2^1, h^*) \geq b_1$,

$$\begin{aligned} d(h_2^1, \sigma_2) &= \frac{l \times d(h_2^1, h^*) + u \times d(h_1^0, h_2^1)}{l+u} \\ &\geq \frac{lb_1 + u \times d(h_1^0, h_2^1)}{l+u} \\ &\geq \frac{lb_0 + ua_0}{l+u} \\ &> \frac{M}{l+u}. \end{aligned}$$

The function $C_m^s x^s (1-x)^{m-s}$ is monotonically decreasing as x increases when $s/m < x < 1$. So, if $d(h_2^1, h^*) \geq b_1$, the value of Eq.5 is smaller than

$$C_{l+u}^{M} (\frac{lb_0 + ua_0}{l+u})^M (1 - \frac{lb_0 + ua_0}{l+u})^{l+u-M}. \tag{6}$$

In other words, the probability for that the classifier with generalization error no less than b_1 has a lower observed rate of disagreement with σ_2 than h^* is smaller than the value of Eq.6.

The calculation of the real value of Eq.6 is quite complex, so we approximate it by using the Poisson Theorem:

$$C_{l+u}^{M} (\frac{lb_0 + ua_0}{l+u})^M (1 - \frac{lb_0 + ua_0}{l+u})^{l+u-M} \approx \frac{(lb_0 + ua_0)^M}{M!} e^{-(lb_0 + ua_0)} \tag{7}$$

When $lb_0 \leq e \sqrt[M]{M!} - M$, the right-hand term of Eq.7 is no more than e^{-lb_0}. Since the classifier h_2^0 is PAC-learnable and the sample size of $\mathcal{L}$ is at least $\frac{1}{b_0} ln \frac{|\mathcal{H}|}{\delta}$,

$e^{-lb_0} \leq \delta/|\mathcal{H}|$. Therefore, the value of Eq.6 is no more than $\delta/|\mathcal{H}|$. Considering that there are at most $|\mathcal{H}| - 1$ classifiers with generalization error no less than b_1 having a lower observed rate of disagreement with σ_2 than h^* in $\mathcal{H}$, the probability that $Pr[d(h_2^1, h^*) \geq b_1]$ is at most δ. □

Theorem 1 shows that given the initial labeled data, if we can train two learners which have large difference, the learners can be improved by exploiting the unlabeled data through the Co-Training Process. It is easy to see that the Co-Training Process reassembles the main process of existing co-training style algorithms [5, 8, 16]. It can be recognized that the two views used in the standard co-training algorithm [5], the two different supervised learning algorithms used in Goldman and Zhou's algorithm [8], and the different configurations of the base learner used in Zhou and Li's algorithm [16] are actually exploited to make the classifiers to have large difference. This explains why co-training without two views [8, 16] can succeed.

When co-training style algorithms are executed, the number of labeled examples is usually small while the initial classifiers are usually not very bad, thus the condition that $lb_0 \leq e \sqrt[M]{M!} - M$ in Theorem 1 can be satisfied. Note that using a bigger u will increase the upper bound of lb_0. This is because that a bigger u will result in a bigger M since $M = ua_0$, while Fig. 1 shows that the value of the function $f(M) = e \sqrt[M]{M!} - M$ increases as M increases.

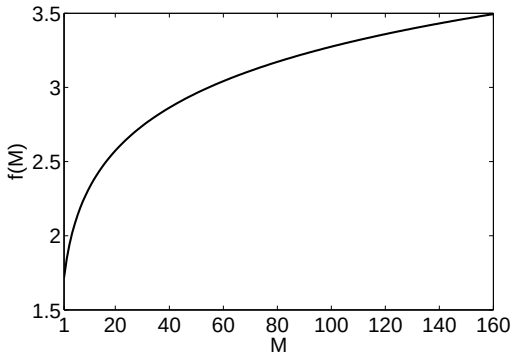

Fig. 1. The value of $f(M) = e \sqrt[M]{M!} - M$

In Theorem 1 we know that when the difference $d(h_1^0, h_2^1)$ is bigger than a_0, the upper bound $b_1 = \max\{\frac{lb_0 + ua_0 - ud(h_1^0, h_2^1)}{l}, 0\}$ is smaller than b_0. The bigger the difference $d(h_1^0, h_2^1)$, the smaller the upper bound of the generalization error of the classifiers h_2^1. In the Co-Training Process, the difference between the two learners decreases as the value of u increases. When u increases to a certain degree, the difference between the two learners becomes very small. This is easy

to understand since the learner h_2^1 is trained by minimizing the empirical risk with a large number of examples provided by h_1^0.

From the above we know that when $d(h_1^0, h_2^1)$ is larger than a_0, we can generate 'good' classifiers according to Theorem 1. But when $d(h_1^0, h_2^1)$ is smaller than a_0, what is the performance of the Co-Training Process? In this case, if $d(h_2^1, \sigma_2)$ is bigger than $d(h^*, \sigma_2)$ (for any $d(h_2^1, h^*) \geq b_1$), we can still obtain 'good' classifiers ($d(h_2^1, h^*) < b_1$) by minimizing the empirical risk because the expected rate of disagreement between the 'bad' classifiers and sample sequence is bigger than that between the ground truth h^* and sample sequence. So, with the Co-Training Process we can obtain classifiers which satisfy $d(h_2^1, h^*) < b_1$ with big probability. It is requested that u should be smaller than $lb_1/[a_0 - d(h_1^0, h_2^1)]$ for searching the 'good' classifiers. Thus, if $d(h_1^0, h_2^1)$ is smaller than a_0, the Co-Training Process will work well when $u < lb_1/[a_0 - d(h_1^0, h_2^1)]$. For $u \gg l$, we have Theorem 2:

Theorem 2. *In the Co-Training Process, if $u \gg l$, then for any $0 < \epsilon < 1$,*

$$Pr[d(h_1^0, h_2^1) \geq \epsilon] \leq \delta, \tag{8}$$

and

$$Pr[|d(h_1^0, h^*) - d(h_2^1, h^*)| \geq \epsilon] \leq \delta. \tag{9}$$

Proof. In the Co-Training Process, the training data of the classifier h_2^1 contain the l number of initial labeled examples and the u number of newly labeled examples given by the classifier h_1^0 . When $u \gg l$, it could be considered that the training data of the classifier h_2^1 comes from another distribution $\mathcal{D}'$ generated by the classifier h_1^0 which is different from h^*. In distribution $\mathcal{D}'$, the ground truth is h_1^0. According to the PAC learning theory, for any $0 < \epsilon < 1$, in distribution $\mathcal{D}'$ if u is large enough,

$$Pr[d(h_1^0, h_2^1) \geq \epsilon] \leq \delta.$$

Eq.9 is true considering

$$\begin{aligned} & Pr_{x \in \mathcal{D}}[h_1^0(x) \neq h^*(x))] - Pr_{x \in \mathcal{D}}[h_2^1(x) \neq h_1^0(x))] \\ \leq\ & Pr_{x \in \mathcal{D}}[h_2^1(x) \neq h^*(x))] \\ \leq\ & Pr_{x \in \mathcal{D}}[h_1^0(x) \neq h^*(x))] + Pr_{x \in \mathcal{D}}[h_2^1(x) \neq h_1^0(x))] \end{aligned}$$

□

From Theorem 2 we can find that when $u \gg l$, the difference between the two learners is very small (less than ϵ). The two learners become very similar and could not improve each other any more. In Section 4, we will report on an empirical study to show that whether the appropriate learning round of co-training could be estimated based on Theorem 2.

In the above we have discussed the situation when we should proceed with the Co-Training Process to improve the two learners, and when we should terminate the Co-Training Process. As a short summary, our theoretical study shows that

– If the two initial learners have large difference, they can be improved through the Co-Training Process;
– If the two initial learners have small difference, they can be improved if u/l is small;
– As the Co-Training Process proceeds, more and more unlabeled data are labeled for the learners each other, which makes the difference between the two learners become smaller and smaller. Thus, after a number of learning rounds the Co-Training Process could not improve the performance further.

4 Empirical Study

In order to study that whether the appropriate number of learning rounds of co-training could be estimated based on Theorem 2, we perform an empirical study.

4.1 Configurations

In the experiments we use the *course* data set [5], *ads* data set [11] and three UCI data sets , i.e. *kr-vs-kp*, *mushroom* and *tic-tac-toe* [4]. The *course* and *ads* data sets have multiple views but the UCI data sets have only one view.

The *course* data set has two views (*pages* view and *links* view) and contains 1,051 examples each corresponds to a web page, and the task is to predict whether an unseen web page is a *course page* or not. There are 230 positive examples (roughly 22%). Sixty-six attributes are used in *pages* view and five attributes in *links* view. The *ads* data set has five views. We use the 1st and 3rd views since the standard co-training algorithm only uses two views. This data set contains 983 examples, among which there are 138 positive examples (roughly 14%). As for the UCI data sets, *kr-vs-kp* contains 3,196 examples, among which there are 1,527 positive examples (roughly 48%); *mushroom* contains 8,124 examples, among which there are 3,916 positive examples (roughly 48%); *tic-tac-toe* contains 958 examples, among which there are 332 positive examples (roughly 35%). For each of these data sets, we randomly use 25% data as the test set while using the remaining 75% data to generate a labeled data set $\mathcal{L}$ whose concrete size will be mentioned later, and using the rest of the 75% data to generate the unlabeled data set $\mathcal{U}$.

In each learning round, each classifier labels the positive and negative examples on which it is with the most confidence for the other classifier, and the number of newly labeled positive and negative examples is in proportion to that of the positive and negative examples in $\mathcal{L}$. Since the size of $\mathcal{L}$ plays an important role in the Co-Training Process, we run experiments with different sizes of $\mathcal{L}$ on each data set. Moreover, each experiment is repeated for 20 runs and the average performance is recorded.

We study whether we can estimate the appropriate number of learning rounds to terminate the co-training process for avoiding wasteful further training. Here

we estimate the value of $d(h_1^i, h_2^i)$ using $\mathcal{L} \cup \mathcal{U}$. Note that this is a simplification since as described in Section 3, the difference between h_1^{i-1} and h_2^i and that between h_1^i and h_2^{i-1} should be estimated. When the difference between the two classifiers is smaller than $min[a_i, b_i]$ and $u > max\{Lb_i/[a_{i-1} - d(h_1^{i-1}, h_2^i)], La_i/[b_{i-1} - d(h_1^i, h_2^{i-1})]\}$, we terminate the Co-Training Process. Since in real-world tasks we do not have the test data to estimate the error of the two classifiers, we could not estimate the value of a_i and b_i directly. In Theorem 2 we know that the difference between the two classifiers will be stable when u is large, so we could utilize the stability of the difference to roughly estimate the appropriate round for termination. Note that this is an approximation which may cause the estimated round inaccurate. In our experiments we set to terminate the Co-Training Process when the change of the difference in consecutive three rounds is smaller than 0.001. We run the Co-Training Process with various classifiers including SMO, NaiveBayes and MultilayerPerceptron in WEKA [15].

4.2 Results on Data with Two Views

Firstly, we run experiments with the same classifier (SMO) on data sets with two views (i.e., the *course* and *ads* data sets) using the standard co-training algorithm [5]. The results are shown in Table 1.

Table 1. Experimental results on data sets with two views. *SMO* is used to train the classifiers. *data-a-b-c-d* means that on the data set *data*, the initial labeled training set contains a positive examples and b negative examples, and in each round each classifier labels c positive and d negative examples for the other classifier. e_{C_1} and e_{C_2} denote the error rates of the two classifiers trained in the two views, respectively. *dis* denotes the disagreement of the two classifiers. r denotes the number of learning rounds.

	Initial round				Estimated round				Last round			
Data set	e_{C_1}	e_{C_2}	*dis*	r	e_{C_1}	e_{C_2}	*dis*	r	e_{C_1}	e_{C_2}	*dis*	r
course-3-9-1-3	.151	.179	.157	0	.119	.127	.145	60	.119	.127	.145	60
course-6-18-1-3	.127	.155	.177	0	.124	.121	.166	20	.113	.122	.143	60
course-9-27-1-3	.125	.136	.171	0	.118	.116	.154	49	.114	.118	.148	60
ads-4-24-1-6	.114	.100	.046	0	.086	.075	.038	15	.079	.068	.032	25
ads-8-48-1-6	.104	.081	.055	0	.079	.067	.034	21	.078	.062	.031	25
ads-12-72-1-6	.093	.072	.058	0	.082	.065	.041	11	.076	.059	.034	25

We can find from Table 1 that the performances of the classifiers at the estimated round can be quite close to the performances of the classifiers at the last learning round, e.g. on *course*-9-27-1-3. This suggests that estimating the appropriate terminating round based on Theorem 2 is feasible for the standard co-training algorithm.

4.3 Results on Data without Two Views

Then, we run experiments on data sets without two views, by using two different classifiers on the same view. Here we use SMO and NaiveBayes as the two different base learners on the *kr-vs-kp*, *mushroom* and *tic-tac-toe* data set. The results are shown in Table 2.

Table 2. Experimental results on data sets without two views. *SMO* and *NaiveBayes* are used to train the two classifiers, respectively. *data-a-b-c-d* means that on the data set *data*, the initial labeled training set contains a positive examples and b negative examples, and in each round each classifier labels c positive and d negative examples for the other classifier. e_{C_1} and e_{C_2} denote the error rates of the classifiers *SMO* and *NaiveBayes*, respectively. *dis* denotes the disagreement of the two classifiers. r denotes the number of learning rounds.

	Initial round				Estimated round				Last round			
Data set	e_{C_1}	e_{C_2}	*dis*	r	e_{C_1}	e_{C_2}	*dis*	r	e_{C_1}	e_{C_2}	*dis*	r
kr-vs-kp-35-35-1-1	.137	.220	.175	0	.134	.164	.096	50	.134	.164	.096	50
kr-vs-kp-50-50-1-1	.096	.186	.178	0	.097	.136	.079	50	.097	.136	.079	50
kr-vs-kp-65-65-1-1	.087	.182	.172	0	.090	.128	.079	50	.090	.128	.079	50
mushroom-3-3-1-1	.173	.168	.060	0	.130	.129	.026	32	.130	.134	.027	50
mushroom-6-6-1-1	.096	.100	.059	0	.089	.093	.044	26	.082	.088	.028	50
mushroom-12-12-1-1	.077	.097	.064	0	.069	.085	.043	19	.066	.080	.030	50
tic-tac-toe-10-10-1-1	.432	.433	.197	0	.423	.419	.099	39	.424	.424	.084	50
tic-tac-toe-15-15-1-1	.378	.410	.191	0	.370	.403	.104	31	.373	.399	.102	50
tic-tac-toe-20-20-1-1	.355	.392	.181	0	.353	.403	.102	39	.359	.396	.093	50

We can find that the performances of the classifiers at the estimated round can be quite close to the performances of the classifiers at the last learning round, e.g. on *mushroom*-3-3-1-1. This suggests that estimating the appropriate terminating round based on Theorem 2 is also feasible for co-training style algorithms which do not require two views, e.g. [8, 16].

The estimated round is sometimes relatively loose, but the results shown in Tables 1 and 2 verify that after a number of learning rounds, continuing the co-training process could not improve the performance further. It is expected that by developing better methods for estimating the difference between the learners, tighter estimation on the appropriate terminating round could be obtained, which is a future issue.

4.4 Further Experiments and Discussion

In order to study the influence of the difference between the two learners further, more experiments are conducted. We run the Co-Training Process with two different groups of base learners on the *pages* view of the *course* data set. The

first group is SMO and MultilayerPerceptron and the second group is SMO and NaiveBayes. With this experiment, it could be more clear that whether the learners with larger difference could be improved more than the learners with smaller difference. The results are shown in Table 3.

Table 3. Comparing the performance of co-training using two different groups of base learners on the *pages* view of the *course* data set. *SMO* and *MultilayerPerceptron* (or *SMO* and *NaiveBayes*) denote the two classifiers, respectively. *data-a-b-c-d* means that on the data set *data*, the initial labeled training set contains a positive examples and b negative examples, and in each round each classifier labels c positive and d negative examples for the other classifier. e_{C_1} and e_{C_2} denote the error rates of the classifiers *SMO* and *MultilayerPerceptron*(or *SMO* and *NaiveBayes*), respectively. *dis* denotes the disagreement of the two classifiers. r denotes the number of learning rounds.

	Initial round				Estimated round				Last round			
Data set	e_{C_1}	e_{C_2}	*dis*	r	e_{C_1}	e_{C_2}	*dis*	r	e_{C_1}	e_{C_2}	*dis*	r
	C_1 = SMO & C_2 = MultilayerPerceptron											
pagesview-3-9-1-3	.137	.139	.018	0	.127	.126	.043	11	.123	.118	.026	50
pagesview-9-27-1-3	.113	.118	.036	0	.107	.108	.041	13	.099	.105	.028	50
pagesview-15-45-1-3	.100	.101	.038	0	.089	.090	.033	35	.087	.087	.029	50
	C_1 = SMO & C_2 = NaiveBayes											
pagesview-3-9-1-3	.137	.133	.069	0	.106	.095	.040	15	.097	.087	.031	50
pagesview-9-27-1-3	.113	.097	.075	0	.096	.085	.045	23	.087	.078	.036	50
pagesview-15-45-1-3	.100	.081	.076	0	.089	.075	.048	20	.078	.074	.032	50

It can be found from Table 3 that the difference between the second group of classifiers is larger than that between the first group of classifiers. Note that the SMO classifier appears in both groups, while its improvement is larger in the second group than in the first group. This confirms that the larger the difference between the two learners, the more the improvement from the Co-Training Process.

5 Conclusion

In this paper, we present a new PAC analysis on co-training style algorithms. We theoretically explain that why co-training without two views can succeed, and that why co-training could not improve the performance further after a number of learning rounds. Our theoretical result on the second issue provides a feasible approach for estimating the appropriate learning rounds to terminate the co-training process to avoid wasteful learning rounds. We study the effectiveness of the approach in empirical study.

The current estimation of the appropriate learning rounds requires information on the generalization ability of the learners. Since such information is not available in real-world tasks, we use an approximation to realize the approach.

So, although the approach has a theoretical foundation and empirical study shows that it works not bad, the approximation makes the estimation not as accurate as we have expected. To improve the estimation in real-world tasks is a future issue.

Note that in some empirical study of the natural language processing community, it has been found that sometimes the performances of the two learners can degrade if the co-training process is run to convergence [14]. Our theoretical study in this paper gives an explanation to this phenomenon. That is, after a number of learning rounds the co-training process could not improve the performance further since the difference between the learners becomes very small. In other words, the two learners becomes very similar. Thus, if the co-training process is continued to convergence, these two learners will have very high chance to make similar errors. Since the co-trained learners are usually combined to use, the similar errors will be reinforced. Thus, overfitting is aggravated and therefore the degradation of performance is observed.

Acknowledgment

We want to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Science Foundation of China (60635030, 60505013) and the Foundation for the Author of National Excellent Doctoral Dissertation of China (200343).

References

1. Abney, S.: Bootstrapping. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA (2002) 360–367
2. Balcan, M.F., Blum, A., Yang, K.: Co-training and expansion: Towards bridging theory and practice. In Saul, L.K., Weiss, Y., Bottou, L., eds.: Advances in Neural Information Processing Systems 17. MIT Press, Cambridge, MA (2005) 89–96
3. Belkin, M., Niyogi, P.: Semi-supervised learning on Riemannian manifolds. Machine Learning **56** (2004) 209–239
4. Blake, C., Keogh, E., Merz, C.J.: UCI repository of machine learning databases. [http://www.ics.uci.edu/∼mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA (1998)
5. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory, Madison, WI (1998) 92–100
6. Chapelle, O., Weston, J., Schölkopf, B.: Cluster kernels for semi-supervised learning. In Becker, S., Thrun, S., Obermayer, K., eds.: Advances in Neural Information Processing Systems 15. MIT Press, Cambridge, MA (2003) 585–592
7. Dasgupta, S., Littman, M., McAllester, D.: PAC generalization bounds for co-training. In Dietterich, T.G., Becker, S., Ghahramani, Z., eds.: Advances in Neural Information Processing Systems 14. MIT Press, Cambridge, MA (2002) 375–382
8. Goldman, S., Zhou, Y.: Enhancing supervised learning with unlabeled data. In: Proceedings of the 17th International Conference on Machine Learning, San Francisco, CA (2000) 327–334

9. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the 16th International Conference on Machine Learning, Bled, Slovenia (1999) 200–209
10. Krogel, M.A., Scheffer, T.: Effectiveness of information extraction, multi-relational, and semi-supervised learning for predicting functional properties of genes. In: Proceedings of the 3rd IEEE International Conference on Data Mining, Melbourne, FL (2003) 569–572
11. Kushmerick, N.: Learning to remove internet advertisements. In: Proceedings of the 3rd Annual Conference on Autonomous Agents, Seattle, WA (1999) 175–181
12. Mladenic, D.: Modeling information in textual data combining labeled and unlabeled data. (In: Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery) 170–179
13. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Machine Learning **39** (2000) 103–134
14. Pierce, D., Cardie, C.: Limitations of co-training for natural language learning from large data sets. In: Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing, Pittsburgh, PA (2001) 1–9
15. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. 2nd edn. Morgan Kaufmann, San Francisco, CA (2005)
16. Zhou, Z.H., Li, M.: Semi-supervised regression with co-training. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh, Scotland (2005) 908–913
17. 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, Washington, DC (2003) 912–919