A Novel Inductive Semi-supervised SVM with Graph-Based Self-training

ShengJun Cheng, QingCheng Huang, JiaFeng Liu, and XiangLong Tang

Department of Computer Science and Technology
Harbin Institute of Technology
Harbin, China
{chengsj,huangqc,jefferyliu,tangxl}@hit.edu.cn

Abstract. In this paper, a novel inductive support vector machine for semi-supervised learning, named IS3VM, is proposed, which aims to improve SVM by bootstrapping unlabeled data with self-training. The SVM classifier is iteratively refined through the augmentation of the training set. An improved self-training method is given by employing neighborhood graph for guarantying the reliability of newly added training examples. 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 improved self-training is beneficial and the proposed IS3VM algorithm can effectively exploit unlabeled data to achieve better performance, and is comparable to the-state-of-the-art semi-supervised SVM.

Keywords: Semi-supervised learning, Graph-based method, Self-training, Support vector machine.

1 Introduction

For many practical classification applications, labeled data can be difficult to obtain, but there may exist enormous amount of unlabeled data which is readily available. In recent years, semi-supervised learning has received considerable attention due to its potential for reducing the effort of labeling data. Existing semi-supervised learning can be categorized into several paradigms [1], including generative models (EM), Transductive SVMs, graph-based approaches and bootstrap learning.

Bootstrap learning [2] is referred to as a learner bootstraps from unlabeled data in order to augment the training data set. Self-training [3] is probably the most simple semi-supervised learning algorithm, which is characterized by the fact that the learning process uses its own predictions to teach itself. The assumption of self-training is that its own predictions, at least the high confidence ones, tend to be correct. This is likely to be the case when the classes form well-separated clusters. The major advantages of self-training are its simplicity and the fact that it is a *wrapper* method. Self-training improves the classification margin by selecting the unlabeled examples with the highest classification confidence, and assigns them the class labels that are predicted by simply the current classifier using its posteriori

J. Yang, F. Fang, and C. Sun (Eds.): IScIDE 2012, LNCS 7751, pp. 82–89, 2013.
© Springer-Verlag Berlin Heidelberg 2013

probability outputs. The assigned labels are hereafter referred to as the *pseudo-labels.* The labeled data, along with the selected pseudo-labeled data are utilized in the next iteration for updating the classifiers parameters. This strategy is also well-applied in the famous co-training [4]. However, a problem with this strategy is that the introduction of examples with predicted class labels may only help to increase the classification margin, without actually providing any novel information to the classifier. Since the selected unlabeled examples are the ones that can be classified confidently, they often are far away from the decision boundary. As a consequence, adding these examples to the training set may not help improving the decision boundary; this is because by adjusting the decision boundary, the examples with high classification confidence will gain even higher confidence [5]. Besides, the pseudo-labels with high classification confidence may not be the ground-truth, since the confidence is estimated based on the current classifier and training set [6]. The estimated information may therefore be biased or distorted, thus, it is necessary to provide some means to escape from the distortion or bias from the current classifier. This implies that we may need alternative strategy other than only using classifiers' posteriori probability as the confidence measurement.

TSVM, also called as S3VM, is a popular method for employing SVM in the semi-supervised settings. This approach was introduced by Bennet&Demiriz [7]. S3VM reformulates the original definition by adding two constraints to the unlabeled examples. Considering a binary SVM, one constraint calculates the misclassification error as if the instance were in class 1 and the second constraint as if the instance were in class -1. S3VM tries to minimize these two possible misclassification errors [8]. The labeling with the smallest error is the final labeling. Moreover, TSVM is non-convex and finding its exact solution is NP-hard, several approximation algorithms have been established. However, when the size of test data set is big (e.g. larger than 1000), TSVM type algorithms are still time-consuming [3].

Besides TSVM, support vector machines are incorporated into semi-supervised settings in a different way. In several other studies, a multi-view co-training support vector machine and its variants were presented. For text classification, experiments have clearly shown that the co-training SVM outperforms the co-training Naive Bayes [9] Compared with TSVM algorithms, the computational burden of the co-training support vector machine is much lower.

In this paper, a novel inductive support vector machine for semi-supervised learning, named IS3VM, is proposed, which exploits the unlabeled data by leveraging SVM and a modified self-training. IS3VM not only does not require redundant views like co-training SVM, but also is more computational convenient than TSVM, since it does not need solve the optimization problem. By pseudo-labeling the unlabeled example with high classification confidence, IS3VM can be improved through the augmentation of the training set. This setting tackles the problem of determining how to label the unlabeled examples, which contributes much to the efficiency of the algorithm. Moreover, high reliability of the pseudo-label of the unlabeled example can be achieved through combining self-training with the local *cut edge weight* statistic [10] from the constructed neighborhood graph. Experiments on UCI data sets show that, the improved self-training is beneficial and the proposed IS3VM algorithm can effectively exploit unlabeled data to achieve better generalization performance, and is comparable to the-state-of-the-art semi-supervised SVM methods.

The remainder of this paper is organized as follows: Section 2 presents the graph-based self-training; Section 3 presents the proposed inductive S3VMs; Section 4 reports on the experiments on UCI data sets; Finally, Section 5 concludes and raises several issues for future work.

2 Graph-Based Self-training

Let X be the input space and $Y \in \{0,1\}$ be the output space. Suppose we have a small labeled training set $L = \{(x_i, y_i) \mid i = 1,2,...,l\}$ a large unlabeled set $U = \{x_j \mid j = l+1,...u, u >> l\}$.

First of all, a neighborhood graph is constructed from $L \cup U$, U is pseudo-labeled by the current classifier. The neighborhood graph which conveys the local information from all the examples in the training set, is conducted by the k-nearest neighbor criterion. In the graph, every example represent a vertex, and there exists an edge between two vertices *a* and *b* if either *a* or *b* is among the k nearest neighbor of the other. In this way, one example is *not only* related to its own neighbors, *but also* related to those ones which regard it as their neighbors. Furthermore, a weight $\omega_{ab} \in [0,1]$ is associated with the edge connecting a and b, which is computed as $(1+dist(a,b))^{-1}$, where $dist(a,b)$ corresponds to the distance between a and b. In this paper distance is measured by the *EUCLIDEAN* distance.

After the graph is constructed, we evaluate the confidence of each pseudo-label being correct by employing the cutting edge technique. An edge in the graph is called a *cut edge* if the two vertices connected by it have different associated labels[11]. Intuitively, this coincides with the *manifold assumption* that examples with high similarity in the input space would also have high similarity in the output space. The basic assumption is that a correctly labeled example should possess the same label to most of its neighboring examples. Thus, the pseudo-label confidence of every x_i in U can be measured based on the following *cut edge weight statistic:*

$$J_i = \sum_{xj \in Ne_i} \omega_{ij} I_{ij} \tag{1}$$

where Ne_i is the neighborhood of x_i, ω_{ij} is the weight on the edge between x_i and x_j, I_{ij} are i.i.d random variables according to the Bernouilli law of parameter $P(y \neq \hat{y}_i)$, $\hat{y}_i$ is the pseudo-label of x_i produced by the current classifier. Let $H0$ be the null hypothesis that vertices of the graph are labeled independently according to distribution $D(Y) = \{\Pr(y=1), \Pr(y=0)\}$. Here, $\Pr(y=1)(\Pr(y=0))$ denotes the prior probability of an example being positive (negative), which is usually estimated as the fraction of positive (negative) examples. 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 by a normal distribution with mean and variance estimated by Eq.1 Recall the manifold assumption encoded in the neighborhood graph, correctly labeled examples tend to have few cut edges as its label should be consistent with most of its connected examples. Hence, the smaller the

value of J_i, the higher the confidence of the pseudo-label $\hat{y}_i$ being correct. Therefore, we can select the candidate examples from U by the *cut edge weight statistic* J_i. The proposed method adds the examples whose neighbors have less cut edges to the training set. Instead of directing employing classifiers' predictions to teach itself, we accomplish this goal with an explicitly way, using the cut edge strategy to acquire more reliable candidates with high labeling confidence.

3 The Proposed Inductive IS3VM

In this section, we first present the steps of the proposed algorithm and conceive a method for choosing the appropriate set for parameter C in the SVM formula.

A standard SVM classifier for two-class problem can be defined as:

$$\min \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \tag{2}$$

s.t. $y_i(w^T x_i + b) \geq 1-\xi_i, \quad \xi_i \geq 0, \quad i = 1,...,N,$ where $x_i \in R^n$ is a feature vector of a training sample, $y_i \in \{-1,1\}$ is the label of x_i, $C > 0$ is a regularization constant. In the following paragraph, we will give the algorithm sketch of the proposed IS3VM.

IS3VM algorithm:

Initialize: binary classification problem: $D = L \cup U$

$L = \{(x_i, y_i) \mid i = 1,2,...,l\}$: Labeled training set,

$U = \{x_j \mid j = l+1,...u, u >> l\}$: Unlabeled set

SVM model with parameters: w , ξ , b

$L' \leftarrow L$, $U' \leftarrow U$

k =0

Step1 Train a SVM classifier f on L', perform classification on U', adding pseudo-labels to all the example in U'.The parameters of f are denoted as $w^0 \in R^n$, $\xi^0 \in R^l$ and b^0

Loop until $\mid f(w^k, \xi^k) - f(w^{k-1}, \xi^{k-1}) \mid < \delta_0$

Step2 Construct a neighbor graph based on $L' \cup U'$, for each $x_i \in U'$,compute its cut edge statistic J_i ,using Eq.0

Step3 Select 10% candidate examples from U' with the smallest J_i, associated by the corresponding pseudo-label, form a subset L'' , $L' \leftarrow L' \cup L''$

Step4 Update f using L', the SVM parameters are denoted as $w^k \in R^n$, $\xi^k \in R^{l+u}$, b^k

Step5 Calculate the object function value in (1), $f(w^k, \xi^k) = \frac{1}{2}\left\|w^k\right\| + C\sum_{i=1}^{l+u}\xi_i^k$

Go to Loop

In [12], a standard EM algorithm with a naive Bayesian classifier was analyzed, which is a special case of self-training. Furthermore, the EM algorithm is convergent since the objective function of this algorithm monotonically increases during its iterations. The proposed algorithm has a similar working mechanism to the EM algorithm although the objective functions and classifiers of these two algorithms are different. Generally, if the distribution of the data is Gaussian (or close to Gaussian) and the dimension of the data is not very high, the EM algorithm may be used for classification; otherwise, the proposed IS3VM might achieve better results.

IS3VM can be regarded as a bridge between inductive models and transductive models. Our method takes advantage of both cluster assumption and manifold assumption, which can complement each other. We utilized a graph-based self-training method to give more reliable pseudo-labels to the unlabeled examples instead of the standard self-training which selects candidates only by the current classifiers' output posteriori probability. Especially, when the labeled examples are sparse, the current classifier is not strong enough to provide reliable predictions. In this case, the graph-based self-training method used in our algorithm can exhibit high advantage. Moreover, since IS3VM is more like a bootstrapping, the computational complexity is much lower than TSVM, which is a NP-hard problem, requiring a approximate optimization [14].

4 Experiments

In this section, we design experiments to verify the efficacy of the proposed IS3VM. 15 UCI data sets are used in the experiments. The characteristics of each dataset are shown in Table 1. Our experiments are configured as follows. For each data set, about 25% data are kept as test examples. 25% of the remaining data set is used as the labeled training set L; and all the other examples are treated as the unlabeled set U.

Table 1. The characteristics of 15 UCI datasets

DataSet	Size	Attributes	Class	DataSet	Size	Attributes	Class
anneal	898	39	2	*ionosphere*	351	34	2
australian	690	15	2	*kr-vs-kp*	3916	36	2
breast-c	286	9	2	*segement*	2310	20	2
bupa	345	6	2	*sick*	3772	29	2
colic	368	22	2	*vehicle*	846	19	2
diabetes	768	8	2	*vote*	435	17	2
german	1000	20	2	*wdbc*	569	30	2
hypothyroid	3163	25	2				

The performance of IS3VM is compared with three algorithms, i.e. supervised SVM, self-SVM, TSVM. Supervised SVM is referred to as training a SVM classifier barely on the initial labeled training set, which is equivalent to other algorithms' initial. Self-SVM is based on the standard self-training algorithm, wherein SVM is used as the underlying classifier. TSVM is implemented as the SVM^{light}. *LIBSVM*[13] is used for all the other algorithms. Parameters are set as follows: The kernel trick is

RBF; for fair comparison, 50 examples are selected per round; maximal number of iterations is set to 50. The accuracy score is used to evaluate the performances of algorithms. In our experiments, the accuracy scores of each algorithm are obtained via 10runs of ten-fold cross-validation and evaluated on the same test set. Finally, we conduct two-tailed t-test with a 95% confidence level to compare the proposed algorithm to the others. The results are shown in Table 2.

Table 2. Average accuracy on 15 UCI data sets

Dataset	IS3VM	SVM	Self-SVM	TSVM
anneal	**85.36**	76.24	79.48	80.19
australian	78.83	65.91	78.5	79.87
breast-c	**72.25**	65.94	71.63	70.97
bupa	82.55	76.33	79.45	**85.87**
colic	68.42	**70.58**	66.23	64.38
diabetes	**71.66**	71.9	69.37	70.75
german	**84.64**	78.69	80.37	82.34
hypothyroid	68.54	75.44	71.31	**76.85**
ionosphere	**84.64**	80.34	80.6	82.73
kr-vs-kp	76.56	68.44	72.43	**81.35**
segement	**87.99**	85.77	85.43	78.44
sick	81.67	85.65	84.45	**86.23**
vehicle	81.45	77.4	83.47	**85.4**
vote	68.19	65.37	66.82	**69.38**
wdbc	**84.36**	79.43	81.65	80.12
w/t/l		13/0/2	11/4/0	7/4/4

The two-tailed t-test results are shown in the bottom row, where each entry has the format of *w/t/l*. This means that, comparing with IS3VM, the algorithm in the corresponding column wins *w* times, ties *t* times, and loses *l* times. Table 1 shows that IS3VM can effectively exploit unlabeled data to boost performance, and is superior to the other compared algorithms, where it wins 13 times and loses 2 times against SVM, wins 11 times and never loses against Self-SVM, wins 7 times and loses 4 times against TSVM.

Note that, although IS3VM achieves lower accuracy on 3 data sets (colic, hypothyroid, sick) than SVM, but on average, IS3VM actually achieves higher classification accuracy. One possible explanation of the degradation is that IS3VM suffers imbalance of the data set. In *hypothyroid* and *sick* data set, positive examples are much less than negative examples. Since the data set is unbalanced, a correctly labeled positive example 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, hence less chance for a correctly labeled positive example available for further training. The more the distribution of the training set is distorted, the easier for the learner to be mislabeled. Consequently, the performance degrades.

Furthermore, Table 2 also shows that IS3VM outperforms Self-SVM on 11 data sets, among which significance is evident in 8 data sets under a two-tailed pair-wise

t-test with the significance level of 95%. This evidence supports our claim that the improved graph-based self-training is beneficial, and IS3VM is robust to noise in the self-labeled examples hence achieves better performance than Self-SVM.

Compared with TSVM, IS3VM achieve higher classification accuracy on 7 data sets under a two-tailed pair-wise *t*-test with the significance level of 95%. This suggests that our proposed method is comparable to the transductive SVM, sometimes even better than TSVM. While TSVM suffers from high computational complexity, our method is quiet computational convenient, especially when the data set has high dimensional attributes.

In summary, the experiments show that IS3VM can benefit from the unlabeled examples. The graph-based self-training used in IS3VM is robust to the noise introduced in self-labeling process and its learned hypothesis outperforms that learned via standard self-training. Moreover, IS3VM is more efficient than the famous TSVM, since it does not need solve the optimization problem.

5 Conclusions and Future Work

In this paper, a novel inductive support vector machine for semi-supervised learning, named IS3VM, is proposed, which aims to improve SVM by bootstrapping unlabeled data with self-training. In detail, during every iteration of self-training, the SVM classifier is refined through the augmentation of the training set. An improved self-training method is given by employing neighborhood graph for guarantying the reliability of newly added training examples. Specifically, 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. The experiment results on 15 UCI data sets show that IS3VM is able to benefit from the unlabeled examples, and the proposed graph-based self-training is able to provide more reliable pseudo-labels of the unlabeled examples. Since IS3VM is sensitive to imbalance data, exploring a way to solve this problem will be investigated in future.

In the future, we will combine our method with active learning for the purpose of obtaining better performance of the learned hypothesis. Theoretical verification of this method will be done, which might help to understand the functionality of this method. Moreover, it will also be interesting to apply IS3VM algorithm to real world applications, especially for the applications suitable for semi-supervised learning, such as natural language processing (NLP) and bioinformatics.

Acknowledgements. This paper is supported by the National Science Foundation of China (NSFC) under the Grant No. 61173087 and NO. 61073128.

References

1. Chapelle, O., Schölkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge (2006)
2. Zhu, X.: Semi-supervised learning literature survey. Department of Computer Science, University of Wisconsin at Madison, Madison, WI, Tech. Rep. 1530 (2008)

3. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 189–196 (1995)
4. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory, Wisconsin, MI, pp. 92–100 (1998)
5. Mallapragada, P.K., Jin, R., Jain, A.K., Liu, Y.: SemiBoost: Boosting for Semi-supervised Learning. IEEE Transaction on Pattern Analysis and Machine Intelligence 31(11), 2000–2014 (2009)
6. Li, M., Zhou, Z.-H.: SETRED: Self-training with Editing. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 611–621. Springer, Heidelberg (2005)
7. Bennett, K., Demiriz, A.: Semi-supervised support vector machines. In: NIPS, vol. 11, pp. 368–374 (1999)
8. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of International Conference on Machine Learning (ICML), Bled, Slovenia (1999)
9. Kockelkorn, M., Lüneburg, A., Scheffer, T.: Using Transduction and Multi-view Learning to Answer Emails. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 266–277. Springer, Heidelberg (2003)
10. Muhlenbach, F., Lallich, S., Zighed, D.A.: Identifying and handling mislabeled instances. Journal of Intelligent Information Systems 22, 89–100 (2004)
11. Zhang, M.-L., Zhou, Z.-H.: CoTrade: Confident co-training with data editing. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics 41(6), 1612–1626 (2011)
12. Xu, L., Jordan, M.I.: On convergence properties of the EM algorithm for gaussian mixtures. Neural Computing, 129–151 (1999)
13. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. Optimization Techniques for Semi-Supervised Support Vector Machines
14. Chapelle, O., Sindhwani, V., Keerthi, S.S., Cristianini, N.: Optimization Techniques for Semi-Supervised Support Vector Machines. Journal of Machine Learning Research 9, 203–233 (2008)
15. Asuncion, A., Newman, D.J.: UCI machine learning repository (January 10, 2010), http://archive.ics.uci.edu/ml/datasets.html