A SELF-TRAINING SEMI-SUPERVISED SUPPORT VECTOR MACHINE ALGORITHM AND ITS APPLICATIONS IN BRAIN COMPUTER INTERFACE

Yuanqing Li, Huiqi Li, Cuntai Guan and Zhengyang Chin

Institute for Infocomm Research, Singapore 119613

ABSTRACT

In this paper, we analyze the convergence of an iterative self-training semi-supervised support vector machine (SVM) algorithm, which is designed for classification in small training data case. This algorithm converges fast and has low computational burden. Its effectiveness is also demonstrated by our data analysis results. Furthermore, we illustrate that this algorithm can be used to significantly reduce training effort and improve adaptability of a brain computer interface (BCI) system, a P300-based speller.

Index Terms— Supporter Vector Machine (SVM), semi-supervised learning, convergence, brain computer interface (BCI), P300.

1. INTRODUCTION

In real-world application, labeling data for training a classifier is often expensive and time-consuming. When small amount of labeled data and a large amount of unlabeled data are available, semi-supervised learning can generally provide us a classifier with satisfactory performance. Thus semi-supervised learning has received considerable attention in recent years. Existing semi-supervised algorithms include expectation maximization (EM) algorithm, self-training algorithms, co-training algorithms [1], transductive support vector machines (TSVMs) [2], etc. Since the optimization problem of a TSVM is non-convex and finding its exact solution is NP-hard, several approximation algorithms have been established [3]-[5]. In several studies e.g. [6], multi-view co-training support vector machine and its variants were presented. To avoid the computational complexity of transductive SVM algorithms, we resort to a self-training SVM algorithm, which is similar to the case of the co-training SVM in [6] when only one view of data is available. The co-training SVM is an incremental algorithm. However, the self-training SVM in this paper is iterative, whose convergence will be addressed in this paper. Our real-world data analysis results will demonstrate the fast convergence and the effectiveness of this algorithm. Furthermore, we will illustrate its application in a brain computer interface (BCI) system, a P300-based speller.

Correspondence to: Dr. Yuanqing Li, E-mail: yqli2@i2r.a-star.edu.sg

2. A SELF-TRAINING SEMI-SUPERVISED SVM

In this section, we present an iterative self-training semi-supervised SVM algorithm, and prove its convergence. A data analysis example is also presented to demonstrate its validity.

First, a standard SVM classifier for two-class problem can be defined as

$$\min \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i, \;\; subject\ to, \qquad (1)$$
$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1-\xi_i,\ \xi_i \geq 0,\ i = 1, \cdots, N,$$

where $\mathbf{x}_i \in R^n$ is a training sample (feature vector), $y_i \in \{-1, 1\}$ is the label of $\mathbf{x}_i$, $i = 1, \cdots, N$. $C > 0$ is a regularization constant.

Algorithm 1

Suppose that we have a small training set F_I containing N_1 samples $\{\mathbf{x}_i, i = 1, \cdots, N_1\}$ with given labels $[y_0(1), \cdots, y_0(N_1)]$, and a test set F_T containing N_2 samples $\{\mathbf{x}_{N_1+i}, i = 1, \cdots, N_2\}$ with their labels unknown.

Step 1 Using F_I, we train a SVM, and perform classification on F_T. The parameters of the SVM are denoted as $\mathbf{w}^{(1)} \in R^n$, $\xi^{(1)} \in R^{N_1}$ and $b^{(1)}$. The predicted labels are denoted as $[y^{(1)}(1), \cdots, y^{(1)}(N_2)]$;.

Step 2 The kth iteration ($k = 2, \cdots$) follows Steps 2.1-2.3.

Step 2.1 Define a new training set as $F_N = F_I + F_T$, where the labels of F_T are predicted in the $(k-1)$th iteration.

Step 2.2 Using the training set F_N, we train a SVM, and perform classification again on F_T. The parameters of the SVM are denoted as $\mathbf{w}^{(k)} \in R^n$, $\xi^{(k)} \in R^{N_1+N_2}$ and $b^{(k)}$. The predicted labels are denoted as $[y^{(k)}(1), \cdots, y^{(k)}(N_2)]$.

Step 2.3 Calculate the objective function value in (1),

$$f(\mathbf{w}^{(k)}, \xi^{(k)}) = \frac{1}{2}||\mathbf{w}^{(k)}||^2 + C\sum_{i=1}^{N_1+N_2}\xi_i^{(k)}. \qquad (2)$$

Step 3 (Termination step) Given that δ_0 is a pre-determined positive constant, if $|f(\mathbf{w}^{(k)}, \xi^{(k)}) - f(\mathbf{w}^{(k-1)}, \xi^{(k-1)})| < \delta_0$, the algorithm stops after the kth iteration, and the predicted labels $[y^{(k)}(1), \cdots, y^{(k)}(N_2)]$ of the test set are the final classification results. Otherwise, perform the $k+1$th iteration.

The following theorem shows the convergence of Alg. 1.

Theorem 1 *For $f(\mathbf{w}^{(k)}, \xi^{(k)})$ defined in (2), we have,*

$$f(\mathbf{w}^{(k-1)}, \xi^{(k-1)}) \geq f(\mathbf{w}^{(k)}, \xi^{(k)}). \quad (3)$$

Proof: According to Step 1 of Alg. 1, $(\mathbf{w}^{(1)}, \xi^{(1)}, b^{(1)})$ is the solution of following optimization problem with training data set F_I,

$$\min \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N_1}\xi_i, \ \ subject\ to, \quad (4)$$
$$y_0(i)(\mathbf{w}^T x_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ i = 1, \cdots, N_1.$$

Note that the predicted labels $[y^{(1)}(1), \cdots, y^{(1)}(N_2)]$ are defined by the following inequalities,

$$y^{(1)}(i)((\mathbf{w}^{(1)})^T \mathbf{x}_{N_1+i} + b^{(1)}) \geq 0, \ i = 1, \cdots, N_2. \quad (5)$$

We now expand the vector $\xi^{(1)} \in R^{N_1+N_2}$ by defining

$$\xi^{(1)}_{N_1+i} = \begin{cases} 0, \text{ if } y^{(1)}(i)((\mathbf{w}^{(1)})^T \mathbf{x}_{N_1+i} + b^{(1)}) \geq 1, \\ 1 - y^{(1)}(i)((\mathbf{w}^{(1)})^T \mathbf{x}_{N_1+i} + b^{(1)}), \text{ otherwise}, \end{cases} \quad (6)$$

where $i = 1, \cdots, N_2$.

Define a label vector $\mathbf{y}^{(1)} = [y_1^{(1)}, \cdots, y_{N_1+N_2}^{(1)}] = [y_0(1), \cdots, y_0(N_1), y^{(1)}(1), \cdots, y^{(1)}(N_2)]$. For the second iteration of Alg. 1, we can find that $(\mathbf{w}^{(2)}, \xi^{(2)}, b^{(2)})$ is the solution of the following optimization problem with training data set $F_I + F_T$,

$$\frac{1}{2}\min ||\mathbf{w}||^2 + C\sum_{i=1}^{N_1+N_2}\xi_i, \ \ subject\ to, \quad (7)$$
$$y_i^{(1)}(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ i = 1, \cdots, N_1 + N_2.$$

From (5) and (6), $(\mathbf{w}^{(1)}, \xi^{(1)}, b^{(1)})$ is a feasible solution of (7). Since $(\mathbf{w}^{(2)}, \xi^{(2)}, b^{(2)})$ is an optimal solution of (7), thus we have

$$f(\mathbf{w}^{(1)}, \xi^{(1)}) \geq f(\mathbf{w}^{(2)}, \xi^{(2)}). \quad (8)$$

Similarly, $(\mathbf{w}^{(k-1)}, \xi^{(k-1)}, b^{(k-1)})$ $(k > 2)$ is the solution of following optimization problem, where label vector $\mathbf{y}^{(k-2)} = [y_0(1), \cdots, y_0(N_1), y^{(k-2)}(1), \cdots, y^{(k-2)}(N_2)]$,

$$\min \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N_1+N_2}\xi_i, \ \ subject\ to, \quad (9)$$
$$y_i^{(k-2)}(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i,$$
$$\xi_i \geq 0, \ i = 1, \cdots, N_1 + N_2,$$

and $(\mathbf{w}^{(k)}, \xi^{(k)}, b^{(k)})$ is the solution of following optimization problem,

$$\min \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N_1+N_2}\xi_i, \ \ subject\ to, \quad (10)$$
$$y_i^{(k-1)}(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i,$$
$$\xi_i \geq 0, \ i = 1, \cdots, N_1 + N_2.$$

Through considering the two cases: i. $y_i^{(k-2)} = y_i^{(k-1)}$, ii. $y_i^{(k-2)} \neq y_i^{(k-1)}$, and the definition of $y_i^{(k-1)}$, we can prove that $(\mathbf{w}^{(k-1)}, \xi^{(k-1)}, b^{(k-1)})$ satisfies the constraints of (10), i.e. it is a feasible solution of (10). Noting that $(\mathbf{w}^{(k)}, \xi^{(k)}, b^{(k)})$ is the optimal solution of (10), hence we have the inequality in (3). The theorem is proven. ♯

Remark 1: i. Since $f(\mathbf{w}^{(k)}, \xi^{(k)}) \geq 0$, it follows from Theorem 1 that $\{f(\mathbf{w}^{(k)}, \xi^{(k)})\}$ is convergent. Thus Alg. 1 is convergent. ii. The objective function in (1) can be explained as a structural risk. According to Theorem 1, the structural risk decreases as the iterations of Alg. 1. This generally leads to increased classification accuracy.

Example 1: In this example, we demonstrate the validity and convergence of Alg. 1 by real-world data analysis. We apply Alg. Algorithm 1 to 4 real-world data sets "Diabetes," "Wla," "Cancer" [7]. The number of examples of the four real data sets are $768, 2000, 2200$ and 680 respectively. For each data set, we perform a 5-fold cross-validation. In each fold, the data set is divided into three parts, the first is called the initial training data set, the second is the test data set which is used in retraining, the third is an independent test set for further validation. The sizes of the initial training data sets for the four real data sets are arbitrarily set to $100, 45, 18$ and 100 respectively. For each real data set, the ratio of the sizes of test data set and the independent test set is $4:1$.

We apply Alg. 1 to each of the 4 real-world data sets. We obtain the accuracy rates for each of the 5 folds of the test set and the independent test set. Thus there are a total of 10 accuracy rates, which are then averaged. The average predication accuracy rate for each real-world data set is listed in the $4th$ column in Table 1. For the purpose of comparison, we use the EM algorithm to replace Alg. 1 and perform a similar analysis for each of the 4 real-world data sets. 10 accuracy rates are obtained and then averaged. The average predication accuracy rate is listed in the $3th$ column in Table 1. Using T-test, we further compare the 10 accuracy rates obtained by Alg. 1 and the 10 accuracy rates obtained by EM algorithm. The corresponding P value is listed in the 5th column.

Table 1. Analysis results (accuracy rates %) for three data sets

Data	Dimension	EM	Alg. 1	P values
Dim	14	67.4	89.3	0.02
Dia	8	69.8	77.8	0.01
Cancer	10	96.7	96.4	0.69
Wla	180	84.9	85.1	0.95)

From statistical tests, the performance of Alg. 1 is significantly better than that of EM algorithm for the first 2 data sets. For the last 2 data sets, the is no significant difference between the performance of Alg. 1 and that of EM algorithm.

To demonstrate the convergence of Alg. 1, we show two

curves of SVM objective function (structural risk) in Fig. 1, which are obtained in two of the 5-folds cross-validation for data set 'Dia'. The two curves converge to lower limits. The decrease of structural risk generally leads to a higher classification accuracy. This also explains the effectiveness of Alg. 1.

The are two main advantages of Alg. 1: i. As shown in Fig. 1, this iterative algorithm converges fast (it generally converges by 10 iterations); (2) Compared to a typical TSVM algorithm, it is not NP-hard in computational complexity.

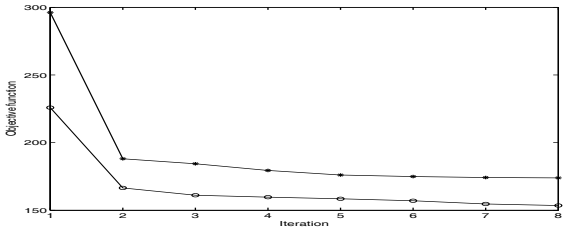

Fig. 1. Decreasing tendency of structural risk.

3. APPLICATION IN A BCI SYSTEM

In this section, we illustrate the application of Alg. 1 in a P300-based BCI speller.

BCIs provide an alternative communication and control channel to convey messages and commands from brain to the external world without using nerves and muscles [8]. Currently, the electroencephalogram (EEG) is the most prevailing brain signal for non-invasive BCIs. One robust feature is a positive displacement of EEG amplitude (event related potential) occurring around 300ms after stimulus. This is also known as the P300 [10]. P300 is a common feature in BCIs. Based on P300, a speller was developed in [9]. The main issue in P300-based BCIs is the robust classify of the P300 response and the background noise. However, a generally tedious and time consuming training process is needed for P300-based BCIs in order to build a reliable classification model for each subject. It is essential to reduce training effort so that P300-based BCI can be convenient to use.

In the following data analysis, we will show that Alg. 1 can be used to reduce training effort in a P300-based speller. The dataset employed in this example was collected by a P300-based speller paradigm described in [10]. During the experiment, a 6-by-6 matrix that includes characters and numbers is presented to the user on a computer screen (Fig. 2). Each row or column of the matrix is intensified successively in a random order. Each intensification lasts for 100 ms followed by a 75 ms break. The user focuses on his desired character or number. Each run consists of 12 intensifications which covers all the rows and columns of the matrix. Two of these twelve consecutive intensifications in each run contain the desired symbol, where P300 potential is generated. For each symbol, 10 runs of twelve intensifications are carried out. The dataset contains training data and testing data collected from 10 subjects. The phrase with 41 characters "THE QUICK BROWN FOX JUMPS OVER LAZY DOG 246138 579" is used for data collection. The same phrase is also used to testing data collection with random word order. In our data analysis, the

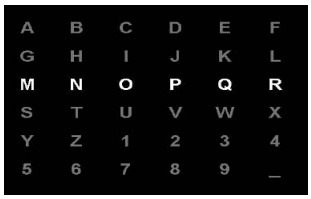

Fig. 2. The stimulus matrix shown on a computer monitor to the user. One row or one column of the matrix is intensified.

original test data is used as an independent test set, which is not used for retraining in Alg. 1. Algorithm 1. The data corresponding to the first three characters original training data set is used for the initial training data set. The other 38 characters are used for retraining and testing testing for Algorithm 1. Lowpass filtering is first performed on the EEG data. Next, the data segment from 24 EEG channels, between 150 ms to 500 ms from the start of each intensification, is selected for constructing the feature. The accuracy rates averaged over 10 subjects obtained by Algorithm 1 are given in row 2 of Table 2. The accuracy rates in row 3 are given for a standard SVM, which also uses the first three characters for training and the characters are used for testing only. The accuracy rate in row 4 is given for a standard SVM, which uses all the 41 characters for training. Since none of these characters are used for test, no accuracy rate is given for the test dataset in row 4. It can be It can be seen from Table 2 that: Alg. 1 can obtain a comparable accuracy rate (98.8%) as the standard SVM (99.0%), however the the initial training set (3 characters) of Alg. 1 is much smaller than that of the standard SVM (41 characters). Thus the training data collection time (training time) of the BCI speller can be reduced much while the accuracy is not affected

Table 2. Accuracy rates (%) for a data set from a P300-based BCI speller

Alg. 1 (3 training symbols)	95.8	98.8
	Test dataset	Ind. test dataset
Standard SVM (3 training symbols)	80.8	83.9
Standard SVM (41 training symbols)	no accuracy	99.0

In the above offline analysis, we use the data of 38 characters (without true labels) for retraining in Alg. 1. The left subplot of Fig. 3 shows the curves of average accuracy rates,

for the test data set and the independent test data set, obtained after each after each iteration of Algorithm 1.

We also simulate online data analysis. We use an incremental version of Alg. 1. First, we use the data of the first 3 characters to train a SVM. The data of the subsequent 10 characters ($4th$ to 13th characters) are then classified. Using the initial training data set and the data of these 10 characters with predicted labels, we retrain a new SVM using Alg. Algorithm 1 and classify the next 10 characters ($14th$ to characters) and so on. We stop retraining the SVM model after all the 38 characters are used. The independent test set are then classified by the finalized SVM model. The curve of average accuracy for the simulated online case is depicted by the solid line with "*" in the right subplot of Fig. 3. The dashed line with "o" is the average accuracy curve for the standard SVM, which uses all the first 41 characters for training as in our original experiment. Note that these 41 characters are not classified in the simulated online case, i.e. 0 accuracy rate. The prediction accuracy rates of the last 41 characters obtained by Alg. 1 and the standard SVM are almost equal in the simulated online case. Using an incremental version of Alg. 1, we can start classification for incoming data much earlier than the standard SVM while keeping a satisfactory accuracy.

Furthermore, retraining of the SVM can be restarted using the incoming data when the performance of the system is not satisfactory. Therefore, we can also use Alg. 1 to improve the adaptability of the BCI system.

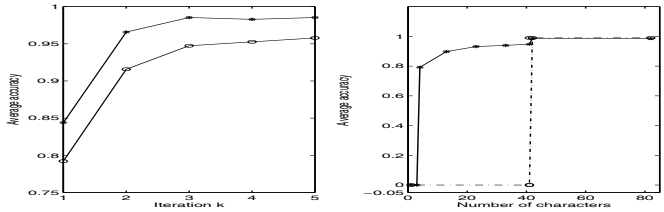

Fig. 3. Left: Accuracy curves obtained after each iteration of Alg. 1 (the curve with 'o' is for the test data set, while the curve with '*' for the independent test set. Right: Solid line with '*' is the accuracy curve obtained by an incremental version of Alg. 1 which can be used in online cases. Dashed line with 'o' is the accuracy curve obtained by the standard SVM which uses all the first 41 characters for training as in our original experiment.

Remark 2: We also use EM algorithm to replace Alg. 1 in the above offline data analysis and find that EM algorithm does not work here. The reasons are: i. The dimension of feature vectors is large (408) and the training data set samples are very few; ii. The score outputs (posterior probabilities) from EM algorithm are not so reliable as the scores from SVM. These scores are used to determine the output (character) of the system.

4. CONCLUSION

In this paper, we present a self-training semi-supervised SVM algorithm and prove its convergence. By comparing with EM algorithm in several real-world datasets, the effectiveness and fast convergence of this algorithm are demonstrated. Finally, an application of the algorithm in a P300-based BCI speller is illustrated. This algorithm can be used to reduce training effort and improve adaptability of the P300-based BCI speller.

5. REFERENCES

[1] K. Nigam, and R. Ghani, "Analyzing the effectiveness and applicability of co-training," Proceedings of 9th International Conference on Information and Knowledge Management, pp. 86-93, 2000.

[2] V. Vapnik, Statistical Learning Theory, Springer, 1998.

[3] T. Joachims, "Transductive Inference for Text Classification using Support Vector Machines," Proceedings of the International Conference on Machine Learning, 1999.

[4] K. P. Bennett, A. Demiriz, "Semi-supervised support vector machines," Advances in Neural Information Processing Systems, 12, MIT Press, Cambridge, MA, pp 368-374, 1998.

[5] O. Chapelle, A. Zien, "Semi-supervised classification by low density separation," Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, pp. 57-64, Barbados, 2005.

[6] S. Park, B. Zhang, "Co-trained support vector machines for large scale unstructured document classification using unlabeled data and syntactic information," Information Processing and Management, Vol. 40(3), pp. 421-439, 2004.

[7] D. J. Newman, S. Hettich, C. L. Blake, C. J. Merz, "UCI Repository of machine learning databases," University of California, Irvine, CA, http://www.ics.uci.edu/ mlearn/MLRepository.html.

[8] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller and T. M. Vaughan, "Brain-computer interfaces for communication and control," Clinical Neurophysiology, 113, pp. 767-791, 2002.

[9] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. kubler, J. Perelmouter, E. Taub, H. Flor, "A spelling device for the paralysed," Nature, vol. 398, pp. 297-298, 1999.

[10] E. Donchin, K. M. Spencer, and R. Wijesinghe, "The mental prosthesis: Assessing the speed of a P300-based brain-computer interface", IEEE Transactions on Rehabilitation Engineering, Vol. 8(1), pp. 174-179, 2000.