

SETRED: Self-training with Editing

Ming Li and Zhi-Hua Zhou

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

Abstract. Self-training is a semi-supervised learning algorithm in which a learner keeps on labeling unlabeled examples and retraining itself on an enlarged labeled training set. Since the self-training process may erroneously label some unlabeled examples, sometimes the learned hypothesis does not perform well. In this paper, a new algorithm named SETRED is proposed, which utilizes a specific data editing method to identify and remove the mislabeled examples from the self-labeled data. In detail, in each iteration of the self-training process, the local *cut edge weight* statistic is used to help estimate whether a newly labeled example is reliable or not, and only the reliable self-labeled examples are used to enlarge the labeled training set. Experiments show that the introduction of data editing is beneficial, and the learned hypotheses of SETRED outperform those learned by the standard self-training algorithm.

1 Introduction

In many practical machine learning applications, obtaining a fully labeled data set is usually difficult. The requirement of lots of human expertise makes the labeling process fairly expensive. A more feasible way is to label just a small part of data set, leaving a huge amount of examples in data set unlabeled. The learner itself should find a way to exploit the merit of unlabeled data.

Semi-supervised learning is to learn a hypothesis by combining information in both labeled and unlabeled data. *Self-training* [10] is a well-known semi-supervised algorithm. In self-training process, a base learner is firstly trained on labeled set. Then, iteratively, it attempts to choose to label several examples that it is most confident of in the unlabeled set. After that it enlarges its labeled training set with these self-labeled examples. Since the labeled set is usually insufficient for learning, misclassifying a certain amount of unlabeled data is unavoidable. Thus, the enlarged labeled set for the learner to learn in the next iteration could contain much noise. Once those noisy examples are added into the learner's training set, there is no chance for the self-trained learner to reconsider the validity of those incorrect labels, and the mislabeled examples will keep on affecting the learner in the following iterations. If the distribution the learner has caught is badly distorted by those mislabeled examples, the generalization ability degrades as the self-training process goes on. Therefore, it is obvious that

identifying and removing the mislabeled examples in each iteration might help improve the generalization ability of the learned hypothesis.

In this paper, a new self-training style algorithm named SETRED (Self-TRaining with EDiting) is proposed. SETRED introduces a data editing technique to the self-training process to filter out the noise in the self-labeled examples. Specifically, after labeling some examples chosen from the unlabeled set, SETRED actively identifies the possibly mislabeled examples with the help of some local information in a neighborhood graph, and keeps those mislabeled examples from being added to the learner's training set, hence a less noisy training set is obtained. Actually, SETRED could be considered as a semi-supervised algorithm that utilizes an active-learning-like technique to improve its performance. Experiments on ten UCI data sets show that SETRED is more robust than the standard self-training algorithm, and the generalization ability of its learned hypotheses outperform those learned by standard self-training.

The rest of the paper is organized as follows. Section 2 briefly reviews some works on learning from labeled and unlabeled data. Section 3 presents SETRED. Section 4 reports the experiment result. Finally, Section 5 concludes and issues some future work.

2 Learning from Labeled and Unlabeled Data

An effective way to utilize unlabeled data in assistance of supervised learning is known as *semi-supervised learning* [13], where an initial hypothesis is learned from the labeled set and then refined through information derived from the unlabeled set.

In some methods only one base learner is used, which uses the unlabeled examples iteratively according its own knowledge. Such methods include using Estimation-Maximization approach to estimate posterior parameters of a generative model, such as Naive Bayes, by assigning each unlabeled example a soft label, i.e. a probability for each class [11]; using the unlabeled data to search for a better structure of Bayesian Network [2]; using a transductive inference for support vector machines on a special test set [4]. The self-training algorithm [10] is of this kind, where in each iteration the learner converts the most confidently predicted unlabeled example of each class into a labeled training example.

In some other methods, the unlabeled data is utilized with more learners. A representative is the co-training paradigm proposed by Blum and Mitchell [1]. In co-training, two base learners are trained within the multi-view framework, i.e. two sets of independent attributes, each of which is sufficient for classification. One base learner iteratively labels several examples which it is most confident of from its point of view, and feeds them to the other learner. The co-training paradigm has already been successfully applied to many areas such as nature language processing [12].

Besides semi-supervised learning, there is another effective way to use the unlabeled data, that is, *active learning*. Different from semi-supervised learning choosing confident examples to label by itself, active learning actively chooses

most problematic examples from the unlabeled set and asks a teacher for the labels. Two major techniques in active learning are uncertainty-base sampling [5] and committee-base sampling [14].

Since semi-supervised learning and active learning utilize unlabeled data in different way, their merits can be combined through some specifically designs. McCallum and Nigam [6] combined semi-supervised EM with committee-based sampling in text classification. Muslea et al.[9] employed co-testing [8] to choose unlabeled examples to query, and used co-EM [10] to boost the accuracy of the hypotheses. Zhou et al. [16] combined co-training with co-testing in content-based image retrieval.

3 SETRED

Let L and U denote the labeled and unlabeled set drawn from the same distribution $D(\mathbf{X}, Y)$, respectively, where \mathbf{X} is a p -dimensional feature vector while Y is a class label. In standard self-training process, a learner keeps on choosing to label a small set of its most confident examples, say L' , from U and retraining itself on $L \cup L'$. Self-training requires neither estimation and maximization of some posterior probability nor a sufficient and conditional independent attributes, so it is much easier to use than semi-supervised EM and standard co-training. However, due to the small size of L , the generalization ability of the initial hypothesis may be poor. Consequently, L' may contain much noise because the learner may incorrectly assign labels to some unlabeled examples, and the generalization ability of the final hypothesis will be hurt by the accumulation of such noise in each iteration of the training process. Therefore, it is obvious that if the mislabeled examples in L' could be identified in the self-training process, especially in the early iterations, the learned hypothesis is expected to perform better.

Data editing is a technique which attempts to improve the quality of the training set through identifying and eliminating the training examples wrongly generated in the human labeling process. Some useful data editing methods have been studied in [3][15]. In those works, another learner is used to improve the quality of the training set before the wanted learner are trained. In a recent work, Muhlenbach et al. [7] proposed a method based on a statistical method called *cut edge weight* statistic [17] to identify mislabeled examples in the training set. Here this data editing method is employed to identify the examples possibly mislabeled by the learner in the self-training process.

In detail, SETRED initiates the self-training process by firstly learning a hypothesis from the labeled set. In each self-training iteration, the base learner detects unlabeled examples on which it makes most confident prediction and labels those examples according to the prediction. Then, for each possible label y_j (where j ranges from 1 to the number of possible labels), k_j examples are selected and added to L' according to the prediction confidence of their labels, keeping the class distribution in L' similar to that in L . For instance, if there are 4 positive and 16 negative examples in L , then L' contains 1 positive and 4 negative examples.

After L' is formed, the identification of mislabeled examples are performed on the learner’s potential training set $L \cup L'$. Firstly, SETRED constructs a neighborhood graph [17] that expresses certain local information from all the examples in $L \cup L'$. A neighborhood graph is a graph in p -dimensional feature space where a distance metric could be defined. Each example in the graph is a vertex and there exists an edge between two vertices a and b if the distance between a and b satisfies Eq. 1. An edge connecting two vertices that have different labels is called *cut edge*.

$$\text{Dist}(a, b) \leq \max(\text{Dist}(a, c), \text{Dist}(b, c)) \tag{1}$$

Then SETRED identifies the mislabeled examples based on their neighborhood in the graph. The neighborhood of an example is a set of examples it connected to with edges in graph. Intuitively, most examples possess the same label in a neighborhood. So if an example locates in a neighborhood with too many cut edges, this example should be considered *problematic*. Thus, cut edge plays an important role for identifying mislabeled examples. To explore the information of cut edges, SETRED associates every (x_i, \hat{y}_i) in L' with a local cut edge weight statistic J_i defined in Eq. 2.

$$J_i = \sum_{x_j \in N_i} w_{ij} I_{ij} \tag{2}$$

where N_i is the neighborhood of x_i , w_{ij} is the weight on the edge between x_i and x_j and I_{ij} are i.i.d random variables according to the Bernouilli law of parameter $P(y \neq \hat{y}_i)$.

By definition, J_i describes the relationship between the adjacency of an x_i and other vertices in its neighborhood and the fact that they have the same label [17]. Similarly, a null hypothesis H_0 that can be tested with J_i is defined as every examples in $L \cup L'$ is independently labeled according to the marginal distribution $D(Y)$. H_0 specifies a case that the label y_i is assigned to each example x_i without considering any information from x_i , i.e. for any example (x_i, y_i) , the probability of examples in its neighborhood possessing labels other than y_i is expected to be no more than $1 - P(y = \hat{y}_i)$ under H_0 . Hence, a good example will be *incompatible* with H_0 . To test H_0 with J_i , the distribution of J_i under H_0 is need. The distribution of J_i can be approximated to a normal distribution with mean μ_i and variance σ_i^2 estimated by Eq. 3 and Eq. 4, if the size of neighborhood is big and the weights are not too unbalanced, otherwise a simulation must be proceed [7].

$$\mu_{i|H_0} = (1 - P(y = \hat{y}_i)) \sum_{x_j \in N_i} w_{ij} \tag{3}$$

$$\sigma_{i|H_0}^2 = P(y = \hat{y}_i) (1 - P(y = \hat{y}_i)) \sum_{x_j \in N_i} w_{ij}^2 \tag{4}$$

Therefore, if the observation value of J_i that associates with an example (x_i, \hat{y}_i) in L' locates in the left rejection region, then there are significantly less

Table 1. Pseudo-code describing the SETRED algorithm

Algorithm:	SETRED
Input:	the labeled set L , the unlabeled set U , the left rejection threshold θ , the maximum number of iterations M
Output:	the learned hypothesis h
Progress:	<p>Create a pool U' by randomly selecting examples from U $h \leftarrow \text{Learn}(L)$</p> <p>Repeat for M iterations:</p> <p style="padding-left: 2em;">$L' \leftarrow \phi$</p> <p style="padding-left: 2em;">for each possible label y_j do</p> <p style="padding-left: 4em;">h chooses k_j most confident examples from U'</p> <p style="padding-left: 4em;">Add the chosen examples to L' after giving them the label y_j</p> <p style="padding-left: 2em;">Build a neighborhood graph G with $L \cup L'$</p> <p style="padding-left: 2em;">for each $x_i \in L'$ do</p> <p style="padding-left: 4em;">Compute the observation value o_i of J_i</p> <p style="padding-left: 4em;">Find the neighborhood N_i of x_i in G</p> <p style="padding-left: 4em;">Compute the distribution function of J_i under H_0</p> <p style="padding-left: 4em;">if o_i locates in the left rejection region specified by θ</p> <p style="padding-left: 6em;">$L' \leftarrow L' - \{(x_i, \hat{y}_i)\}$</p> <p style="padding-left: 2em;">$h \leftarrow \text{Learn}(L \cup L')$</p> <p style="padding-left: 2em;">Replenish U' by randomly selecting examples from U</p> <p>End of Repeat</p>

cut edges than expected under H_0 , hence it is a *good* example. In contrast, if the observation value locates in places other than the left rejection region, then lots of examples in the neighborhood disagree with its label, hence it could be regarded as a *mislabelled* example. The left rejection region is specified by a pre-set parameter θ .

Once the possibly mislabeled examples in L' are identified, SETRED simply discards those examples, keeping the good ones intact. Consequently, a filtered set L'' is obtained. Note that SETRED does not try to relabel the identified mislabeled examples in order to avoid introducing new noise to the data set. Finally, SETRED finishes the current iteration by relearning a hypothesis on $L \cup L''$. SETRED stops self-training process after the pre-set maximum times of iteration M is reached. The pseudo-code of SETRED is shown in Table 1.

Note that a pool of unlabeled examples smaller than U is used in the algorithm. Blum and Mitchell [1] suggested to choose examples from a smaller pool in stead of the whole unlabeled set. For convenience, we adopt this strategy directly without verification. Furthermore, it is worth noticing that the SETRED could be regarded as a type of active semi-supervised learning algorithm that actively identify the bad examples from the self-labeled set. Absence of the teacher, SETRED just discards the problematic data after identification instead of asking the teacher for labels as in the standard active learning scenario.

4 Experiments

In order to test the performance of SETRED, ten UCI data sets are used. Information on these data sets are tabulated in Table 2.

Table 2. Data set summary

Data set	Size	Attribute	Class	Class distribution
<i>australian</i>	690	15	2	44.5%/55.5%
<i>breast-w</i>	699	9	2	65.5%/34.5%
<i>colic</i>	368	22	2	63.0%/37.0%
<i>diabetes</i>	768	9	2	65.1%/34.9%
<i>german</i>	1000	20	2	70.0%/30.0%
<i>heart-statlog</i>	227	13	2	55.5%/44.5%
<i>hepatitis</i>	155	19	2	20.6%/79.4%
<i>ionosphere</i>	351	34	2	35.9%/64.1%
<i>vehicle</i>	846	18	4	25.1%/25.7%/25.7%/23.5%
<i>wine</i>	178	13	3	33.1%/39.9%/27.0%

For each data set, 25% data are kept aside to evaluate the performance of learned hypothesis, while the remaining 75% data are partitioned into labeled set and unlabeled set under the unlabeled rate 90%, i.e. just 10% (of the 75%) data are used as labeled examples while the remaining 90% (of the 75%) data are used as unlabeled examples. Note that the class distributions in these splits are similar to that in the original data set.

Since SETRED exploits local information to identify mislabeled examples, the learner that utilizes local information is expected to benefit a lot from this method. Therefore in the experiments, the Nearest Neighbor classifier is used as the base learner. Unlike those probabilistic model such as Naive Bayes, whose confidence for an example belonging to a certain class can be measured by the output probability in prediction, the Nearest Neighbor classifier has no explicitly measured confidence for an example. Here for a Nearest Neighbor classifier, the most confidently predicted unlabeled example with label y_j is defined as the unlabeled example which is the nearest to labeled examples with label y_j while far away from those with labels other than y_j . The pre-set parameter θ that specifies the left rejection region of the distribution of J_i is fixed on 0.1, the same as that in [7]. The self-training process stops when either there are no unlabeled examples available or 40 iterations have been done.

For comparison, the standard self-training, namely Self-training, is run on the same labeled/unlabeled/test splits as those used for evaluating SETRED. Same as SETRED, the maximum iteration is also 40. Moreover, two base lines, denoted by NN-L and NN-A respectively, are used for comparison. One is a Nearest Neighbor trained only from the labeled set L , and the other is the one that trained from $L \cup U$ provided the true label of all the examples in U . Note that

Table 3. Average error rate on the experimental data sets (50 runs)

Data set	NN-A	NN-L	SETRED	Self-training	SETRED-imprv.	Self-imprv.
<i>australian</i>	.185	.188	.167	.170	11.3%	9.4%
<i>breast-w</i>	.046	.046	.038	.038	17.9%	16.9%
<i>colic</i>	.194	.237	.191	.209	19.3%	11.8%
<i>diabetes</i>	.298	.330	.320	.335	3.1%	-1.6%
<i>german</i>	.185	.339	.349	.357	-2.8%	-5.2%
<i>heart-statlog</i>	.237	.248	.209	.226	15.8%	8.7%
<i>hepatitis</i>	.161	.186	.208	.157	-11.9%	15.7%
<i>ionosphere</i>	.143	.228	.197	.254	13.6%	-11.4%
<i>vehicle</i>	.298	.412	.399	.413	2.9%	-0.3%
<i>wine</i>	.048	.090	.066	.079	26.6%	12.8%

NN-L is the initial state of both SETRED and Self-training before they utilize any information from the unlabeled examples. NN-A is the ideal state of SETRED and Self-training since every examples chosen in self-training process are given the correct label and all the examples available in the unlabeled set are used.

Experiments are carried out on each data set for 50 runs. In each run, all the four learners are trained and evaluated on the randomly partitioned labeled/unlabeled/test splits. In Table 3, the first four columns are the average error rates of NN-A, NN-L, SETRED and Self-training respectively over 50 runs on each data set. The last two columns denoted by “SETRED-imprv.” and “Self-imprv.” respectively show the performance improvements of SETRED and Self-training over NN-L, which is computed by the error rate of learned hypothesis of SETRED and Self-training over the error rate of the learned hypothesis of NN-L.

Table 3 shows that SETRED benefits much from the unlabeled examples since the performance improvements are evident in 8 data sets, except that it goes worse on *german* and *hepatitis*. The two-tailed paired *t*-test under the significant level of 95% shows that all the improvement of performance are significant. Note that on 4 data sets SETRED performs even better than NN-A which is able to access all the information of the unlabeled examples. In contrast, although the performance of the learned hypothesis of Self-training improves on 6 data sets, only on five the improvements are significant, including *australian*, *breast-w*, *colic*, *heart-statlog* and *hepatitis*. Furthermore, Table 3 also shows that SETRED outperforms Self-training on 9 data sets, among which significance is evident in 6 data sets under a two-tailed pair-wise *t*-test with the significance level of 95%. This evidence supports our claim that SETRED is robust to noise in the self-labeled examples hence achieves better performance than Self-training.

Interestingly, Self-training does benefit from the unlabeled examples on *hepatitis*, while the performance of SETRED degrades. One possible explanation is that SETRED suffers imbalance of the data set. In *hepatitis* data set, there are only 32 positive examples out of 155 examples in all, which is only 20.6% of the total. Recall the method we used for identifying mislabeled examples, one can only be regarded as a good example only if there exists a significantly large num-

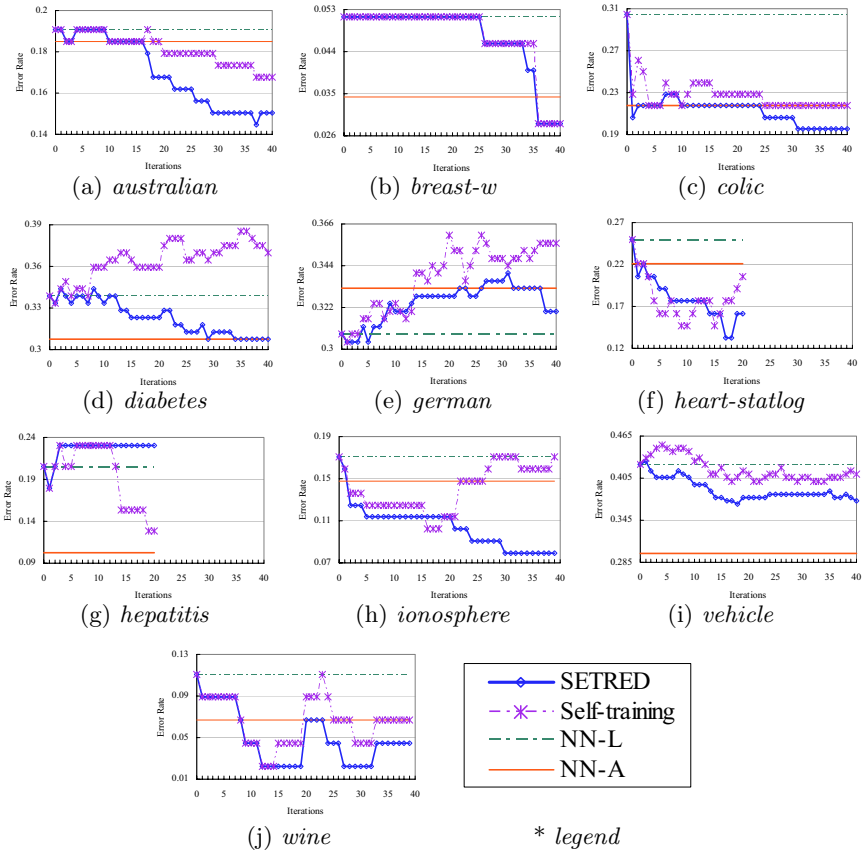


Fig. 1. Median performance on experimental data

ber of examples having the same label in its neighborhood. Since the data set is unbalanced, a correctly labeled positive examples could be easily mis-identified as mislabeled examples and rejected to be added to the labeled set for further training, due to the lack of neighbors possessing the same label. The percentage of the negative examples in the labeled set increases as the self-training process goes on, hence less chance for a correctly labeled positive examples available for further training. The more the distribution of the training set is distorted, the easier for the learner to be misled. Consequently, the performance degrades. Similarly, the error rate of hypothesis learned via SETRED climbs up to 0.349 from the initial error rate of 0.339 on *german*, in which the negative examples are only 30%. The imbalance of this data set might also account for the performance degradation of SETRED.

For further evaluation, one run out of 50 with a median performance on each data set is investigated carefully. Fig. 1 gives plot of error rate versus number of iterations of the median performance on each data set respectively. Note

that SETRED and Self-training stop before the maximum number of iterations is reached on several data set such as *heart-statlog* and *hepatitis*, due to no more unlabeled examples available for further training. In most cases except for *german* and *hepatitis*, SETRED outperforms Self-training, and the error rates of the learned hypothesis by SETRED usually go lower than or converge to the error rates of NN-A. These are consistent with the average performance of the 50 runs on the experimental data sets.

In Fig. 1(g), the error rate curve of SETRED climbs up to a high level after a few iterations and remains unchanged. By contrast, the curve of Self-training drops when many unlabeled examples have been self-labeled and used for further training. This supports the explanation above for the SETRED’s failure that SETRED suffers the imbalance data. Once the correctly labeled positive examples are rejected by SETRED, the misclassified positive examples in the test set, which are probably be correctly classified after more positive examples are learned, will remain being misclassified in the following iterations.

In summary, the experiments show that SETRED can benefit from the unlabeled examples. SETRED is robust to the noise introduced in self-labeling process and its learned hypothesis outperforms that learned via standard self-training.

5 Conclusion

In this paper, a novel self-training style algorithm named SETRED, which incorporates data editing technique to learn actively from the self-labeled examples, is proposed. In detail, SETRED firstly learns from labeled examples and then iteratively chooses to label a few unlabeled examples, on which the learner is most confident in prediction, and adds those self-labeled examples to its labeled set for further training. In each iteration, SETRED does not completely accept all the self-labeled examples that might be highly noisy. Instead, it actively identifies the possibly mislabeled examples from those self-labeled examples by testing a predefined null hypothesis with the local cut edge weight statistic associated with each self-labeled example. If the test indicates a left rejection, the example is regarded as a good example, otherwise it is a possible mislabeled example which should be kept from adding to the learner’s training set. The experiment results on 10 UCI data sets show that SETRED is able to benefit from the information provided by unlabeled examples, and it is robust to the noise introduced in the self-labeling process hence the generalization ability of its learned hypothesis is better than that learned via standard self-training, which is easily affected a lot by those noise. Since SETRED is sensitive to imbalance data, exploring a way to solve this problem will be done in future. Since SETRED uses a Nearest Neighbor as base learner, extending this idea to other base learners will also be future work.

It is noteworthy that the SETRED algorithm is not only a robust self-training style algorithm but shedding a light on a possible way to handle the noise introduced in the learning process by incorporating an active-learning-like technique to refine the self-labeled examples in semi-supervised learning scenario, hence obtaining better performance of the learned hypothesis. Different from others,

the work is done when the teacher to assign labels to the problematic examples is absent. In the future work, theoretical verification of this method will be done, which might help to understand the functionality of this method. Moreover, extending this method to classic semi-supervised learning algorithms, such as co-training [1], or searching for more suitable active learning methods for those algorithms to improve their performance will also be the future work.

Acknowledgement

This work was supported by the National Science Foundation of China under the Grant No. 60473046, the Jiangsu Science Foundation Key Project under the Grant No. BK2004001, the Foundation for the Author of National Excellent Doctoral Dissertation of China under the Grant No. 200343, and the Graduate Innovation Program of Jiangsu Province.

References

1. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory, New York, NY (1998) 92–100
2. Cohen, I., Cozman, F.G., Sebe, N., Cirelo, M.C., Huang, T.S.: Semisupervised learning of classifier: theory, algorithms, and their application to human-computer interaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26** (2004) 1553–1567
3. Jiang, Y., Zhou, Z.-H.: Editing training data for k NN classifiers with neural network ensemble. In: Yin, F., Wang, J., Guo, C. (eds.): *Lecture Notes in Computer Science*, Vol.3173. Springer, Berlin (2004) 356–361
4. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the 16th International Conference on Machine Learning, San Francisco, CA (1999) 200–209
5. Lewis, D., Gale, W.: A sequential algorithm for training text classifiers. In: Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval, Dublin, Ireland (1994) 3–12
6. McCallum, A., Nigam, K.: Employing EM in pool-based active learning for text classification. In: Proceedings of the 15th International Conference on Machine Learning, Madison, WI (1998) 359–367
7. Muhlenbach, F., Lallich, S., Zighed, D.A.: Identifying and handling mislabelled instances. *Journal of Intelligent Information Systems* **39** (2004) 89–109
8. Muslea, I., Minton, S., Knoblock, C.A.: Selective sampling with redundant views. In: Proceeding of the 17th International Conference on Machine Learning, Stanford, CA (2000) 621–626
9. Muslea, I., Minton, S., Knoblock, C.A.: Active + semi-supervised learning = robust multi-view learning. In: Proceeding of the 19th International Conference on Machine Learning, Sydney, Australia (2002) 435–442
10. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: Proceedings of the 9th International Conference on Information and Knowledge Management, Washington, DC (2000) 86–93

11. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. *Machine Learning* **39** (2000) 103–134
12. Sarkar, A.: Applying co-training methods to statistical parsing. In: *Proceedings of the 2nd Annual Meeting of the North American Chapter of the Association of Computational Linguistics*. Pittsburgh, PA (2001) 95–102
13. Seeger, M.: Learning with labeled and unlabeled data. Technical Report, University of Edinburgh, Edinburgh, UK (2001)
14. Seuong, H., Opper, M., Sompolinski, H.: Query by committee. In: *Proceedings of the 5th ACM Workshop on Computational Learning Theory*, Pittsburgh, PA (1992) 287–294
15. Wilson, D.R.: Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics*, **2** (1972) 408–421
16. Zhou, Z.-H., Chen, K.-J., Jiang, Y.: Exploiting unlabeled data in content-based image retrieval. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) *Lecture Notes in Artificial Intelligence*, Vol.3201. Springer, Berlin (2004) 525–536
17. Zighed, D.A., Lallich, S., Muhlenbach, F.: Separability index in supervised learning. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) *Lecture Notes in Artificial Intelligence*, Vol.2431. Springer, Berlin (2002) 475–487