Active Learning with Support Vector Machines for Tornado Prediction

Theodore B. Trafalis[1], Indra Adrianto[1], and Michael B. Richman[2]

[1] School of Industrial Engineering, University of Oklahoma, 202 West Boyd St, Room 124, Norman, OK 73019, USA
ttrafalis@ou.edu, adrianto@ou.edu
[2] School of Meteorology, University of Oklahoma, 120 David L. Boren Blvd, Suite 5900, Norman, OK 73072, USA
mrichman@ou.edu

Abstract. In this paper, active learning with support vector machines (SVMs) is applied to the problem of tornado prediction. This method is used to predict which storm-scale circulations yield tornadoes based on the radar derived Mesocyclone Detection Algorithm (MDA) and near-storm environment (NSE) attributes. The main goal of active learning is to choose the instances or data points that are important or have influence to our model to be labeled and included in the training set. We compare this method to passive learning with SVMs where the next instances to be included to the training set are randomly selected. The preliminary results show that active learning can achieve high performance and significantly reduce the size of training set.

Keywords: Active learning, support vector machines, tornado prediction, machine learning, weather forecasting.

1 Introduction

Most conventional learning methods use static data in the training set to construct a model or classifier. The ability of learning methods to update the model dynamically, using new incoming data, is important. One method that has this ability is active learning. The objective of active learning for classification is to choose the instances or data points to be labeled and included in the training set. In many machine learning tasks, collecting data and/or labeling data to create a training set is costly and time-consuming. Rather than selecting and labeling data randomly, it is better if we can label the data that are important or have influence to our model or classifier.

In tornado prediction, labeling data is considered costly and time consuming since we need to verify which storm-scale circulations produce tornadoes in the ground. The tornado events can be verified from facts in the ground including photographs, videos, damage surveys, and eyewitness reports. Based on tornado verification, we then determine and label which circulations produce tornadoes or not. Therefore, applying active learning for tornado prediction to minimize the need for the instances and use the most informative instances in the training set in order to update the classifier would be beneficial.

Y. Shi et al. (Eds.): ICCS 2007, Part I, LNCS 4487, pp. 1130–1137, 2007.

In the literature, the Mesocyclone Detection Algorithm (MDA) attributes [1] derived from Doppler radar velocity data have been used to detect tornado circulations. Marzban and Stumpf [1] applied artificial neural networks (ANNs) to classify MDA detections as tornadic or non-tornadic circulations. Additionally, Lakshmanan et al. [2] used ANNs and added the near-storm environment (NSE) data into the original MDA data set and determined that the skill improved marginally. Application of support vector machines (SVMs) using the same data set used by Marzban and Stumpf [1] has been investigated by Trafalis et al. [3]. Trafalis et al. [3] compared SVMs with other classification methods, such as ANNs and radial basis function networks, concluding that SVMs provided better performance in tornado detection. Moreover, a study by Adrianto et al. [4] revealed that the addition of NSE data into the MDA data can improve performance of the classifiers significantly. However, those experiments in the literature were conducted using static data.

In this paper, we investigated the application of active learning with SVMs for tornado prediction using the MDA and NSE data. We also compared this method to passive learning with SVMs using these data where the next instances to be added to the training set are randomly selected.

2 Data and Analysis

The original data set was comprised of 23 attributes taken from the MDA algorithm [1]. These attributes measure radar-derived velocity parameters that describe various aspects of the mesocyclone. Subsequently, 59 attributes from the NSE data [2] were incorporated to this data set. The NSE data described the pre-storm environment of the atmosphere on a broader scale than the MDA data, as the MDA attributes are radar-based. Information on wind speed, direction, wind shear, humidity lapse rate and the predisposition of the atmosphere to accelerate air rapidly upward over specific heights were measured by the NSE data. Therefore, the MDA+NSE data consist of 82 attributes.

3 Methodology

3.1 Support Vector Machines

The SVM algorithm was developed by Vapnik and has proliferated into a powerful method in machine learning [5-7]. This algorithm has been used in real-world applications and is well known for its superior practical results. In binary classification problems, the SVM algorithm constructs a hyperplane that separates a set of training vectors into two classes (Fig. 1). The objective of SVMs (the primal problem) is to maximize the margin of separation and to minimize the misclassification error. The SVM formulation can be written as follows [8]:

$$\begin{aligned} &\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i \\ &\text{subject to } y_i\left(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b\right) \geq 1-\xi_i,\ \xi_i \geq 0,\ \ i=1,...,l \end{aligned} \tag{1}$$

where w is the weight vector perpendicular to the separating hyperplane, b is the bias of the separating hyperplane, ξ_i is a slack variable, and C is a user-specified parameter which represents a trade off between generalization and misclassification. Using Lagrange multipliers α, the SVM dual formulation becomes [8]:

$$\max Q(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=i}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j$$
$$\text{subject to} \sum_{i=1}^{l} \alpha_i y_i = 0, \quad 0 \le \alpha_i \le C, \quad i = 1,...,l \tag{2}$$

The optimal solution of Eq. (1) is given by $\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i$ where $\alpha = (\alpha_1,...,\alpha_l)$ is the optimal solution of the optimization problem in Eq. (2). The decision function is defined as:

$$g(\mathbf{x}) = \text{sign}\,(f(\mathbf{x})) \text{ , where } f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \tag{3}$$

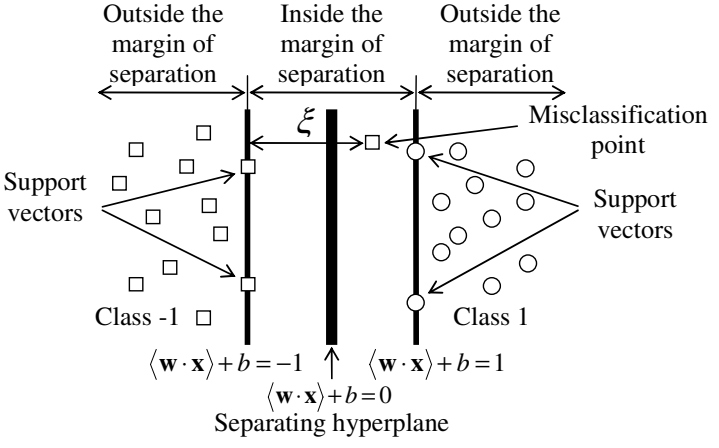

Fig. 1. Illustration of support vector machines

For solving nonlinear problems, the SVM algorithm maps the input vector **x** into a higher-dimensional feature space through some nonlinear mapping Φ and constructs an optimal separating hyperplane [7]. Suppose we map the vector **x** into a vector in the feature space $(\Phi_1(\mathbf{x}),\ldots,\Phi_n(\mathbf{x}),\ldots)$, then an inner product in feature space has an equivalent representation defined through a kernel function K as $K(\mathbf{x}_1,\mathbf{x}_2) = \langle \Phi(\mathbf{x}_1).\Phi(\mathbf{x}_2) \rangle$ [8]. Therefore, we can introduce the inner-product kernel as $K(\mathbf{x}_i,\mathbf{x}_j) = \langle \Phi(\mathbf{x}_i).\Phi(\mathbf{x}_j) \rangle$ and substitute the dot-product $\langle \mathbf{x}_i . \mathbf{x}_j \rangle$ in the dual problem in Eq. (2) with this kernel function. The kernel function used in this study is the radial basis function (RBF) with $K(\mathbf{x}_i,\mathbf{x}_j) = \exp\left(-\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2\right)$ where γ is the parameter that controls the width of RBF.

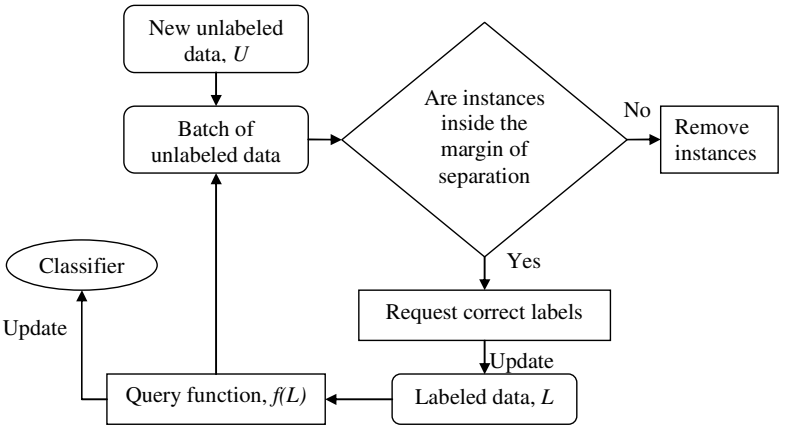

Fig. 2. Active learning with SVMs scheme

3.2 Active Learning with SVMs

Several active learning algorithms with SVMs have been proposed by Campbell et al. [9], Schohn and Cohn [10], and Tong and Koller [11]. Campbell et al. [9] suggested that the generalization performance of a learning machine can be improved significantly with active learning. Using SVMs, the basic idea of the active learning algorithms is to choose the unlabeled instance for the next query closest to the separating hyperplane in the feature space which is the instance with the smallest margin [9-11]. In this paper, we choose the instances that are inside the margin of separation to be labeled and included in the training set. Since the separating hyperplane lies in the middle of the margin of separation, these instances will have an effect on the solution. Thus, the instances outside the margin of separation will be removed.

Suppose we are given an unlabeled pool U and a set of labeled data L. The first step is to find a query function $f(L)$ where, given a set of labeled data L, determine which instances in U to query next. This idea is called the pool-based active learning. Scheme of active learning can be found in Fig. 2.

3.3 Measuring the Quality of the Forecasts for Tornado Prediction

In order to measure the performance of a tornado prediction classifier, it is important to compute scalar forecast evaluation scores such as the Critical Success Index (CSI), Probability of Detection (POD), False Alarm Ratio (FAR), Bias, and Heidke Skill Score (HSS), based on a "confusion" matrix or contingency table (Table I). Those skill scores are defined as: CSI = a/(a+b+c), POD = a/(a+c), FAR = b/(a+b), Bias = (a+b)/(a+c), and HSS = 2(ad-bc)/[(a+c)(c+d)+(a+b)(b+d)].

It is important not to rely solely on a forecast evaluation statistic incorporating cell d from the confusion matrix, as tornadoes are rare events with many correct nulls. This is important as there is little usefulness in forecasting "no" tornadoes every day. Indeed, the claim of skill associated with such forecasts including correct nulls for rare events has a notorious history in meteorology [13].The CSI measures the

accuracy of a solution equal to the total number of correct event forecasts (hits) divided by the total number of tornado forecasts plus the number of misses (hits + false alarms + misses) [12]. It has a range of 0 to 1, where 1 is a perfect value. The POD calculates the fraction of observed events that are correctly forecast. It has a perfect score of 1 and a range is 0 to 1 [14]. The FAR measures the ratio of false alarms to the number of "yes" forecasts. It has a perfect score of 0 with its range of 0 to 1 [14]. The Bias computes the total number of event forecasts (hits + false alarms) divided by the total number of observed events. It shows whether the forecast system is underforecast (Bias < 1) or overforecast (Bias > 1) events with a range of 0 to $+\infty$ and perfect score of 1 [14]. The HSS [15] is commonly used in forecasting since it considers all elements in the confusion matrix. It measures the relative increase in forecast accuracy over some reference forecast. In the present formulation, the reference forecast is a random guess. A skill value > 0 is more accurate than the reference. It has a perfect score of 1 and a range of -1 to 1.

Table 1. Confusion matrix

		Observation	
		Yes	**No**
Forecast	**Yes**	hit a	false alarm b
	No	miss c	correct null d

4 Experiments

The data were divided into two sets: training and testing. In the training set, we had 382 tornadic instances and 1128 non-tornadic instances. In order to perform online setting experiments, the training instances were arranged in time order. The testing set consisted of 387 tornadic instances and 11872 non-tornadic instances. For both active and passive learning experiments, the initial training set was the first 10 instances consisted of 5 tornadic instances and 5 non-tornadic instances. At each iteration, new data were injected in a batch of several instances. Two different batch sizes, 75 and 150 instances, were used for comparison. In passive learning with SVMs, all incoming data were labeled and included in the training set. Conversely, active learning with SVMs only chooses the instances from each batch which are most informative for the classifier. Therefore, the classifier was updated dynamically at each iteration. The performance of the classifier can be measured by computing the scalar skill scores (Section 3.3) on the testing set. The radial basis function kernel with $\gamma = 0.01$ and $C = 10$ was used in these experiments. The experiments were performed in the Matlab environment using LIBSVM toolbox [16].

Before training a classifier, the data set needs to be normalized. We normalized the training set so that each attribute has the mean of 0 and the standard deviation of 1. Then, we used the mean and standard deviation from each attribute in the training set to normalize each attribute in the testing set.

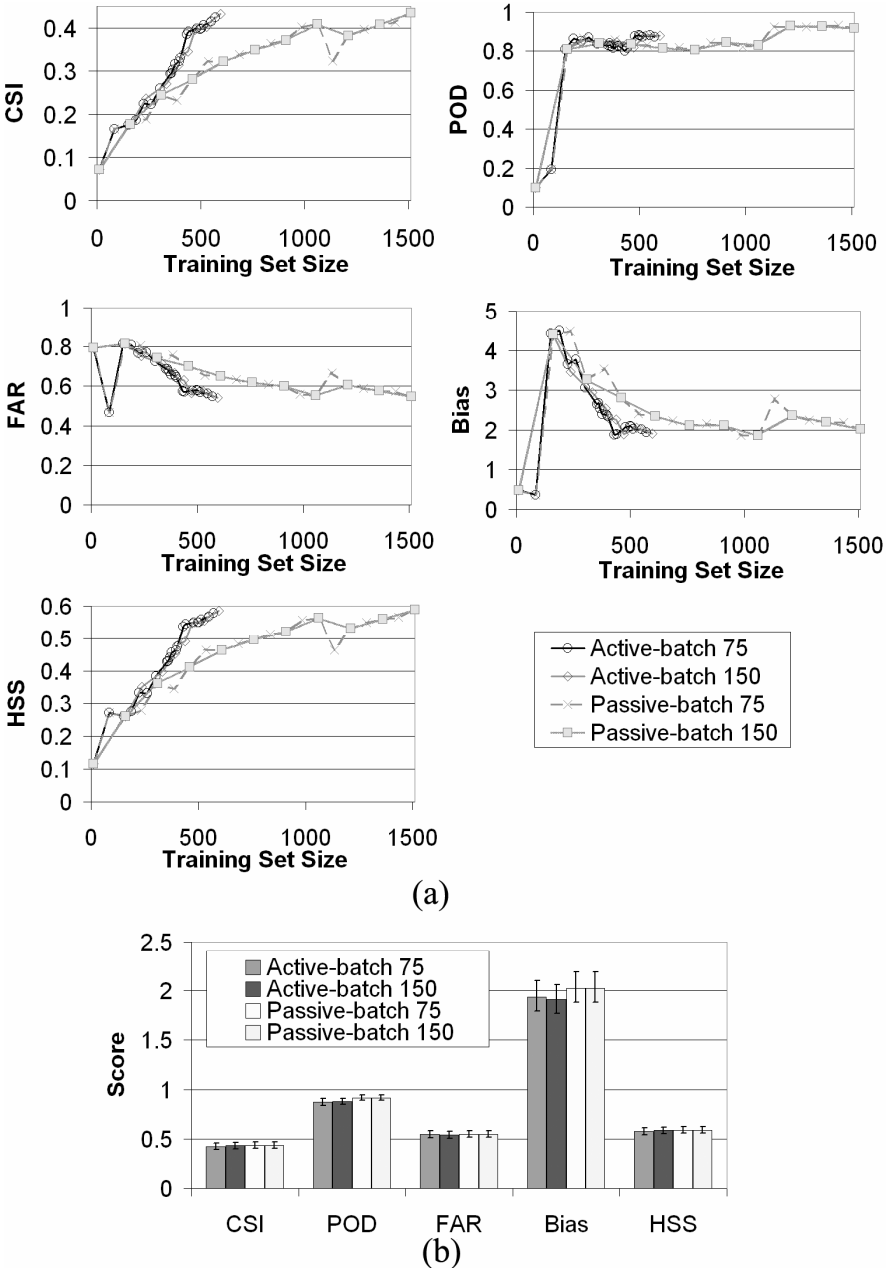

Fig. 3. (a) The results of CSI, POD, FAR, Bias, and HSS on the testing set using active and passive learning at all iterations. (b) The last iteration results with 95% confidence intervals on the testing set.

5 Results

It can bee seen from Fig. 3a for all skill scores, CSI, POD, FAR, Bias, and HSS, active learning achieved relatively the same scores as passive learning using less training instances. From the FAR diagram (Fig. 3a), we noticed that at early iteration the active and passive learning FAR with the batch size of 75 dropped suddenly. It happened because the forecast system was underforecast (Bias < 1) at that stage. Ultimately, every method produced overforecasting. Furthermore, Fig. 3b showed the last iteration results with 95% confidence intervals after conducting bootstrap resampling with 1000 replications [17]. The 95% confidence intervals between active and passive learning results with the batch sizes of 75 and 150 overlapped each other for each skill score, so the differences were not statistically significant. These results indicated that active learning possessed similar performance compared to passive learning using the MDA and NSE data set.

The results in Fig. 4 showed that active learning significantly reduced the training set size to attain relatively the same skill scores as passive learning. Using the batch size of 75 instances, only 571 labeled instances were required in active learning whereas in passive learning 1510 labeled instances were needed (Fig. 4a). This experiment reveals that about 62.6% reduction was realized by active learning. Using the batch size of 150 instances, active learning can reduce the training set size by 60.5% since it only needed 596 labeled instances whereas passive learning required 1510 labeled instances (Fig. 4b).

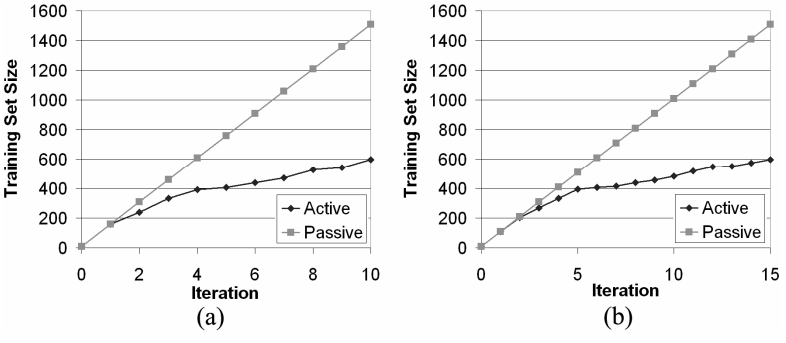

Fig. 4. Diagrams of training set size vs. iteration for the batch sizes of (a) 75 and (b) 150 instances

6 Conclusions

In this paper, active learning with SVMs was used to discriminate between mesocyclones that do not become tornadic from those that do form tornadoes. The preliminary results showed that active learning can significantly reduce the size of training set and achieve relatively similar skill scores compared to passive learning. Since labeling new data is considered costly and time consuming in tornado prediction, active learning would be beneficial in order to update the classifier dynamically.

Acknowledgments. Funding for this research was provided under the National Science Foundation Grant EIA-0205628 and NOAA Grant NA17RJ1227.

References

1. Marzban, C., Stumpf, G.: A neural network for tornado prediction based on Doppler radar-derived attributes. J. Appl. Meteorol. 35 (1996) 617-626
2. Lakshmanan, V., Stumpf, G., Witt, A.: A neural network for detecting and diagnosing tornadic circulations using the mesocyclone detection and near storm environment algorithms. In: 21st International Conference on Information Processing Systems, San Diego, CA, Amer. Meteor. Soc. (2005) CD–ROM J5.2
3. Trafalis, T.B., Ince, H., Richman M.B. Tornado detection with support vector machines. In: Sloot PM et al. (eds). Computational Science-ICCS (2003) 202-211
4. Adrianto, I., Trafalis, T.B., Richman, M.B., Lakshmivarahan, S., Park, J.: Machine learning classifiers for tornado detection: sensitivity analysis on tornado data sets. In: Dagli C. Buczak, A., Enke, D., Embrechts, M., Ersoy, O. (eds.): Intelligent Engineering Systems Through Artificial Neural Networks, Vol. 16. ASME Press (2006) 679-684
5. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Haussler D (ed): 5th Annual ACM Workshop on COLT. ACM Press, Pittsburgh, PA (1992) 144-152
6. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer Verlag, New York (1995)
7. Vapnik, V.N.: Statistical Learning Theory. Springer Verlag, New York (1998)
8. Haykin S.: Neural Networks: A Comprehensive Foundation. 2nd edn. Prentice Hall, New Jersey (1999)
9. Campbell, C., Cristianini, N., Smola, A.: Query learning with large margin classifiers. In: Proceedings of ICML-2000, 17th International Conference on Machine Learning. (2000)111-118
10. Schohn, G., Cohn, D.: Less is more: Active learning with support vector machines. In: ICML Proceedings of ICML-2000, 17th International Conference on Machine Learning, (2000) 839-846
11. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. J. Mach. Learn. Res. 2 (2001) 45-66
12. Donaldson, R., Dyer, R., Krauss, M.: An objective evaluator of techniques for predicting severe weather events. In: 9th Conference on Severe Local Storms, Norman, OK, Amer. Meteor. Soc. (1975) 321-326
13. Murphy, A.H.: The Finley affair: a signal event in the history of forecast verifications. Weather Forecast. 11 (1996) 3-20
14. Wilks, D.: Statistical Methods in Atmospheric Sciences. Academic Press, San Diego, CA (1995)
15. Heidke P.: Berechnung des erfolges und der gute der windstarkvorhersagen im sturmwarnungsdienst, Geogr. Ann. 8 (1926) 301-349
16. Chang, C., Lin, C.: LIBSVM: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm> (2001)
17. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall, New York (1993)