REVIEW ON SUPPORT VECTOR MACHINES AND ITS APPLICATION TO PATTERN RECOGNITION

Naveen Parihar

Mississippi State University
Mississippi State, MS 39762 USA
email: parihar@isip.mstate.edu

ABSTRACT

Support Vector Machines (SVMs) is a relatively novel pattern recognition approach that has attracted a great deal of attention from the machine learning and automatic speech recognition community. SVMs are based on the theory of underlying statistical learning, specially the Structural Risk Minimization(SRM). SVMs have proven to be successful in many pattern classification problems such as image identification and face recognition. In many of these problems, SVMs have outperformed other non-linear classifiers such as artificial neural networks and k-nearest neighbors since these suffer from many deficiencies such as tendency to overfit and slow convergence.

In this paper, we use SVMs for the two multi-class pattern classification problems. The first problem is purely static in nature while the second problem has a temporal dimension associated with it. The excellent recognition rates achieved in both the experiments indicate that SVMs are well-suited for the pattern recognition problems. SVMs performed worst than Hidden markov models (HMMs) on the second data set since the HMMs has an advantage of modelling underlying markov process.

1. INTRODUCTION

The theory of SVMs was first introduced by Vapnik, based on the principle of Structural Risk Minimization [1]. Intuitively, given the set of samples belonging to two classes, SVMs learn the boundary between these two classes by mapping the input samples to a high dimensional space and then finding a hyperplane in this high dimensional space that separates the samples of the two classes. This hyperplane is choosen such that it leaves the largest fraction of samples of the same class on the same side while maximizing the distance from the closest training samples from each class. This distance is referred to as margin in SVMs literature.

The use of the connectionist systems such as neural networks has been limited because of various limitations such as lack of generalization[2], optimization process [3] and slow convergence [2,4]. The SVMs have clearly two distinct advantages. Firstly they have an excellent ability to generalize and secondly they do not need any fine tuning of parameters. SVMs also demonstrate good convergence property.

The aim of this paper is to demonstrate the potential of SVMs on pattern classification problems and familiarization with SVMs theory. The experiments were conducted on two sets of data. First data set without the temporal characteristic and the second data set with temporal characteristic. Good recognition rates were achieved on both the data sets.

This paper is organized as follows. In section 2, we give a brief introduction to the theory of SVMs. In section 3, we describe the experiments done to classify the two data sets using SVMs. Finally, section 4 summarizes the conclusions that can be drawn from the experiments presented.

2. THEORITICAL OVERVIEW

In this section, we recall the basic notions of the theory of SVMs. Lack of space prohibits a detailed discussions on SVMs, a good tutorial [5,6] is recommended for detailed information on the subject. We have not included the theory behind Structural Risk Minimization. Again, this can be referred in [1] for details. We start the introduction to the theory of

SVMs with the simple case of linearly separable data points.

2.1. The Linearly Separable Case

Suppose, we have a set of linearly separable training samples $x_1, x_2, \ldots, x_m$ where $x_i \in R^n$. Each sample has a corresponding label $y_1, y_2, \ldots, y_m$ where $y_i \in \{-1, 1\}$. The label indicates one of the two classes that each corresponding sample belongs to. Now, a plane known as hyperplane in the SVMs literature is given by $(\dot{w} \cdot x) + b$ separates the data if and only if

$$(\dot{w} \cdot x_i) + b > 0 \qquad if \qquad y_i = 1 \tag{1}$$

$$(\dot{w} \cdot x_i) + b < 0 \qquad if \qquad y_i = -1 \tag{2}$$

Here, scaling w and b gives

$$(\dot{w} \cdot x_i) + b \geq 1 \qquad if \qquad y_i = 1 \tag{3}$$

$$(\dot{w} \cdot x_i) + b \leq -1 \qquad if \qquad y_i = -1 \tag{4}$$

We can combine the above two equations to get an equivalent form

$$y_i((\dot{w} \cdot x_i) + b) \geq 1 \qquad \forall i \tag{5}$$

To find the optimal hyperplane that separates the two classes, we need to find the plane that maximizes the distance between the itself and the closest sample. This maximized distance of the closest sample is given by

$$d(w, b) = \underset{\{x_i | y_i = 1\}}{min} \frac{w \cdot x_i + b}{|w|} - \underset{\{x_i | y_i = 1\}}{max} \frac{w \cdot x_i + b}{|w|} \tag{6}$$

From equation (3) and (4), we see that the appropriate minimum and maximum values are given by ± 1. So, now we need to maximize the following equation

$$d(w, b) = \frac{1}{|w|} - \frac{-1}{|w|} = \frac{2}{|w|} \tag{7}$$

This equation gives the distance between the two closest samples on either side of the hyperplane. So, now the problem is to maximize equation (7) or equivalently minimizing $|w|^2 / 2$ subject to the constraints as given by equation (5). This constrained minimization problem can made into an unconstrained optimization problem for equality constraints by forming the Lagrangian, and solving for the dual problem. This dual is given by

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i, j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \tag{8}$$

The problem now is to minimize equation (8) subjected to

$$\alpha_i \geq 0 \tag{9}$$

$$\sum_i \alpha_i y_i = 0 \tag{10}$$

Here α_i are the Lagrange multipliers; one for each training sample. The training samples for which the Lagrange multiplier is non-zero are called as S*upport Vectors*, and in this case the equality in equation (5) hold. All the samples with Lagrange multipliers equal to zero can be removed from the training set without affecting the position of the final hyperplane. Another training data set with the same non-zero Lagrange multiple would produce exactly the same hyperplane.

This is a very well known quadratic programming problem that can be solved using software packages available. Such solvers employ non-trivial algorithms like chunking [6] when we have large training data sets.

2.2. The Linearly Non-separable Case

The optimization problem described in section 2.1 will have no solution if the data is not separable. In such a case, we modify the constraints given by equations (3) and (4) to be loose by adding the slack variables ξ_i. However, a penalty is incurred for every misclassification:

$$(\dot{w} \cdot x_i) + b \geq 1 - \xi_i \qquad if \qquad y_i = 1 \tag{11}$$

$$(\dot{w} \cdot x_i) + b \leq \xi_i - 1 \qquad if \qquad y_i = -1 \tag{12}$$

$$\xi_i \geq 0 \qquad \forall i \tag{13}$$

If x_i is to be misclassified, we must have $\xi_i > 1$, and hence we have the upper bound on the number of errors that is $\sum_i \xi_i$. So, we add the penalty for misclassifying training samples by replacing the function with

$\frac{|w|^2}{2} + C\left(\sum_i \xi_i\right)$, where C is a parameter that allows us to specify how strictly we want the classifier to fit the training data. Higher the value of C, harder the system will try to minimize the training errors. Thus, increasing the training time.

As we formed the Lagrangrain in the previous section, we form the Lagrangrain here also. The dual is now given by

$$\sum_i \alpha_i - \frac{1}{2}\sum_{i,\,j} \alpha_i \alpha_j y_i y_j x_i \bullet x_j \qquad (14)$$

This dual is minimized subjected to

$$0 \geq \alpha_i \geq C \qquad (15)$$

$$\sum_i \alpha_i y_i = 0 \qquad (16)$$

Now from the above equations α_i can be calculated from which the position of the hyperplane can found.

2.3. The Non-linear Case

The classification framework mentioned above is limited to the linear separating hyperplanes. SVMs solve this problem by mapping the sample points into a higher dimensional space where the sample points are separable using a non-linear mapping. This mapping is chosen before training according to the type of data. A mapping is done by the map $\Phi;R^n \rightarrow H$, where H is a space with dimensions higher than n. We now seek a hyperplane in this higher dimensional space. This is equivalent to finding a non-linear separating surface in R^n.

We see that the data appears only as dot products in training equations (8), (9) and (10). Hence, in high dimensional space we have the dot products of the form $\Phi(x_i) \cdot \Phi(x_j)$. This may be very difficult or computationally very expensive to find, especially if the dimension of H are very large. A very neat way to overcome this problem is to use *kernel function* given by $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$. Now, we can use this in place of dot product in the optimization equations and we would never need to know explicitly what Φ is.

Some of the kernel functions that are normally used are the polynomial kernel $K(x_i, x_j) = (x \cdot y + 1)^p$ and the Gaussian radial basis function (RBF) given by

$$K(x_i, x_j) = e^{|x-y|/(2\sigma\ )}$$

3. EXPERIMENTS

The SVMs were applied to the two given pattern classification problems [7] since SVMs have proven to be very effective on various pattern classification problems. The main aim of the two set of experiments was to demonstrate that SVMs provide a good discrimination. All the experiments described in this paper were conducted using the publicly available SVM toolkit, SVMTorch II [8]. All the experiments were conducted by using N one vs. all classifiers and then combining these N classifiers using “multiclass” mode of the software.

Data set 1 consisted of static classification problem[7] with 10 dimension of vectors, 11 classes, 83 eval set vectors, 379 development set vectors and 528 training set vectors. Table 1 shows the performance by varying the parameter C = 1, 50, 100, 150, 1000 on a linear kernel. C controls the trade-off between empirical fit to the data and the capacity of the learning machine. The best performance was obtained for C=100. The results of two-class SVM classifiers with various kernels are shown in Table 2. The best result with a classification accuracy of 92.63% was obtained with RBF kernel when using 11 one vs. all classifiers. This result was obtained by averaging the classification rates of all the 11 two-class classifiers.The SVMTorch employs the maximum score theory to classify in multiclass mode. The classification error obtained in this mode was **46.7%**. Extracting the probabilities from SVM outputs as post classification processing is a possible way to increase the accuracy.

Data set 2 consisted of temporal modeling problem involving 39 dimension of vectors, 5 classes, 225 eval set vectors (sets of 5 vectors for each class), 350 development set vectors (sets of 5 vectors for each class) and 925 training set vectors (sets of 5 vectors

Data Set	Kernel	C	Accuracy
one	RBF	1	91.20%
one	RBF	10	92.20%
one	RBF	50	92.37%
one	RBF	100	92.63%
one	RBF	150	92.61%
one	RBF	1000	92.13%

Table 1: Performance of two-class SVM as a function of C.

Data Set	Kernel	Accuracy
one	Polynomial (II degree)	91.44%
one	Polynomial (III degree)	90.41%
one	Polynomial (IV degree)	81.07%
one	RBF	92.63%
two	Polynomial (II degree)	84.57%
two	Polynomial (III degree)	84.23%
two	RBF	83.60%

Table 2: Classification Results in two-class mode.

for each class). As shown in Table 2, the best classification accuracy of 83.60% with C=100 was obtained with RBF Kernel in two-class mode. In multi-class mode the performance degraded to an classification error of **27.43%**. Comparing this result to previous 1999 course HMMs results we see that SVMs accuracy is less by 9.63%. This is the drawback of SVMs since they are incapable of successfully modelling the time-varying dynamics of the data such as speech.

4. SUMMARY

The experiments conducted demonstrate that the SVMs can be successfully applied to pattern classification problems though they have drawback of inability to model temporal characteristics. We also observed that the multiclass classification requires a postprocessing. Since SVMs need only the support vectors for classification and not the complete training set, they are very efficient classifiers.

5. FUTURE WORK

Due to the time limitation, code that extracts the probabilities from two-class SVM outputs as post classification processing was not implemented. Implementing this code looks a promising idea that would lower the error classification rate.

REFERENCES

[1] V. Vapnik, *The Nature of Statistical Learning Theory,* Springer-Verlag, New York, 1995.

[2] S. Lawrence, C.L. Giles, and A.C. Tsoi, “Lessons in neural-network training: Overfitting may be harder than expected,” *Proceedings of the 14th National Conference on Artificial Intelligence,* AAAI Press, pp. 540-545, 1997.

[3] F. Rosenblatt, “The Perceptron: A Perceiving and Recognizing Automaton”, *Cornell Aeronautical Laboratory Report*, 85-460-1, 1957.

[4] S. Lawrence, C. L. Giles, and A. C. Tsoi, “What size neural network gives optimal generalization,” *Technical Report UMIACSTR-96-22*, Institute for Advanced Computer Studies, University of Maryland, USA, April 1996.

[5] Cristopher Burges, “A tutorial on Support Vector Machines for Pattern Recognition,” *Data Mining and Knowledge Discovery*, 2(2), 1998.

[6] A. Ganapathiraju, “Support Vector Machines for Speech Recognition,” Ph.D. Dissertation, Department of Electrical and Computer Engineering, Mississippi State University, June 2000.

[7] “Common Evaluation, Pattern Recognition ECE 8990,” htt*p://www.isip.msstate.edu/publications/courses/ece_8990_pr/exams/2001/*, Mississippi State University, Mississippi State, Mississippi, USA, 2001.

[8] Ronan Collobert and Samy Bengio, “SVMTorch: Support Vector Machines for Large-Scale Regression Problems,” *Journal of Machine Learning Research*, vol 1, pages 143-160, 2001.