The Application of Support Vector Machine for Speech Classification

O. Gauci, C.J. Debono, E.Gatt, P. Micallef

Department of Communications and Computer Engineering

University of Malta

Msida

{olgauc, cjdebo, ejgatt, pjmica}@eng.um.edu.mt

ABSTRACT

For the classical statistical classification algorithms the probability distribution models are known. However, in many real life applications, such as speech recognition, there is not enough information about the probability distribution function. This is a very common scenario and poses a very serious restriction in classification. Support Vector Machines (SVMs) can help in such situations because they are distribution free algorithms that originated from statistical learning theory and Structural Risk Minimization (SRM). In the most basic approach SVMs use linearly separating Hyperplanes to create classification with maximal margins.

However in application, the classification problem requires a constrained nonlinear approach to be taken during the learning stages, and a quadratic problem has to be solved. For the case where the classes cannot be linearly separable due to overlap, the SVM algorithm will transform the original input space into a higher dimensional feature space, where the new features are potentially linearly separable. In this paper we present a study on the performance of these classifiers when applied to speech classification and provide computational results on phonemes from the TIMIT database.

Categories and Subject Descriptors

I.5.1 [**Pattern Recognition**]: Models – Statistical.

General Terms

Algorithms, Performance, Theory.

Keywords

Speech recognition, Statistical Learning Theory, Support Vector Machine (SVM).

1. INTRODUCTION

In many practical learning algorithms we find many difficulties which manifest themselves in misclassifications during the learning phases[1]. Some of these complications are:

(i) The inefficiency of the learning algorithm itself, for example, the convergence to a local minima in gradient-descent based algorithms.

(ii) The size of the hypothesis that can become very large, thus requiring a large computational time making the solution impractical.

(iii) The available training set can be small. In this case the hypothesis class will become too rich, which leads to overfitting and hence poor generalization performance.

(iv) For a multidimensional search space, the learning algorithm requires a large number of parameters to be tuned, making the system difficult to use.

Support Vector Machines are learning systems that utilize a hypothesis space of linear functions in the implicitly defined feature space, trained using an algorithm from optimization theory that calculates a learning bias resulting from the statistical learning theory. The use of a kernel function ensures that the high dimensional feature space is used efficiently. The overfitting problem in the high dimensional feature space requires a learning bias which can be derived from the statistical learning theory. Optimization theory provides a clear characterization of the properties of the solution which leads to the implementation of efficient learning algorithms and makes sure that the hypothesis is represented in compact form. The convex learning bias will also ensure that local minima are not present so a solution can always be found efficiently even for training sets with thousands of examples[1].

The structure of this paper is as follows: In section 2 we present the theory behind the linear Support Vector Machine. This is followed by the concepts of the nonlinear Support Vector Machine in section 3. Finally sections 4 and 5 present some experimental results and a conclusion respectively.

2. LINEAR SVM

The reason behind using Support Vector Machines for classification is to find an efficient way of learning by separating Hyperplanes in the high dimensional feature space. The Hyperplanes must optimize the generalization bounds and the learning algorithm must be capable of dealing with thousands of training examples. The generalization theory gives a clear set of instructions on how to prevent overfitting by controlling the Hyperplane margin measures. Optimization theory can then be applied to give the necessary mathematical analysis to find the Hyperplanes which optimize these measures.

The Maximal Margin Classifier is the simplest Support Vector Machine[1]. It is based on the linear separability of the data in the high dimensional feature space and thus cannot be used in many real life applications. The Maximal Margin Classifier forms the basic building block of Support Vector Machines, i.e. to find the most separating Hyperplane in a proper kernel-induced feature space. This method is implemented by using a convex

optimization problem, minimizing a quadratic problem under linear inequality constraints.

Suppose that we have N training data points given by $(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$ where the input $x_i \in R^d$ and the output $y_i \in \{\pm 1\}$. The input is assigned a positive class if f(x) ≥ 0, and a negative class otherwise. Considering the case where f(x) is a linear function of x, f(x) can be written as,

$$f(x) = \langle w \cdot x \rangle + b \tag{1}$$

where $(w, b) \in R^n \times R$ are the parameters that control the decision function, and the decision rule is given by sgn(f(x)). As shown in Figure 1, a geometric interpretation of this hypothesis is that the input space is split into two parts by the Hyperplane,

$$\langle w \cdot x \rangle - b = 0 \tag{2}$$

The vector w defines a direction perpendicular to the Hyperplane while changing the value of b moves the Hyperplane parallel to itself. The parameters w and b are referred to as the weight and the bias terms respectively.

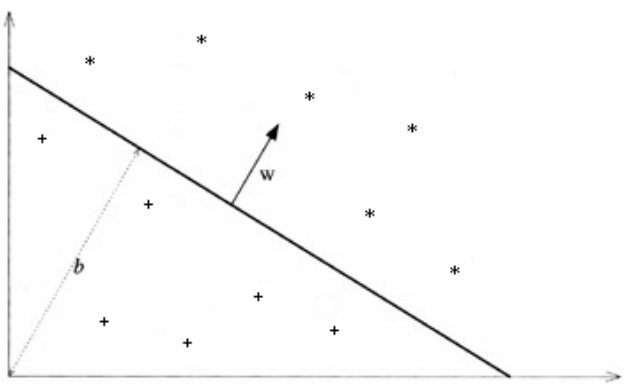

Figure 1. A Hyperplane (w, b) separating two classes

Further, we want this Hyperplane to have the maximum separating margin with respect to the two classes. Mathematically, we want to find the Hyperplane,

$$H : w \cdot x - b = 0 \tag{3}$$

and another two Hyperplanes,

$$H_1 : w \cdot x - b = +1 \tag{4}$$

$$H_{-1} : w \cdot x - b = -1 \tag{5}$$

parallel to it, with the restriction that there are no points between H_1 and H_{-1}, and that the distance between H_1 and H_{-1} is a maximum. Figure 2 shows an example for such a scenario where some positive examples are on Hyperplane H_1 while some negative examples are on Hyperplane H_{-1}. These examples are called Support Vectors because they define the separating Hyperplane.

The distance of a point on H_1 to H is given by:

$$\frac{\|w \cdot x - b\|}{\|w\|} = \frac{1}{\|w\|} \tag{6}$$

Therefore in order to maximize the distance separating the two

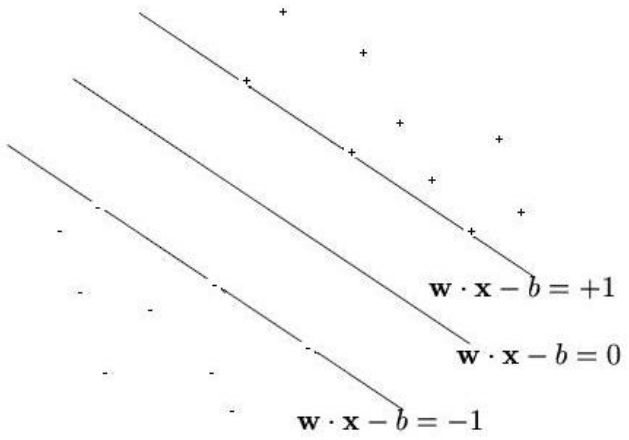

Figure 2. Maximal Margin Classifier

classes, we need to minimize $\|w\| = w^T w$ with the condition that no example is between the two Hyperplanes H_1 and H_{-1}.

Therefore,

$$w \cdot x - b \geq +1 \text{ for } y_i = +1, \tag{7}$$

$$w \cdot x - b \leq -1 \text{ for } y_i = -1. \tag{8}$$

Combining these two conditions we get,

$$y_i (w \cdot x_i - b) \geq 1 \tag{9}$$

Now our problem can be written as,

$$\min \frac{1}{2} w^T w \text{ subject to } y_i (w \cdot x_i - b) \geq 1 \tag{10}$$

However this is a convex, quadratic programming problem in a convex set. We can transform this optimization problem into its corresponding dual form by first considering the primal Lagrangian[1],

$$L(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i y_i (w \cdot x_i - b) + \sum_{i=1}^{N} \alpha_i \tag{11}$$

where $\alpha_i \geq 0$ are the Lagrange multipliers. Instead of solving this equation, we can solve the Wolfe dual form by maximizing the function $L(w, b, \alpha)$ subject to the constraint that the gradients of $L(w, b, \alpha)$ with respect to the primal variables w and b vanish, that is:

$$\frac{\partial L}{\partial w} = 0, \tag{12}$$

$$\frac{\partial L}{\partial b} = 0 \tag{13}$$

and the Lagrange multipliers $\alpha \geq 0$. Solving equations (12) and (13), we get,

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{14}$$

and

$$\sum_{i=1}^{N} \alpha_i y_i = 0. \tag{15}$$

Substituting equations (14) and (15) into the function $L(w,b,\alpha)$ we find:

$$L = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{16}$$

In this form the primal variables w and b have been eliminated and we end up with only one variable α. When the Lagrange multipliers are solved, we can go back to (14) to determine w, and we can classify an example x with:

$$f(x) = \text{sgn}(\sum_{i=1}^{N} \alpha_i y_i (x_i \cdot x) + b) \tag{17}$$

3. NONLINEAR SVM

In cases where the surface which separates the two classes is not linear, we have to implicitly transform the data examples into another high dimensional space such that the data points will be linearly separable in that space. Let the transformation into the high dimension feature space be $\Phi(\cdot)$. The dual problem now becomes[1]:

$$L = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j) \tag{18}$$

The dot product in the high dimensional space is equivalent to a kernel function of the input space,

$$k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \tag{19}$$

Therefore, we do not need be explicit about the transformation $\Phi(\cdot)$. There are many kernel functions that can be used to solve this such as the radial basis function:

$$K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2} \tag{20}$$

SVMs can also be extended to allow for noise or imperfect separation, hence the name soft margin Support Vector Machines. We do not strictly require the total absence of points between the Hyperplanes H_1 and H_{-1}, but we penalize the examples that cross the boundaries with the finite penalty factor C. We have also introduced a positive slack variable $\xi_i \geq 0$ in an attempt to include points which lie outside the Hyperplane separating their family. Figure 3 illustrates graphically the concept of slack variables introduced for an imperfect separation. Therefore, the separating Hyperplanes become:

$$w \cdot x_i - b \geq +1 - \xi_i \text{ for } y_i = +1 \tag{21}$$

$$w \cdot x_i - b \geq -1 + \xi_i \text{ for } y_i = -1, \tag{22}$$

$$\xi_i \geq 0, \forall i \tag{23}$$

We add the penalizing term to the objective function, so that it becomes:

$$\min \frac{1}{2} w^T w + C \sum_i \xi_i \text{ subject to } y_i (w^T x_i - b) + \xi_i - 1 \geq 0 \tag{24}$$

With this change the corresponding Lagrangian becomes[1]:

$$\begin{aligned} L(w,b,\xi,\alpha,r) = {} & \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i \\ & - \sum_{i=1}^{N} \alpha_i [y_i (\langle x_i \cdot w \rangle + b) - 1 + \xi_i] - \sum_{i=1}^{N} r_i \xi_i \end{aligned} \tag{25}$$

The Wolfe dual problem can now be stated as:

$$\max L = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{26}$$

subject to,

$$0 \leq \alpha_i \leq C, \tag{27}$$

$$\sum_i \alpha_i y_i x_i = 0 \tag{28}$$

The Lagrange multipliers are now bounded by C instead of infinity. The solution is again given by:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{29}$$

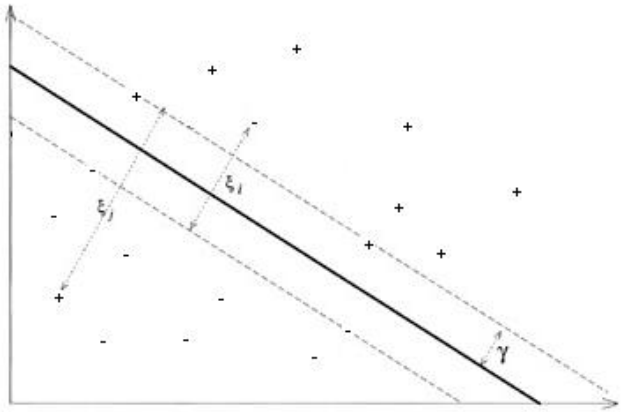

Figure 3. Slack Variable classification

4. EXPERIMENTAL RESULTS

Phonemes from the TIMIT database were segmented into frames of length 20 ms with a frame shift of 10 ms and filtered using a Hamming window. A Daubechies 10 filter was used to create a four level wavelet packet. The energies of the wavelet packets were calculated, thus obtaining a 16 dimensional feature vector.

The complete vowel dataset consisting of 20 phonemes from the TIMIT database were used for our experiments. The 20 class problem thus consisted of: /iy/, /ih/, /eh/, /ey/, /ae/, /aa/, /aw/, /ay/, /ah/, /ao/, /oy/, /ow/, /uh/, /uw/, /ux/, /er/, /ax/, /ix/, /axr/, and /ax-h/. 10, 000 samples were used to train the SVMs while 2,000 samples were used for testing.

During our experimentation the penalization factor C of the SVM was set to 60 while the value of σ for the radial basis kernel was set to 4. Table 1 presents some of the results showing the performance of the Support Vector Machine when vowels from the TIMIT Database were combined as many binary problems.

These results show that SVMs provide a good solution towards classification of binary phonemes. SVMs were also tested for multiclass applications, where the whole 20 phoneme set was used, but the results obtained were far from ideal. Further investigation in optimizing these tools is still necessary to apply satisfactorily these algorithms for speech recognition.

Table 1. Phonemes from TIMIT Database trained as Binary problems

Binary Problem	Precision(%)
'iy' – 'ih'	68.7 – 70.65
'ey' – 'aw'	86.1 – 93.5
'ih' – 'aa'	97.1 – 95.2
'iy' – 'ow'	96.5 – 95.2
'uw' – 'ax'	76.1 – 75.1
'ae' – 'aa'	94.0 – 84.1
'ao' – 'aa'	78.1 – 78.6
'ax' – 'ix'	89.1 – 87.5

5. CONCUSION

In this paper we evaluate the performance of a Support Vector Machine for phoneme recognition. The results obtained clearly show the classification power of Support Vector Machines in this application. Although good results have been achieved in the case of binary problems, future research work is required to extend these Learning systems for multiclass classification.

6. ACKNOWLEDGMENTS

This project is funded by a University of Malta's Research Grant.

7. REFERENCES

[1] N. Cristianini and J. S. Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning methods*, Cambridge University press 2000.

[2] A. Ganapathiraju, J.E. Hamaker, and J. Picone, "Applications of Support Vector Machines to Speech Recognition", *IEEE Trans. Signal Processing*, Vol. 52, No. 8, August 2004.

[3] C. Ma, M. A. Randolph and J. Dirsh, "A Support Vector Machine- Based Rejection Technique for Speech Recognition", *Proc. IEEE Int. Conf on Acoustics, Speech, Signal Processing 2001*.

[4] A. E. Cherif, M. Kohili, A.Benyetteou and M. Benyettou, "Lagrangian Support Vector Machines for Phoneme Classification", *Proc 9th International Conf. on Neural Information Processing*, ICONIP 02', Vol. 5.

[5] P. Clarkson, and P. J. Moreno, "On the use of support vector machines for phonetic classification", *Proc ICASSP March 99'*, Vol. 5 pp 585-588, 1999.

[6] J. C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", Microsoft Research, Tech. report MSR-TR-98-14. April 1998.

[7] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer – Verlang, New York 2000.

[8] R. Herbrich, *Learning Kernal Classifiers, Theory and Algorithms*, MIT Press 2002.

[9] B. Scholkopf and A. J. Smola, *Learning with Kernels, Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press 2002.