Incremental Learning of Support Vector Machines by Classifier Combining

Yi-Min Wen[1,2] and Bao-Liang Lu[1,⋆]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dong Chuan Road, Shanghai 200240, China
{wenyimin; bllu}@sjtu.edu.cn
[2] Hunan Industry Polytechnic, Changsha 410007, China

Abstract. How to acquire new knowledge from new added training data while retaining the knowledge learned before is an important problem for incremental learning. In order to handle this problem, we propose a novel algorithm that enables support vector machines to accommodate new data, including samples that correspond to previously unseen classes, while it retains previously acquired knowledge. Furthermore, our new algorithm does not require access to previously used data during subsequent incremental learning sessions. The proposed algorithm trains a support vector machine that can output posterior probability information once an incremental batch training data is acquired. The outputs of all the resulting support vector machines are simply combined by averaging. Experiments are carried out on three benchmark datasets as well as a real world text categorization task. The experimental results indicate that the proposed algorithm is superior to the traditional incremental learning algorithm, Learn++. Due to the simplicity of the proposed algorithm, it can be used more effectively in practice.

1 Introduction

The brain of human beings has powerful ability of incremental learning. Therefore, how to develop brain-like computing model, how to implement incremental learning is one challenge problem in machine learning research. In real world applications, there are three scenarios need incremental learning: all training data cannot be gathered at one time for the cost of collecting data. As a result the data are acquired batch by batch; some real world applications need instant learning once some training data obtained; all training data cannot be loaded into the memory of computers if the training set is very large. According to Jantke [1], incremental learning is to construct new hypothesis by using only the hypothesis before and the recent information on hand. Zhou and Chen [2] distinguished three kinds of incremental learning tasks: Example-incremental learning

⋆ To whome correspondence should be addressed. This work was supported in part by the National Natural Science Foundation of China under the grants NSFC 60375022 and NSFC 60473040, and the Microsoft Laboratory for Intelligent Computing and Intelligent Systems of Shanghai Jiao Tong University.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 904–911, 2007.

(E-IL); Class-incremental learning (C-IL); and Attribute-incremental learning (A-IL). However, C-IL and A-IL have not been received much attention so far. Syed *et al.* [3] introduced two types of incremental learning methods: instance learning, which uses one example at a time, and block by block learning, which uses a suitable-size subset of samples at a time.

At present, however, the essence of the training algorithms of various kinds of artificial learning systems is an optimization procedure that aims to ensure the generalization ability based on the current learning environment. Therefore, all the current machine learning algorithms don't adapt for incremental learning in nature. The non-adaption lies in that the computation model lacks the ability to get new knowledge or cannot retain the knowledge learned before [4]. The training of artificial neural networks is a gradient descent process, and therefore the modification of connection weights will damage the learned knowledge. The training of SVMs is a global optimization based on all training data. As a result, new added training data will make support vectors change [5].

Classifier combining is a useful method for machine learning [6] [7] [8]. Many scholars have applied classifier combining techniques to incremental learning. Polikar *et al.* proposed Learn++ based on AdaBoost algorithm [9]. Lu and Ichikawa proposed an incremental learning model based on emergence theory [10]. Macek proposed incremental learning algorithms based on bagging and boosting and successfully applied them to EEG data classification [11]. Wang *et al.* used weighted ensemble classifiers to mine concept-drifting data stream [12]. Like bagging, a model of incremental learning by classifier combining (ILbyCC) is proposed in this paper.

2 Incremental Learning by Classifier Combining

2.1 Definition of Batch Incremental Learning

Definition 1. *Given a sequence of training datasets* $S_1, S_2, ..., S_m$*, where* $S_i = \{(x_{i_j}, c_{i_j}) | x_{i_j} \in R^n, c_{i_j} \in L_i \subseteq \{1, 2, ..., k\}, 1 \leq j \leq n_i\}, 1 \leq i \leq m$*.* L_i *indicates the set of class label in training dataset* S_i*. Lets* E_1 *denotes the classifier trained on* S_1*, the batch incremental learning procedure* IL *can be illustrated as:* $IL(S_i, E_{i-1}) = E_i,\ 2 \leq i \leq m.$

In this paper, we only consider the case where the number of class labels don't decrease, i.e., $L_1 \subseteq L_2 \subseteq ... \subseteq L_m$.

ILbyCC takes a frame of modular architecture. Modular architecture can make classifier easy adapt to incremental learning. ILbyCC trains a new classifier on an incremental batch and saves it. All the classifiers trained by far are combined into one combined classifier. The training algorithm of ILbyCC can be illustrated as: $M(f_1, f_2, ..., f_{i-1}, f_i) = E_i$, where M denotes the strategy for classifier combining, and E_i denotes the current combined classifier.

Table 1. The problem statistics and the parameters used in SVMs

Data set	#attributes	#training data	#test data	#class	C	γ
Optical Digits	1024	1200	4420	10	128	0.002
Vehicle Silhouette	18	630	216	4	1500	0.00001
Concentric Circle	2	1200	500	5	128	0.125
Yomiuri News Corpus	5000	424310	87268	9	64	0.125

2.2 Combining Classifiers by Averaged Bayes

Given m classifiers that can output posterior probability information, when a test input x comes, the j-th classifier outputs the posterior probability of x belonging to all the classes:

$$P_j(y = i|x), i \in \{1, 2, ..., k\}, j = 1, 2, ..., m \tag{1}$$

According to Averaged Bayes, the combined classifier E_m computes the posterior probability of x belonging to all classes as follows:

$$P_{E_m}(y = i|x) = \frac{1}{m} \sum_{j=1}^{m} P_j(y = i|x), \ i \in \{1, 2, ..., k\} \tag{2}$$

According to Bayes rule, x can be classified as the i-th class:

$$i = \arg\, max_{i=1}^{i=k} P_{E_m}(y = i|x) \tag{3}$$

2.3 Incremental Learning Algorithm by Classifier Combining

ILbyCC algorithm is described as Fig.1.

3 Experiments

3.1 Datasets

In order to evaluate the performance of ILbyCC algorithm, experiments are run on four data sets. The first three data sets, Optical Digits Database, Vehicle Silhouette Database, and Concentric Circle Database, are took from Poliker's paper [9] and used as Poliker's strategy. The fourth data set is a part of Yomiuri News Corpus database. We select all the instances of nine classes, such as crime, sport, Asian-Pacific, North-South-American, health, accident, by-time, society, and finance, which will be called as class 1 through class 9. The training data set is randomly divided into 9 incremental batches, S_1 through S_9, where S_1 through S_3 have instances from classes 1, 2, and 3; S_4 through S_6 contain instances from classes 1 through 6; and S_7 to S_9 have instances from classes 1 through 9. The statistics of the tasks are illustrated in Table.1. The parameters used in SVMs are selected by cross-validation.

Algorithm: ILbyCC

Input: given two example-incremental learning sequences: $List_1 = \{S_1^1, S_1^2, ..., S_1^m\}$ and $List_2 = \{S_2^1, S_2^2, ..., S_2^n\}$, where $L_1^1 = L_1^2 = ... = L_1^m = L1$, $L_2^1 = L_2^2 = ... = L_2^n = L2$, $L1 \subset L2$. Let $n = 0$, if there is only one example-incremental learning sequence.

Steps:

1. For $t = 1, 2, ..., m$
   (a) Take cross-validation on S_1^t to select the optimal parameters of training algorithm and train a classifier f_1^t on the incremental batch S_1^t.
   (b) Save classifier f_1^t and S_1^t can be discarded.
2. For $t = 1, 2, ..., n$
   (a) Take cross-validation on S_2^t to select the optimal parameters of training algorithm and train a classifier f_2^t on the incremental batch S_2^t.
   (b) Save classifier f_2^t and S_2^t can be discarded.
3. Testing:
   (a) Import a test input x into each f_2^t, $1 \le t \le n$, and calculate the posterior probability of x belonging to all classes: $P_t^j, 1 \le t \le n, j \in L2$.
   (b) Take the rule of classifier combining M to combine f_2^t, $1 \le t \le n$, and get the combined classifier $E_n = M(f_2^1, f_2^2, ..., f_2^n)$, where E_n outputs the posterior probability of x belonging to all classes: $P_{E_n}^j, j \in L2$.
4. If $\arg max_{j \in L2} P_{E_n}^j \in (L2 - L1)$, x can be classified by the value of $\arg max_{j \in (L2-L1)} P_{E_n}^j$. The algorithm ends.
5. If $\arg max_{j \in L2} P_{E_n}^j \in L1$, modify the outputs of E_n by setting $P_{E_n}^j = 0, j \in (L2 - L1)$ and $P_{E_n}^j = \frac{P_{E_n}^j}{\sum_{j \in L1} P_{E_n}^j}, j \in L1$, then take the classifier combining rule M to combine classifiers $\{f_1^1, f_1^2, ..., f_1^m, E_n\}$ and get the combined classifier E. E outputs the posterior probability of x belonging to all classes: $P_E^j, j \in L1$.
6. Classify the test input x by the value of $\arg max_{j \in L1} P_E^j$.
7. The algorithm ends.

Fig. 1. Incremental learning algorithm by classifier combining

In order to test ILbyCC's performance on incremental learning when different incremental step takes different parameters. Optimal parameters in each incremental step were chosen among 25 pairs of (C, γ) by 10-cross-validation. 25 pairs of (C, γ) were generated around the values of (C, γ) in Table.1 by a product factor of 2.

In order to ensure the reliability of the experimental results, the first three experiments were repeated 10 times and averaged results were presented. Only the last experiment was run one time for its large size. In order to evaluate the performance of ILbyCC , several exsiting algorithms were run for a comparison study. We adopted the algorithm of Syed [3] that was denoted as ILbySV for convenience. In addition, the basic incremental learning algorithm is *Batch-training*, i.e. when the i-th incremental batch comes, the classifiers trained before are all discarded and $S_1 \bigcup S_2 \bigcup ... \bigcup S_i$ is used to train a new classifier. Obviously, Batch-training should keep all training data gotten by far, and further,

catastrophic forgetting takes place when new data comes. In order to compare ILbyCC with Learn++, the paper directly quotes the experimental results of Learn++ [9]. For convenience, when all the training sessions of ILbyCC uses the same parameters, ILbyCC is denoted as ILbyCC1, when different session of ILbyCC use different parameters, ILbyCC is denoted as ILbyCC2.

3.2 Results and Analysis

Both Fig. 2 and Fig. 4 show that ILbyCC was able to preserve the knowledge learned before and acquire new information. Fig. 3 and Fig. 5 illustrate that ILbyCC can incrementally learn successfully, ILbyCC1 and ILbyCC2 have nearly the same generalization ability, and ILbyCC is slightly good then Learn++. Because all incremental batches are not always in the same distribution, the incremental learning performance of ILbySV fluctuates.

Fig.6 and Fig.8 show that the generalization performance of ILbyCC first decreases slightly when new classes are introduced and increases when training data with the same class labels are continuously added, indicating that ILbyCC can preserve the learned knowledge. From Fig. 7 and Fig.9, it seems that a large improvement on the performance is obtained after new classes that were not available earlier are introduced, but only minor improvements in the performance can be observed from the test accuracy curves when new classes are not introduced, indicating that ILbyCC can learn from new introduced classes.

In Fig. 10, it can be seen that the training time of ILbyCC is far smaller than the training time of Batch-training and ILbySV. The large speedup of ILbyCC can compensate the slight decrease of its generalization performance compared with Batch-training.

Why can ILbyCC work effectively? According to the theory of bias-variance [13], decomposing training data will introduce bias and makes the generalization ability of single classifier decrease, however, decomposing training data will increase the variances between all classifiers and increase the generalization ability

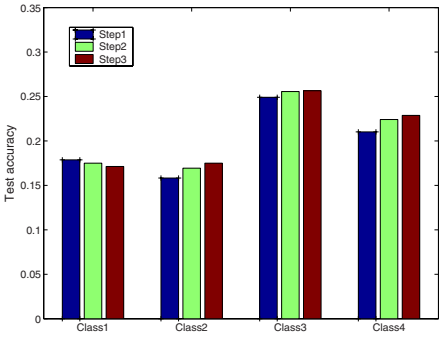

Fig. 2. The generalization performance of ILbyCC1 on each class in Vehicle Silhouette database

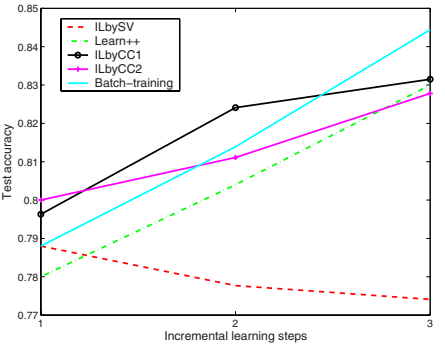

Fig. 3. Accuracy comparison of various incremental learning algorithms on Vehicle Silhouette database

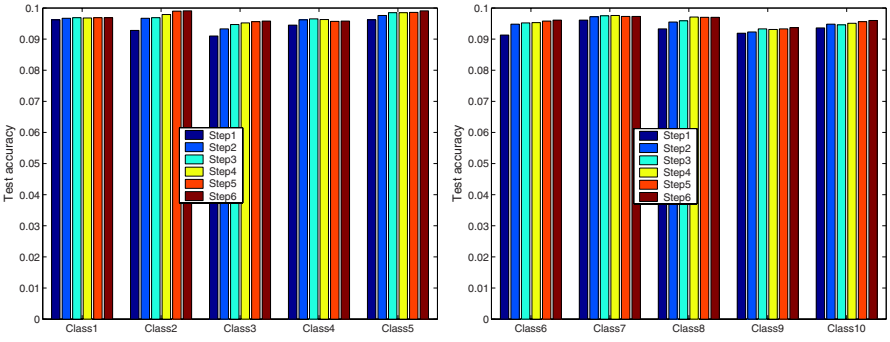

Fig. 4. The generalization performance of ILbyCC1 on each class of Optical digits database

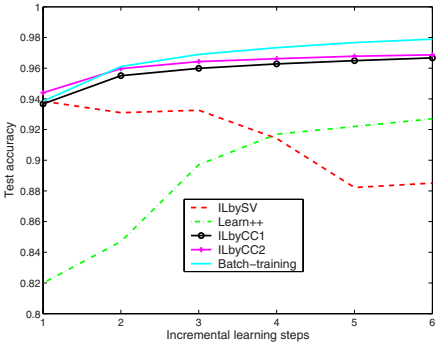

Fig. 5. Accuracy comparison of various incremental learning algorithms on Optical digits database

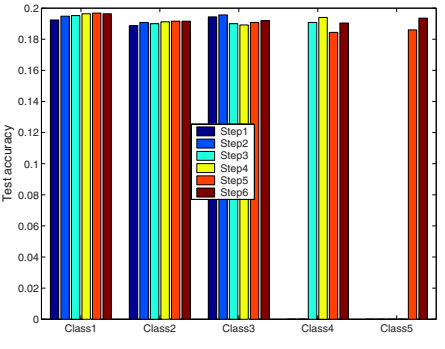

Fig. 6. The generalization performance of ILbyCC1 on each class of Concentric Circle database

of the combined classifier, which compensates the decrease of the generalization ability caused by decomposition. Therefore, ILbyCC has nearly the same test accuracy with Batch-training. In addition, the combining rule (2) can automatically invalidate the classifiers that is not much confident of its outputs, i.e., given $P_j(y = 1|x) \approx ... \approx P_j(y = k|x)$, the result of the equation (3) will not be influenced by the outputs of the j-th classifier. Therefore, Averaged Bayes can automatically select the classifiers that is confident of its outputs to combine.

Note that the performance of ILbyCC1 and ILbyCC2 in all the simulations are nearly the same, it is very interesting to observe that the time complexity for selecting optimal parameters is decreased by training data decomposition. It is not reasonable for incremental learning algorithm to wait for all training data collected to select optimal parameters. It is also not reasonable to apply the parameters, which is gotten from the first incremental batch, to the following incremental steps. Therefore, ILbyCC not only decreases the time complexity of parameter selection but also makes incremental learning possible.

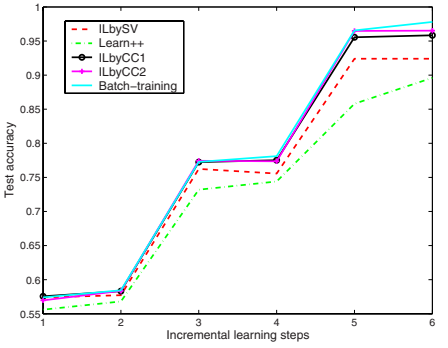

Fig. 7. Accuracy comparison of various incremental learning algorithms on Concentric Circle database

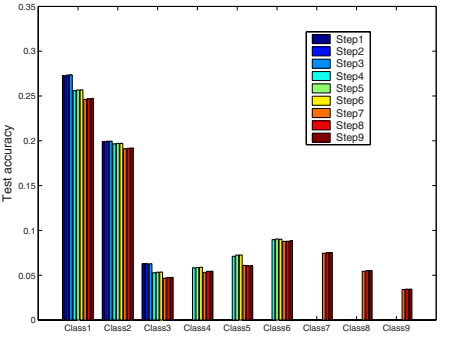

Fig. 8. The generalization performance of ILbyCC1 on each class in Yomiuri News Corpus database

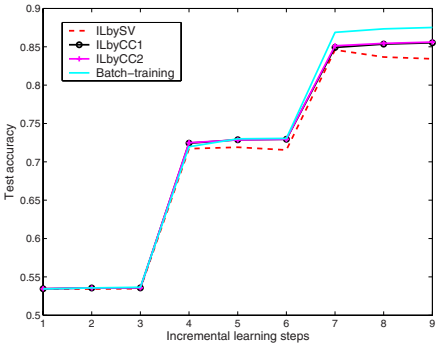

Fig. 9. Accuracy comparison of various incremental learning algorithms on Yomiuri News Corpus database

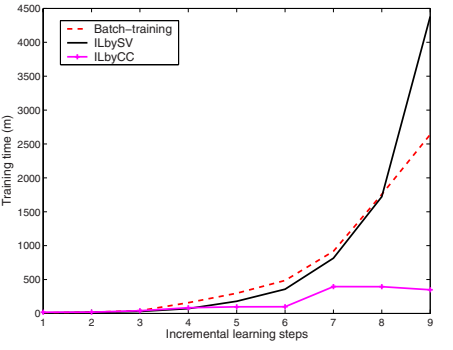

Fig. 10. Comparison of training time on Yomiuri News Corpus database

3.3 Discussions

Compared with Learn++, the proposed ILbyCC satisfies the criteria proposed by Polikar [9] and has comparable incremental learning ability, but ILbyCC can be implemented more simply. Learn++ is a kind of AdaBoost in essence, Learn++ should use more parameters and train more classifiers. Note that ILbyCC is a bagging-like algorithm, ILbyCC can be parallized for training speedup, while Learn++ can only be implemented in serial. In addition, ILbyCC needs no communication between classifiers, it can well protect the privacy of data. The work in this paper can prove the availability of the algorithm estimating the posterior probabilistic of SVMs. To our best knowledge, ILbyCC is the first application to apply posterior probabilistic SVMs to real problem.

4 Conclusions

In this paper, we have proposed a novel incremental learning algorithm ILbyCC that uses Averaged Bayes rule to combine classifiers. The experimental results indicate that ILbyCC can not only preserve the knowledge learned before but also can learn new knowledge from new added data and further new knowledge from new introduced classes. Three main advantages of ILbyCC over existing algorithms are simply implementing, small time complexity for parameter selection, and training time saving. In addition, the proposed algorithm is a general framework of incremental learning and any machine learning algorithm that can output posterior probabilistic can be integrated into ILbyCC.

References

1. Jantke, P.: Types of Incremental Learning. AAAI Symposium on Training Issues in Incremental Learning, March 23-25, Standford CA, 1993
2. Zhou, Z.H. and Chen, Z.Q.: Hybrid Decisions Tree. Knowledge-Based System, 15 (2002) 515-528
3. Syed, N.A., Huan, L., and Sung, K.K.: Handling Concept Drifts in Incremental Learning with Support Vector Machines. In: Proceedings of KDD-99, San Diego, CA,USA, 1999
4. Grossberg, S.: Nonlinear Neural Networks: Principles, Mechanisms and Architectures. Neural Networks, 1 (1988) 17-61
5. R*ü*ping, S.: Incremental Learning with Support Vector Machines. In: Proceedings of the IEEE International Conference on Data Mining, San Jose, CA (2001)
6. Lu, B.L., and Ito, M.: Task Decomposition and Module Combination Based on Class Relations: a Modular Neural Networks for Pattern Classification. IEEE Transaction on Neural Networks, 10 (1999) 1244-1256
7. Zhou, Z.H. and Chen S.F.: Neural Network Ensemble. Chinese J.Computers (in Chinese), 25 (2002) 1-8
8. Xu, L., Krzy*ż*ak, A., and Suen, C.Y.: Methods of Combining Multiple Classifiers and Their Application to Handwriting Recognition. IEEE Transaction on Systems, Man, and Cybernetics, 22 (1992) 418-434
9. Polikar, R., Udpa, L., Udpa, S.S., and Honavar, V.: Learn++: An Incremental Learning Algorithm for Supervised Neural Networks, IEEE Transaction on Systems, Man, and Cybernetics, 31 (2001) 497-508
10. Lu, B.L. and Ichikawa, M.: Emergent Online Learning in Min-max Modular Neural Networks. In: Proceedings of IJCNN'01 (2001) 2650-2655
11. Macek, J.: Incremental Learning of Ensemble Classifiers on ECG data. In: Proceedings of CBMS'05 (2005)
12. Wang, H.X., Fan, W., Yu, P.S., and Han, J.W.: Mining Concept-drifting Data Streams Using Ensemble Classifiers. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (2003)
13. Breiman, L.: Bagging Predictors. Machine Learning, 24 (1996) 123-140