Instance Selection in Semi-supervised Learning

Yuanyuan Guo[1], Harry Zhang[1], and Xiaobo Liu[2]

[1] Faculty of Computer Science, University of New Brunswick
P.O. Box 4400, Fredericton, NB, Canada E3B 5A3
{yuanyuan.guo,hzhang}@unb.ca
[2] School of Computer Science, China University of Geosciences
Wuhan, Hubei, China 430074
jerrycug@yahoo.com.cn

Abstract. Semi-supervised learning methods utilize abundant unlabeled data to help to learn a better classifier when the number of labeled instances is very small. A common method is to select and label unlabeled instances that the current classifier has high classification confidence to enlarge the labeled training set and then to update the classifier, which is widely used in two paradigms of semi-supervised learning: self-training and co-training. However, the original labeled instances are more reliable than the self-labeled instances that are labeled by the classifier. If unlabeled instances are assigned wrong labels and then used to update the classifier, classification accuracy will be jeopardized. In this paper, we present a new instance selection method based on the original labeled data (ISBOLD). ISBOLD considers not only the prediction confidence of the current classifier on unlabeled data but also its performance on the original labeled data only. In each iteration, ISBOLD uses the change of accuracy of the newly learned classifier on the original labeled data as a criterion to decide whether the selected most confident unlabeled instances will be accepted to the next iteration or not. We conducted experiments in self-training and co-training scenarios when using Naive Bayes as the base classifier. Experimental results on 26 UCI datasets show that, ISBOLD can significantly improve accuracy and AUC of self-training and co-training.

Keywords: self-training, co-training, instance selection.

1 Introduction

In many real-world machine learning applications, it may be expensive or time-consuming to obtain a large amount of labeled data. On the other hand, it is relatively easy to collect lots of unlabeled data. Learning classifiers from a small number of labeled training instances may not produce good performance. Therefore, various algorithms have been proposed to exploit and utilize the unlabeled data to help to learn better classifiers. Semi-supervised learning is one kind of such algorithms that use both labeled data and unlabeled data.

C. Butz and P. Lingras (Eds.): Canadian AI 2011, LNAI 6657, pp. 158–169, 2011.

Many semi-supervised learning algorithms have been proposed in the past decades, including self-training, co-training, semi-supervised support vector machines, graph-based methods, and so on [2,13]. The general idea of self-training [12] and co-training [1] is to iteratively pick some unlabeled instances according to a given selection criterion and move them (together with the labels assigned by the classifier) to the training set to build a new classifier. These selected instances are called "self-labeled" instances in [5]. The main difference between self-training and co-training is that, in co-training, the attributes are split into two separate sub-views and every operation is conducted on the two sub-views, respectively.

A commonly used instance selection criterion is "confidence selection" which selects unlabeled instances that are predicted by the current classifier with high confidence [1,2,6,8,12], that is, the instances with the high class membership probabilities. Other selection methods have also been proposed by researchers. Wang et al. presented an adapted Value Difference Metric as the selection metric in self-training, which does not depend on class membership probabilities [10]. In [5], a method named SETRED is presented that utilizes the information of the neighbors of each self-labeled instance to identify and remove the mislabeled examples from the self-labeled data.

Ideally, the selected unlabeled instances (together with the predicted labels) can finally help to learn a better classifier. In [3], however, it concludes that unlabeled data may degrade classification performance in some extreme conditions and under common assumptions when the model assumptions are incorrect. In our previous work [4], an extensive empirical study was conducted on some common semi-supervised learning algorithms (including self-training and co-training) using different base Bayesian classifiers. Results on 26 UCI datasets show that, the performance of using "confidence selection" is not necessarily superior to that of randomly selecting unlabeled instances. If the current classifier has poor performance and wrongly assigns labels to some self-labeled instances, the final performance will be jeopardized due to the accumulation of mislabeled data. It is a general problem for the methods based on the classifier performance on the expanded data, including the original labeled data and the self-labeled data. Since the originally labeled instances are generally more reliable than self-labeled instances, the performance on the former instances alone is more critical. Thus, we conjecture that, the classifier should have a good performance on the original labeled data if it wants to have good prediction performance on future data. More precisely, when the accuracy of the classifier evaluated on the original labeled data decreases, the accuracy on the future testing set generally degrades as well. Hence, utilizing the accuracy on the original labeled data to select more reliable unlabeled instances seems crucial to the final performance of semi-supervised learning.

In this paper, we present an effective instance selection method based on the original labeled data (ISBOLD) to improve the performance of self-training and co-training when using Naive Bayes (NB) as the base classifier. ISBOLD considers both the prediction confidence of the current classifier on the self-labeled

data and the accuracy on the original labeled data only. In each iteration, after the selection of the most confident unlabeled instances, the accuracy of the current classifier on the original labeled data is computed and then used to decide whether to add the selected instances to the training set in the next iteration. Experiments on 26 UCI datasets demonstrate that, ISBOLD significantly improves the accuracy of self-training and co-training on 6 to 7 datasets and prevents the performance being degraded on the other datasets, compared to our experimental results in [4]. Besides, ISBOLD significantly improves AUC on 8 to 9 datasets.

The rest of the paper is organized as follows. Section 2 briefly describes self-training and co-training algorithms and reviews related research work. A new instance selection method based on the original labeled data (ISBOLD) is presented in Section 3. Section 4 shows experimental results on 26 UCI datasets, as well as detailed performance analysis. Finally, it is concluded in Section 5.

2 Related Work

Semi-supervised learning methods utilize unlabeled data to help to learn better classifiers when the amount of labeled training data is small. A set L of labeled training instances and a set U of unlabeled instances are given in semi-supervised learning scenario. In [13], a good survey of research work on several well-known semi-supervised learning methods has been given. These algorithms and their variants are also analyzed and compared in [2]. Self-training and co-training are two common algorithms among them.

2.1 Self-training and Co-training Algorithms

Self-training works as follows [12]. A classifier is built from L and used to predict the labels for instances in U. Then m instances in U that the current classifier has high classification confidence are labeled and moved to enlarge L. The whole process iterates until stopped.

Co-training works in a similar way except that it is a two-view learning method [1]. Initially, the attribute set (view) is partitioned into two conditionally independent sub-sets (sub-views). A data pool U' is created by randomly choosing some instances from U for each sub-view, respectively. On each sub-view, a classifier is built from the labeled data and then used to predict labels for the unlabeled data in its data pool. A certain number of unlabeled instances that one classifier has high classification confidence are labeled and moved to expand the labeled data of the other classifier. And the same number of unlabeled instances will be randomly moved from U to replenish U'. Then the two classifiers are rebuilt from their corresponding updated labeled data, respectively. The process iterates until stopped. In other words, in co-training, it iteratively and alternately uses one classifier to help to "train" another classifier.

The stopping criterion in self-training and co-training is that, either there is no unlabeled instance left or the maximum number of iterations has been reached.

There are two assumptions in co-training to ensure good performance [1]: each sub-view is sufficient to build a good classifier; and the two sub-views are conditionally independent of each other given the class. The two assumptions may be violated in real-world applications. In [8], it is stated that, co-training still works when the attribute set is randomly divided into two separate subsets, although the performance may not be as good as when the attributes are split sufficiently and independently.

2.2 Variants of Self-training and Co-training Algorithms

Researchers have presented different variants of self-training and co-training algorithms.

One kind of methods is to use all the unlabeled instances in each iteration so that no selection criterion is needed. A self-training style method, semi-supervised EM, is presented in [9]. During each iteration, all the unlabeled instances are given predicted labels and then used to enlarge the training set and update the classifier. In [8], co-training is combined with EM to generate a new algorithm co-EM which in each iteration uses all the unlabeled instances instead of a number of instances picked from the data pool.

Another kind of methods is to use active learning method to select unlabeled instances and then ask human experts to label them. Hence, no mislabeled examples will occur, in principle. In [7], an active learning method is used to select unlabeled instances for the multi-view semi-supervised Co-EM algorithm. And labels are assigned to the selected unlabeled instances by experts. However, active learning methods are not applicable if we do not have available human experts.

Some researchers also used different selection techniques to decide which unlabeled instances should be used in each iteration. In [10], the authors presented an adapted Value Difference Metric as the selection metric in self-training. In [5], a data editing method is applied to identify and remove the mislabeled examples from the self-labeled data.

In our previous work [4], an empirical study on 26 UCI datasets shows that, in self-training and co-training, using "confidence selection" cannot always outperform that of randomly selecting unlabeled instances. If the classification performance of the current classifier is poor, wrong labels may be predicted to most unlabeled instances and the final performance of semi-supervised learning will be affected accordingly. Generally speaking, the original labeled instances are more reliable than the instances with predicted labels by the current classifier. Hence, the performance on the original labeled data is an important factor to reflect the final performance of semi-supervised learning.

3 Instance Selection Based on the Original Labeled Data

Motivated by the existing work, in this paper, we present a new method, Instance Selection Based on the Original Labeled Data (ISBOLD), to improve the

performance of self-training and co-training when using NB as the base classifier. The main idea of ISBOLD is to use the accuracy on the original labeled data only to prevent adding unlabeled instances that will possibly degrade the performance. How to use ISBOLD in self-training and co-training scenarios is described in following two subsections, respectively.

3.1 ISBOLD for Self-training

In order to describe our method, some notations are used here. In iteration t, we use L_t to denote the new labeled training set, C_t to represent the classifier built on L_t, and Acc_t as the accuracy of C_t on the original labeled data L_0. The detailed algorithm is shown in Figure 1.

---

1. Set t, the iteration counter, to 0.
2. Build a classifier C_t on the original labeled data L_0.
3. Compute Acc_t, which is the accuracy of C_t on L_0.
4. While the stopping criteria are not satisfied,
   (a) Use C_t to predict a label for each instance in U.
   (b) Generate L^s_{t+1}: select m unlabeled instances that C_t has high classification confidence, and assign a predicted label to each selected instance. Delete the selected instances from U.
   (c) $L_{t+1} = L_t \cup L^s_{t+1}$.
   (d) Build a classifier C_{t+1} on L_{t+1}.
   (e) Compute Acc_{t+1}, which is the accuracy of C_{t+1} on L_0.
   (f) If $Acc_{t+1} < Acc_t$, then $L_{t+1} = L_t$, and rebuild C_{t+1} on L_{t+1}.
   (g) Increase t by 1.
5. Return the final classifier.

Fig. 1. Algorithm of ISBOLD for self-training

The difference between ISBOLD and the common confidence selection method in self-training is displayed in steps 4(e) and 4(f). In iteration $t+1$, after selecting the most confident unlabeled instances and assigning labels to them (for simplicity, the set of those selected instances is denoted as L^s_{t+1}), the training set $L_{t+1} = L_t \cup L^s_{t+1}$. Now we build a classifier C_{t+1} on L_{t+1} and compute Acc_{t+1}. If $Acc_{t+1} < Acc_t$, L_{t+1} is reset to be equal to L_t, and C_{t+1} is updated on L_{t+1} accordingly. The whole process iterates until there is no unlabeled instance left or the maximum number of iterations is reached.

The reason that we remove L^s_{t+1} from L_{t+1} once the accuracy on L_0 decreases is that, if adding L^s_{t+1} to the training set degrades the classifier's performance on L_0, it is very possible that the performance of the current classifier on the test set degrades as well. Hence, we use this method to roughly prevent possible performance degradation. Furthermore, notice that in step 4(b), all the selected instances are removed from U, which means that each selected instance is either added to the labeled data or removed from U.

3.2 ISBOLD for Co-training

A similar selection method is used in co-training. We denote the classifiers on the two sub-views in iteration t as C_t^a and C_t^b. The algorithm is shown in Figure 2.

1. Set t, the iteration counter, to 0.
2. Randomly partition the attribute set Att into two separate sets Att_a and Att_b. Generate L_0^a and L_0^b from L. Generate U_a and U_b from U.
3. Generate data pool U_a' and U_b' by randomly choosing u instances from U_a and U_b, respectively.
4. Use L_0^a to train a classifier C_t^a.
5. Use L_0^b to train a classifier C_t^b.
6. Compute Acc_t^a, which is the accuracy of C_t^a on L_0^a.
7. Compute Acc_t^b, which is the accuracy of C_t^b on L_0^b.
8. While the stopping criteria are not satisfied,
   (a) Use C_t^a to predict a label for each instance in U_a'. Use C_t^b to predict a label for each instance in U_b'.
   (b) Generate $L_{t+1}^{a^s}$: select m unlabeled instances that C_t^b has high classification confidence, together with predicted labels. Delete the selected instances from U_b'.
   (c) Generate $L_{t+1}^{b^s}$: select m unlabeled instances that C_t^a has high classification confidence, together with predicted labels. Delete the selected instances from U_a'.
   (d) $L_{t+1}^a = L_t^a \cup L_{t+1}^{a^s}$. $L_{t+1}^b = L_t^b \cup L_{t+1}^{b^s}$.
   (e) Use L_{t+1}^a to train a classifier C_{t+1}^a.
   (f) Compute Acc_{t+1}^a, which is the accuracy of C_{t+1}^a on L_0^a.
   (g) If $Acc_{t+1}^a < Acc_t^a$, then $L_{t+1}^a = L_t^a$, and rebuild C_{t+1}^a on L_{t+1}^a.
   (h) Use L_{t+1}^b to train a classifier C_{t+1}^b.
   (i) Compute Acc_{t+1}^b, which is the accuracy of C_{t+1}^b on L_0^b.
   (j) If $Acc_{t+1}^b < Acc_t^b$, then $L_{t+1}^b = L_t^b$, and rebuild C_{t+1}^b on L_{t+1}^b.
   (k) Randomly move m instances from U_a to replenish U_a'.
   Randomly move m instances from U_b to replenish U_b'.
   (l) Increase t by 1.

Fig. 2. Algorithm of ISBOLD for co-training

The difference between ISBOLD and the common confidence selection method in co-training is displayed in steps 8(f), 8(g), 8(i) and 8(j). In iteration $t+1$, on sub-view a, after selecting a certain number of unlabeled instances that C_t^b has high classification confidence, a label is assigned to each selected instance (for simplicity, the set of those selected instances is denoted as $L_{t+1}^{a^s}$). Then $L_{t+1}^a = L_t^a \cup L_{t+1}^{a^s}$ and C_{t+1}^a is built on L_{t+1}^a. Now we compute Acc_{t+1}^a that represents the accuracy of C_{t+1}^a on L_0^a. If $Acc_{t+1}^a < Acc_t^a$, $L_{t+1}^a = L_t^a$ and C_{t+1}^a is updated accordingly. The same steps are repeated on sub-view b to generate L_{t+1}^b and C_{t+1}^b. New unlabeled instances will be replenished from the remaining

unlabeled data part to the data pool of each sub-view. The whole process iterates until there is no unlabeled instance left or the maximum number of iterations is reached.

4 Experimental Results and Analysis

4.1 Experimental Settings

In order to examine the performance of ISBOLD, we conducted experiments on 26 UCI datasets, including 18 binary class datasets and 8 multi-class datasets. These datasets are downloaded from a package of 37 classification problems, "datasets-UCI.jar"[1]. Each dataset is then preprocessed in Weka software [11] by replacing missing values, discretization and removing any attribute that its number of attribute values is almost equal to the number of instances in the dataset [4]. We only use 26 datasets out of the package because the other 11 datasets have extremely skewed class distributions. For example, in the *hypothyriod* dataset, the frequency of each class value is 3481, 194, 95 and 2 respectively. When randomly sampling the labeled data set in semi-supervised learning, the classes that have very small values of frequency may not appear in some generated datasets if we want to keep the same class distributions. Usually researchers merge the minor classes into a major class or simply delete instances with minor classes. However, to minimize any possible influence, we ignored those datasets with extremely skewed class distributions. The 26 datasets are the same as those used in our previous work [4].

On each dataset, 10 runs of 4-fold stratified cross-validation are conducted. That is, 25% of the original data will be put aside as the testing set to evaluate the performance of learning algorithms. The remaining 75% data are divided into labeled data (L) and unlabeled data (U) according to a pre-defined percentage of labeled data (lp). The data splitting setting follows those in [1,4,5,6]. In our experiments, lp is set to be 5%. Therefore, 25% data are kept as the testing set, 5% of the 75% data are randomly sampled as L while the remaining 95% of the 75% data are saved as U. When generating L, we made sure that L and the original training data had the same class distributions.

Naive Bayes is used in self-training and co-training. The maximum number of iterations in both is set to 80. The size of data pool in co-training is set to be 50% of the size of U. Accuracy and AUC are used as performance measurements. In our experiments on co-training, the attributes are randomly split into two subsets.

4.2 Results Analysis

Performance comparison results of using ISBOLD and using the common "confidence selection" method in self-training and co-training are shown in Table 1 and Table 2. For simplicity, the methods are denoted as **ISBOLD** and **CF**

[1] They are available from http://www.cs.waikato.ac.nz/ml/weka/

Table 1. Accuracy of **CF** vs **ISBOLD** in self-training and co-training

(a) self-training

Dataset	CF	ISBOLD
balance-scale	59.52	66.21
breast-cancer	65.09	65.61
breast-w	96.67	96.34
colic	74.54	75.38
colic.ORIG	55.05	60.57
credit-a	80.68	80.78
credit-g	60.62	66.03 v
diabetes	70.55	70.53
heart-c	81.55	81.15
heart-h	83.06	82.41
heart-statlog	81.37	80.74
hepatitis	79.70	78.34
ionosphere	80.97	79.86
iris	90.31	90.05
kr-vs-kp	67.26	80.07 v
labor	88.26	87.92
letter	40.38	57.39 v
mushroom	91.90	92.57 v
segment	63.49	72.88 v
sick	91.54	94.15
sonar	55.72	57.93
splice	82.05	85.48 v
vehicle	41.79	48.35
vote	87.89	88.53
vowel	18.75	21.78
waveform-5000	77.98	78.87
mean	71.80	74.61
w/t/l		6/20/0

(b) co-training

Dataset	CF	ISBOLD
balance-scale	59.10	67.17
breast-cancer	70.41	71.00
breast-w	96.85	96.47
colic	76.60	75.76
colic.ORIG	55.19	62.04
credit-a	81.36	79.67
credit-g	63.04	67.72 v
diabetes	67.51	69.58
heart-c	82.77	80.13
heart-h	81.46	78.60
heart-statlog	82.03	80.30
hepatitis	81.04	80.21
ionosphere	81.50	83.08
iris	80.79	78.98
kr-vs-kp	59.22	77.36 v
labor	77.21	78.43
letter	36.67	56.05 v
mushroom	91.74	92.38 v
segment	61.49	71.64 v
sick	93.40	93.56
sonar	55.43	58.08
splice	73.91	82.63 v
vehicle	41.57	47.86
vote	88.21	88.60
vowel	18.83	23.36
waveform-5000	71.61	75.91 v
mean	70.34	73.71
w/t/l		7/19/0

in the tables. In each table, figures on each row are the average accuracy or AUC over 10-runs of 4-fold cross-validation on the corresponding dataset. Row "w/t/l" represents that using ISBOLD in the corresponding column wins on w datasets (marked by 'v'), ties on t datasets, and loses on l datasets (marked by '*') against using "confidence selection" in self-training or co-training, under a two-tailed pair-wise t-test with the significant level of 95%. Values in row "mean" are the average accuracy or AUC over the 26 datasets.

Table 1(a) shows the average accuracy of using **ISBOLD** and **CF** in self-training. The "w/t/l" t-test results show that, ISBOLD significantly improves classification accuracy on 6 datasets. Values in row "mean" also demonstrate that ISBOLD improves the average performance. Table 1(b) shows the average accuracies in co-training. The "w/t/l" t-test results tell that ISBOLD significantly improves the performance of co-training on 7 datasets. And the mean value increases from 70.34 to 73.71.

Table 2. AUC of **CF** vs **ISBOLD** in self-training and co-training

(a) self-training

Dataset	CF	ISBOLD
balance-scale	61.37	66.68
breast-cancer	63.98	63.48
breast-w	99.07	99.08
colic	79.24	78.43
colic.ORIG	51.62	58.49
credit-a	86.81	86.79
credit-g	56.56	65.24 v
diabetes	78.03	76.36
heart-c	83.97	83.92
heart-h	83.74	83.74
heart-statlog	88.93	88.64
hepatitis	83.02	80.99
ionosphere	86.86	86.68
iris	98.33	98.29
kr-vs-kp	74.65	89.03 v
labor	96.59	96.72
letter	86.09	93.08 v
mushroom	98.04	98.81 v
segment	90.86	95.24 v
sick	91.51	93.96
sonar	58.64	62.21
splice	94.40	96.23 v
vehicle	59.63	66.95 v
vote	96.31	96.52
vowel	57.65	64.49 v
waveform-5000	88.85	90.96 v
mean	80.57	83.12
w/t/l		9/17/0

(b) co-training

Dataset	CF	ISBOLD
balance-scale	60.44	65.34
breast-cancer	63.51	64.37
breast-w	99.22	99.19
colic	78.99	79.08
colic.ORIG	49.62	55.82
credit-a	88.05	86.35
credit-g	55.33	61.62
diabetes	72.61	74.95
heart-c	84.02	83.80
heart-h	83.77	83.50
heart-statlog	90.03	88.03
hepatitis	78.38	73.19
ionosphere	87.89	88.92
iris	93.21	92.27
kr-vs-kp	66.86	86.39 v
labor	87.76	85.18
letter	82.98	92.57 v
mushroom	97.89	98.75 v
segment	87.93	94.82 v
sick	87.74	93.83
sonar	59.59	62.93
splice	88.65	94.87 v
vehicle	59.56	67.09 v
vote	96.31	96.46
vowel	57.97	66.44 v
waveform-5000	84.22	89.54 v
mean	78.56	81.74
w/t/l		8/18/0

Comparison results on AUC in self-training and co-training are displayed in Table 2. It can be observed that, using ISBOLD, the AUC of self-training is significantly improved on 9 datasets. And the mean value increases from 80.57 to 83.12. Similarly, the AUC of co-training is sharply improved on 8 datasets, and the mean value is improved from 78.56 to 81.74.

4.3 Learning Curves Analysis

Based on our previous work [4], we guess that, the classifier should have a good prediction performance on the testing set if the accuracy on the original labeled data does not degrade. To verify our conjecture and to further examine the performance of ISBOLD during each iteration, learning curves of a random running of two self-training methods on datasets *vehicle* and *kr-vs-kp* are displayed in

Figure 3 and Figure 4, respectively. The data splitting setting is the same as that in subsection 4.1. Curves in co-training are omitted here due to space limitation.

On each graph, at each iteration t, the accuracy values of classifier C_t on the original labeled data L_0 and the testing set for using **ISBOLD** or **CF** in self-training are displayed, respectively. Curves "ISBOLD-L0" and "ISBOLD-test" show accuracy values on the original labeled data L_0 and on the testing set, respectively, when using ISBOLD in self-training on the dataset. Curves "CF-L0" and "CF-test" display accuracy values on L_0 and on the testing set, respectively, when using "confidence selection" in self-training on the dataset.

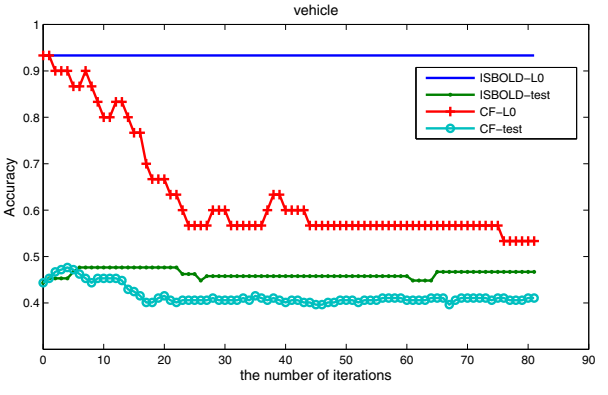

Fig. 3. Learning curves on the *vehicle* dataset

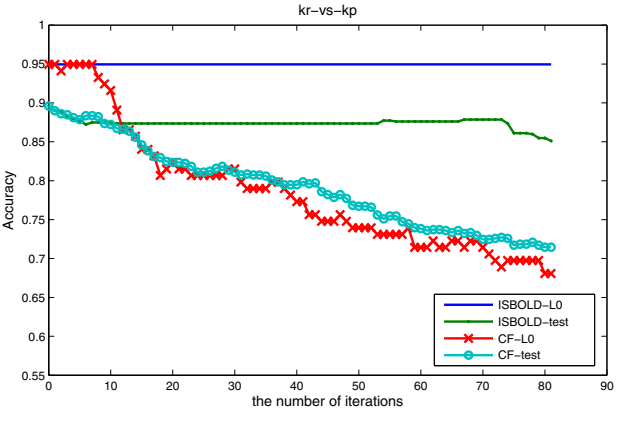

Fig. 4. Learning curves on the *kr-vs-kp* dataset

According to our conjecture, when the accuracy on the original labeled data L_0 decreases, the accuracy on the corresponding testing set generally decreases as well. This is actually observed on the trends of curve "CF-L0" and curve "CF-test" in Figure 3 and Figure 4. Curve "CF-test" generally goes down when curve "CF-L0" goes down.

ISBOLD is presented based on our conjecture that the classifier will have good prediction performance on the testing set if its accuracy on the original labeled data does not degrade during each iteration. As shown in Figure 3 and Figure 4, comparing curves on "confidence selection" method to curves on ISBOLD method, ISBOLD can sharply improve the accuracy on the testing set while improving it on L_0. When the accuracy on L_0 does not degrade, the final accuracy on the testing set does not significantly decrease. These observations confirm that, using the accuracy on the original labeled data to further decide whether to accept the selected unlabeled instances into the next iteration or not is an effective way to improve the performance in semi-supervised learning.

5 Conclusions and Future Work

In this paper, we presented a new instance selection method ISBOLD to improve the performance of self-training and co-training when using NB as the base classifier. During each iteration, after selecting a number of unlabeled instances that the current classifier has high classification confidence, we use the accuracy of the current classifier on the original labeled data to decide whether to accept the selected unlabeled instances to the labeled training set in the next iteration. Experiments on 26 UCI datasets show that ISBOLD can significantly improve the performance of self-training and co-training on many datasets. The learning curve analysis gives a vivid demonstration and experimentally proves the feasibility of our method.

In future work, we will try different base classifiers such as non-naive Bayesian classifiers and decision trees, and extend the method to more semi-supervised learning methods. Besides, theoretical analysis will also be done to help to understand the functionality of the method. Based on these work, we will present new methods to improve the performance of semi-supervised learning.

References

1. Blum, A., Mitchell, T.: Combing labeled and unlabeled data with co-training. In: Proceedings of the 1998 Conference on Computational Learning Theory (1998)
2. Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-supervised learning. MIT Press, Cambridge (2006)
3. Cozman, F.G., Cohen, I.: Unlabeled data can degrade classification performance of generative classifiers. In: Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference (2002)
4. Guo, Y., Niu, X., Zhang, H.: An extensive empirical study on semi-supervised learning. In: The 10th IEEE International Conference on Data Mining (2010)
5. Li, M., Zhou, Z.H.: SETRED: self-training with editing. In: Proceedings of the Advances in Knowledge Discovery and Data Mining (2005)
6. Ling, C.X., Du, J., Zhou, Z.H.: When does co-training work in real data? In: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (2009)

7. Muslea, I., Minton, S., Knoblock, C.A.: Active + semi-supervised learning = robust multi-view learning. In: Proceedings of the Nineteenth International Conference on Machine Learning (2002)
8. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: Proceedings of the 9th International Conference on Information and Knowledge Management (2000)
9. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Machine Learning 39, 103–134 (2000)
10. Wang, B., Spencer, B., Ling, C.X., Zhang, H.: Semi-supervised self-training for sentence subjectivity classification. In: The 21st Canadian Conference on Artificial Intelligence, pp. 344–355 (2008)
11. Witten, I.H., Frank, E. (eds.): Data mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
12. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, pp. 189–196 (1995)
13. Zhu, X.J.: Semi-supervised learning literature survey (2008)