Detection of Tornados Using an Incremental Revised Support Vector Machine with Filters

Hyung-Jin Son and Theodore B. Trafalis

School of Industrial Engineering, The University of Oklahoma
202 W. Boyd, CEC 124, Norman, OK 73019, U.S.A.
{son, ttrafalis}@ou.edu

Abstract. Recently Support Vector Machines (SVMs) have played a leading role in pattern classification. SVMs are quite effective to classify static data in numerous applications. However, the use of SVMs in dynamically data driven application systems (DDDAS) is somewhat limited. This motivates the development of incremental approaches to handle DDDAS. In an incremental learning approach, it is critical to keep a certain number of support vectors (SVs) without seriously sacrificing the generalization performance of SVMs. In this paper a novel incremental SVM method, called an incremental revised support vector machine with filters (IRSVMF) is proposed to resolve the above limitations. Computational experiments with tornado data show that this approach is quite effective to reduce the number of SVs and computing time and to increase the detection rate of tornados.

1 Introduction

Support Vector Machines (SVMs) have played a leading role in pattern classification. Applying SVMs into the real world has motivated the development of incremental approaches to deal with huge data that are continuously coming to a learning system. Numerous publications point out that the standard SVMs cannot properly handle large-scale data sets and that the incremental approach is a remedy to overcome limitations of the standard SVMs [1, 2, 3].

In applying the SVMs approach in an incremental framework for classification problems, we will face several limitations as follows:

First, support vectors (SVs) are accumulated as the incremental learning process is repeated. Therefore, it is important to control the number of SVs in SVMs. Second, SVMs waste most computing time for computation of kernel function values using less important data. If SVMs are used for training data from a particular classification problem such as an unbalanced classification problem in which there are many data in one class (less important) and few data in the other class (important), then use of computing time for kernel function evaluation among data points that are less important should be avoided to reduce the training time.

The tornado detection problem is an application to be considered in this study. It can be characterized as follows: First, it is a two-class (tornado and non-tornado) classification problem. Second, it is a problem with unbalanced data. The tornado class

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part III, LNCS 3993, pp. 506 – 513, 2006.

consists of few data in the entire tornadic and non-tornadic data set. Third, it is an asymmetric data importance problem. That is, the tornado class is relatively more important than the non-tornado class. Fourth, weather data related to tornado are periodically provided by weather radars. The standard SVM or other variants cannot properly train the weather data due to limited capacity of computation for training, and the size and periodic inflow of these data. Thus, the use of the standard SVM in dynamically data driven application systems (DDDAS) such as weather prediction is somewhat limited. This motivates the development of incremental approaches to handle DDDAS related to tornado detection.

Therefore the standard SVM should be revised to overcome these limitations. The objective of this study is (1) to develop a revised SVM to reduce the number of support vectors, (2) to construct an incremental learning procedure with the revised SVM, and (3) to make the incremental learning applicable to on-line settings by creating a filter to discard the most unimportant data.

This paper is organized as follows: Section 2 describes the standard SVM. In section 3, the incremental Revised SVM with filter (IRSVMF) is proposed. The tornado detection problem is described in section 4. Section 5 contains computational experiments and results. Conclusion follows in section 6.

2 Support Vector Machines

The basic idea of SVMs, introduced by Vapnik [4], is to construct a decision hyperplane to separate two-class samples maximizing the margin of separation. SVMs can be applied both in linearly separable and inseparable patterns cases. A brief mathematical explanation of SVMs is as follows:

Consider the training sample set $\{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the input pattern for the ith sample, and y_i is the corresponding output (± 1). Assume that the training samples and corresponding outputs are provided (i.e., supervised learning). The aim of SVM is to obtain the optimal weight vector w_o and bias b_o for the decision hyperplane, $w_o^T x + b_o = 0$, by solving the following optimization problem.

$$Max \ \frac{1}{2}\|w\|^2 \quad (1)$$

$$\text{s.t.} \ \ y_i[w^T x_i + b] \ \geq \ 1 \ , \ i = 1, 2, \cdots, N$$

The data points (x_i, y_i) along the hyperplanes with equality shown in (1) are called support vectors (i.e., critical data points that are located in the closest positions to the decision hyperplane).

In the linearly inseparable patterns case where there are some infeasibilities for the constraints of (1), the objective function is defined as $\Phi(w) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i$, where C is a user-defined parameter that controls the tradeoff between the number of inseparable data points and generalization of the SVM. Note that ξ_i, $i = 1, 2, \cdots, N$ are slack variables that measure the deviation of a data point from the separating hyperplane.

Using duality theory and kernel method [5], we have the following constrained optimization problem:

$$Max \quad F(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \tag{2}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$0 \le \alpha_i \le C \quad \text{for } i = 1, 2, \cdots, N$$

Note that K is a kernel function satisfying Mercer's condition. Possible kernel functions are polynomial or radial basis functions [5]. The resulting decision function in a binary classification problem takes the form $f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b)$, where α_i 's are the optimal solutions of (2), and b is computed using the Kuhn-Tucker conditions [4].

Bennett and Breadensteiner [6] developed the reduced convex hull concept giving a geometric explanation of the standard SVM. Crisp and Burges [7] also developed the same concept in a geometric interpretation of ν-SVM independently. The reduced SVM minimizes the distance of the reduced convex hulls of the positive and negative class respectively. This approach removes the effect of noisy data (e.g., outliers) and reduces the number of support vectors. In contrast, it increases the generalization error.

3 Incremental Revised Support Vector Machine with Filters

3.1 Revised Support Vector Machine

The standard SVM produces a huge amount of support vectors especially when the positive and negative training samples are highly overlapped with each other. It's obvious that the big size of support vectors requires more computing time and more storage space for training. Thus, it is natural to say that reducing the computational time and cost of the SVM is equivalent to decreasing the number of support vectors.

In addition, noisy data (e.g., outliers) might significantly affect the standard SVM producing an incorrect decision function. As a result, the incorrect decision function generates unexpected generalization errors and affects sequentially the next steps in the incremental learning process.

Therefore, the standard SVM must be modified to resolve the above problems. The geometric interpretation of the reduced SVM is quite useful and provides an alternative for modifying the standard SVM. However, it might be inefficient for particular problems such as unbalanced problems in which there are huge differences of data sizes in two classes. A similar situation happens in asymmetric importance problems where there are few data in one class with important meaning and a lot of unimportant data in the other class. For example, tornadic data are very few (as little as 2%) relatively to nontornadic data. If the reduced SVM is applied to this tornado detection problem, some valuable tornadic data will be lost. Thus, in order to properly solve this problem, the following geometric concept is proposed.

Consider a two-class classification problem. One class has few but important data. The other class has a lot of unimportant data. The data in the important class is preserved, and the data in the unimportant class will be reduced.

Geometrically, the revised SVM optimization problem in the linearly inseparable case takes the following form:

$$\begin{aligned} Min \quad & \frac{1}{2}\left\| \sum_{i=1}^{N_P} \alpha_i x_i - \sum_{j=1}^{N_N} \beta_j x_j \right\|^2 \\ \text{s.t.} \quad & \sum_{i=1}^{N_P} \alpha_i = 1 \ ; \ \sum_{j=1}^{N_N} \beta_j = 1 \\ & 0 \le \alpha_i \le 1 \qquad \text{for } i = 1, 2, \cdots, N_P \text{ (important class)} \\ & 0 \le \beta_j \le \mu \qquad \text{for } j = 1, 2, \cdots, N_N \text{ (unimportant class)} \\ & \text{where } 0 < \mu < 1 \end{aligned} \tag{3}$$

By imposing an upper bound on each multiplier, β_j, the convex hull that consists of data points in the unimportant class is shrunk. In contrast, the convex hull in the important class is preserved. The resulting revised SVM optimization problem for the linearly inseparable case is to maximize (4) subject to the constraints of (3).

$$Q(\alpha, \beta) = \sum_{i=1}^{N_P}\sum_{j=1}^{N_N} \alpha_i \beta_j K(x_i, x_j) - \frac{1}{2}\left[\sum_{i=1}^{N_P}\sum_{j=1}^{N_P} \alpha_i \alpha_j K(x_i, x_j) + \sum_{i=1}^{N_N}\sum_{j=1}^{N_N} \beta_i \beta_j K(x_i, x_j)\right] \tag{4}$$

3.2 Incremental Revised Support Vector Machine with Filters

In this study, we utilize the revised SVM instead of the standard SVM for training data in the incremental learning process. A filter is created to remove the most likely unimportant data prior to training.

If the whole data set is available, it is divided into several batches. In an on-line setting, only the first batch is created. If historic data are available, the first batch can be replaced by these data. In literature (e.g., [1]), the size of the batch is determined arbitrarily. The appropriate batch size can be obtained considering a trade-off between generalization error rate and computing time. For details, refer to [8]. Data in the first batch is trained by the revised SVM identifying the support vectors. These support vectors are included in the second batch. The iterative procedure is repeated until all batches are trained. After all data are trained, the final decision function is made for classification. The supporting hyperplane defined through the support vectors of the unimportant class plays a role as a filter. Because most of the unimportant data are located on the side of the supporting hyperplane referring to the unimportant class, this supporting hyperplane is a good yardstick for removing the possible unimportant data before training.

Data points passing this filter are put in the next batch until the size of the batch is filled up to the predetermined batch size. This filter is updated for every batch. Thus, this approach requires fewer batches than the traditional batch method in the incremental learning procedure. Hence, a new algorithm for incremental learning with revised SVM and a filter (IRSVMF) is proposed as follows:

Step 1. Determine the optimal batch size based on generalization error rate and computing time.
Step 2. Train the data in the first batch by the revised SVM.
Step 3. Add data points representing support vectors obtained in step 2, in the next batch.
Step 4. Inject a new point to the filter.
Step 5. If a new data point is located on the negative side of the supporting hyperplane of the reduced convex hull of the unimportant class, then discard the data point. Otherwise, add it in the next batch.
Step 6. If data size in a batch is equal to the predetermined batch size, go to the next step. Otherwise, go to step 4.
Step 7. Train the data in the batch and obtain support vectors, and corresponding decision function.
Step 8. Update the filter. Go to step 3.

Advantages of the incremental learning with the revised SVM and filter are as follows: First, it makes a selected support vector set to be as small as possible. Second, it can keep the memory and time complexity of the learning algorithm at a manageable level. Third, it can predict at a time when the whole data set is not yet available (on-line setting). When the data set is only periodically available (e.g., weather data or financial data), the traditional learning approach should wait until all data are available for training. In contrast, the incremental approach can be applied to train a small portion of the whole data set and has a capability to predict the class of an incoming data point using the constructed classifier before the next data point has arrived. Fourth, it can remove the serious effect of noisy data (e.g., outliers).

4 Tornado Detection

In real world, the SVM concept has been applied to many areas such as image classification, bioinformatics, text-categorization, data mining, and meteorology (e.g., [9]). It's hard to use the standard SVM due to many limitations (e.g., memory requirement, timely manner, availability of input data). Thus, the incremental learning approach developed in this study is quite suitable to the following situations: First, when all information data in a system cannot be obtained at once where periodically information data are injected to a system. Second, when one class has relatively more important points than the other one in a two-class problem.

One interesting application of the above incremental learning approach with good performance is the tornado detection problem. Tornado is a rare but significantly critical event in the real world as well as in the meteorology community. Based on the weather data produced from the Weather Surveillance Radar 1988 Doppler (WSR-88D), the Mesocyclone Detection Algorithm (MDA) is currently used to detect tornados [10]. Typically, a tornado confusion matrix is utilized in order to measure the performance for tornado detection as shown in Table 1.

Table 1. Tornado Confusion Matrix

		Tornado Observed		
Forecast Tornado		Yes	No	
	Yes	Hit (a)	False alarm (b)	"Yes" Forecasts
	No	Miss (c)	Correct (d)	"No" Forecasts
		"Yes" Observation	"No" Observation	Total number of observations

When the revised SVM is used, the "false alarm" rate might be increased because the size of the convex hull containing non-tornado data is reduced. However, the number of "miss" events will be substantially reduced because the convex hull containing tornado data is not shrunk. It is critical to decrease the "miss" rate in the tornado detection problem because a high "miss" rate brings unexpected and serious disasters frequently. From the confusion matrix, the probability of detection (POD) is computed in (5).

$$\text{Probability of detection (POD)} = \frac{a}{a+c}. \tag{5}$$

5 Experiments and Computational Results

5.1 Experiments

In this study, we use the MDA data provided by the National Severe Storms Laboratory (NSSL). These tornadic and nontornadic data generated from 1994 to 1999 are randomly selected to produce ten training sets and ten testing ones where each set has 1500 data. Each datum has 23 attributes that are related to information such as velocity and shear. These attributes have been successfully used for tornado detection in the literature [10]. To reflect the real situation, tornadic data form 10% in a training set as well as in a testing one.

Three approaches are performed and compared: the incremental approach with standard SVM (ISSVM), the incremental approach with revised SVM (IRSVM), and the incremental approach with revised SVM and filter (IRSVMF). These approaches are compared in terms of POD, total CPU time, number of batches, and "miss" rate. For each approach, training and testing are performed ten times respectively, and their average values are computed in terms of the above criteria.

In the incremental approaches with standard SVM and with revised SVM, each batch size is set to 300 (for details, refer to [8]). The incremental approach with filters uses a batch with size of 300 data, which consists of data passing the filter. Note that support vectors obtained from the previous batch are added to the training batch. MATLAB codes provided by [11] are entirely revised to run the incremental step and used in a Pentium IV 2.8GHz with 1 GB RAM to perform all experiments.

5.2 Computational Results

After performing each approach with ten training and testing data sets, the averaged results are shown in Table 2.

Table 2. Comparisons of methods

Methods	POD	Total CPU time (Sec)	Number of SVs	"Miss" rate (%)
ISSVM	0.62	754.62	57	3.83
IRSVM	0.69	406.26	11	3.14
IRSVMF	0.60	314.46	11	3.97

Even if the total CPU time and the number of support vectors are reduced, IRSVM outperforms ISSVM in terms of POD and "miss" rate. The CPU time and the number of support vectors for each batch are shown in Figure 1.

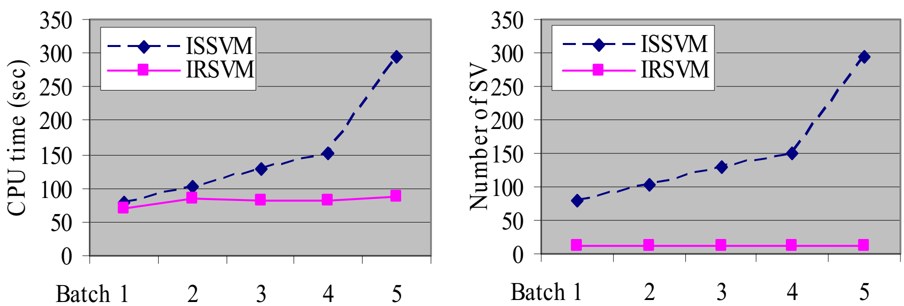

Fig. 1. Comparison of ISSVM and IRSVM in terms of computing time and number of SVs

Since the number of support vectors is increased as each batch is sequentially trained, the computing time is also dramatically increased, whereas IRSVM keeps the same number of support vectors and requires smaller computing time. When IRSVMF is applied, computing time is also significantly reduced although POD is slightly dropped.

6 Conclusions

Tornados are rare but very critical events in real world as well as in the meteorological community. It is quite important that large-scale data such as weather radar data should be properly handled such that the accuracy of tornado prediction should be improved. To accomplish those goals, the incremental revised SVM with a filter concept is presented. Our results show that the revised SVM outperforms the standard SVM in terms of computing time and number of batches and can be used in other DDDAS. The revised SVM concept improves the accuracy of tornado prediction, and the use of filter in the revised SVM also reduces the computing time. In the future, those algorithms will be tested in a real operational setting.

Acknowledgement

This material is based on research funded by National Science Foundation Grant EIA-0205628.

References

1. Demeniconi, C. and D. Gunopulos, 2001. "Incremental support vector machine construction," *Proceedings of the IEEE International Conference on Data Mining,* San Jose, CA, pp. 589-592.
2. Syed, N. A., H. Liu, and K. K. Sung, 1999. "Incremental Learning with Support Vector Machines," *Workshop on Support Vector Machines, International Joint Conference on Artificial Intelligence,* Stockholm, Sweden.
3. Shilton, A., M. Palaniswami, D. Ralph, and A. C. Tsoi, 2005. "Incremental training of support vector machines," IEEE Transactions on neural networks, vol. 16, no. 1, pp 114-131.
4. Vapnik, V. N., 1995. *The Nature of Statistical Learning Theory*. New York, NY: Springer Verlag.
5. Haykin, S., 1998. *Neural Networks: A Comprehensive Foundation,* 2nd edit., Upper Saddle River, NJ: Prentice-Hall.
6. Bennet, K. P, and E. J. Bredensteiner, 2000. "Duality and Geometry in SVM Classifiers," *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 57-64.
7. Crisp, David J. and C. J. C. Burges, 1999. *A Geometric Interpretation of ν-SVM Classifiers.* In Advances in Neural Information Processing Systems (NIPS) vol. 12. Cambridge, MA: MIT Press.
8. Son, H, T. B. Trafalis, and M. Richman, 2005. "Determination of the Optimal Batch Size in Incremental Approaches: An Application to Tornado Detection," *Proceedings of the International Joint Conference of Neural Network*, Montreal, Canada, pp. 2706-2710.
9. Cristianini, N. and J. Shawe-Taylor, 2000. *An Introduction to support vector machines*. Cambridge University Press.
10. Marzban, C. and G. J. Stumpf, "A Neural Network for Tornado Prediction Based on Doppler Radar Derived Attributes," *Journal of Applied Meteorology*, 1996, vol. 35, pp. 617-626.
11. Gunn, S. R., 1997. *Support Vector Machines for Classification and Regression*. Technical Report, Image Speech and Intelligent Systems Research Group, University of Southhampton.