Dynamic Time alignment in Support Vectore Machines for Recognition Systems.

Shantanu Chakrabartty and Yunbin Deng
shantanu,yunbin@bach.ece.jhu.edu
Center for Language and Speech Processing(CLSP) and
Department of Electrical and Computer Engineering
The Johns Hopkins University
Baltimore, MD 21218 USA

***Abstract*— We propose a generic recognition system based on support vector machines. The system is capable of performing continuous time recognition, whether it be digits or phonemes. The heart of the system is a parameterized time aligned kernel which is tailored to discriminate each digit/phoneme and hence gives a novel way to estimate/improve performance. The scheme is fully scalable, however the complexity of the system increases with $O(N^2)$ which can be reduced by trading off the recognition performance.**

I. Introduction

Support Vector Machines (SVM) are universal learning machines which asymptotically converge to a bayes classifier, given sufficient amount of training data [2]. Based on the statistical theoretic framework most of the SVM classifiers designed assume the following

- The input data points are chosen independently of each other.
- The dimension of the input data are all equal.
- Kernel incorporates all the prior knowledge about the data.

In principle the role of the kernel in any support vector machine implementation is of utmost importance. A kernel implicitly computes an inner product between two data vectors in feature space. and hence it gives a metric of similarity or dissimilarity of two vector points. Hence any prior discriminatory knowledge that we have about the data, has to be embedded in the kernel.

In this paper we present a recognition system based on such a kernel that can handle misalignment of input data in time and as a result generate scores that reflect inherent time trajectory similarity. Previous work , covering phoneme recognition has been primarily to generate fixed dimension input vectors by concatenating speech frames in time and then using SVMs for classification . However we would like to generalize the above framework to handle large speech segments like digits into our study.

Section I gives a brief introduction to support vector machines. Section II explains how time aligned kernel fits into the SVM frame work. Section III explains the design of the time aligned kernel Section IV gives some performance evaluation for the recognition system, and Section V provides future insights and conclusions.

II. Support Vector Machines

In its basic form, an SVM classifies a pattern vector x based on the training data points x_ℓ and corresponding labels y_ℓ into classes $\{-1,+1\}$ based on the sign of y in

$$y = \sum_{\ell=1}^{L} \lambda_\ell \, y_\ell \, K(\mathbf{x}_\ell, \mathbf{x}) - b \tag{1}$$

where $K(\cdot,\cdot)$ is a symmetric positive-definite kernel function which can be freely chosen subject to fairly mild constraints [6]. Several widely used classifier functions reduce to special valid forms of kernel functions, like polynomial classifiers, multilayer perceptrons[1], and radial basis functions. The obtained support vectors are found to be almost invariant to the type of kernel function used [2], which indicates that the choice is not critical to classification performance.

The parameters λ_ℓ and b are determined by a linearly constrained quadratic programming problem [2], [3], which can be efficiently implemented by means of a sequence of smaller scale subproblem optimizations [5]. The key point for efficient implementation of the SVM is that typically only a small fraction of the λ_ℓ coefficients are non-zero. In other words, only a small set of *support vectors* ("prototypical" data points $\mathbf{x}_\ell$ with non-zero λ_ℓ) are needed in the classification of y.

Figure (1) shows the principle behind support vector machine where input data points which may be non-linearly separable are mapped into a higher dimensional space using a non linear mapping implicit in the kernel function , and then estimating the maximal margin hyper-plane in this feature space.

III. Problem Definition and Motivation dynamic time alignment

Given a sequence of vectors $\overline{x_1}, \overline{x_2}, .., \overline{x_l}, ..$ and its corresponding labels $y_1, y_2, ...y_l, ..,$ we wish to train a machine of the form

$$f(\overline{Z}) = w_o^T \overline{x}_n + w_1^T \overline{x}_{n-1} + .. + w_p^T \overline{x}_{n-p-1} + b \tag{2}$$

[1] with logistic sigmoidal activation function, for particular values of the threshold parameter only

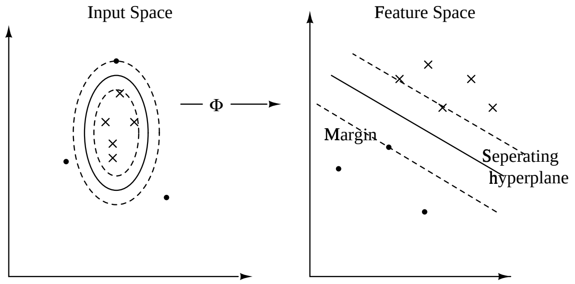

Fig. 1. *Principle behind support vector machine. The top figure is the original data space and the bottom figure is the feature space after the non-linear transformation. Note that the data is linearly separable in feature space.*

where w and b represents the parameters to be estimated from the training data . Thus the output of the SVM at time instant n not only depends on the current feature vector $\overline{x}_n$ but also on the past p vectors.

By minimizing the cost function as in a soft margin SVM classifier framework we obtain the following primal constrained cost function

$$min \quad L = \frac{1}{2}\sum_{k=0}^{p}|w_k|^2 + C\sum_{n=1}^{l}\eta_n \tag{3}$$

$$y_n(\sum_{k=0}^{p} w_k\overline{x}_{n-k} + b) \geq 1 - \eta_n \tag{4}$$

$$\eta_n \geq 0 \tag{5}$$

and its corresponding dual

$$maxM = \sum_{n=1}^{N}\alpha_n - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j(\sum_{k=0}^{p}\overline{x}_{i-k}^T\overline{x}_{j-k}) \tag{6}$$

$$\sum_{i=1}^{N}\alpha_i y_i = 0 \tag{7}$$

$$0 \leq \alpha_i \leq C \tag{8}$$

where α are the lagrange multipliers corresponding to each constraints.

As we can see that the dual leads the formulation of an auto-regressive type of kernel $\sum_{k=0}^{p}\overline{x}_{i-k}^T\overline{x}_{j-k}$ which measures the correlation between any two time instances i and j based on the current vector and its k^{th} history.

Our aim is to make the auto-regressive kernel more general to incorporate time alignment which will handle misalignment due to variability of data generation, like speech. Therefore one needs to introduce a warping function that generates proper alignment between data points before comparing them. We propose a dynamic time alignment kernel to handle the problem.

To generalize the above framework to handle speech segments, we assign cost C_n to labels at each time instant n. Then cost function in(5) is modified to

$$min \quad L = \frac{1}{2}\sum_{k=0}^{p}|w_k|^2 + \sum_{n=1}^{l}C_n\eta_n \tag{9}$$

This is a valid approach especially for speech recognition system where the labels are known only after the end of the speech segment. For simplicity we have assumed $C_n \in 0, 1$ in this paper. Moreover this approach fits in perfectly with digit recognition approach where we know the labels (digits) only after we have received the whole utterance. Hence a positive cost is attached only at the end of the digit utterance. This also has an added advantage of reducing the computational cost in training because zero cost eliminates most of the data points and leads to a sparser quadratic program in the dual (8).

IV. Framework for time aligned kernels

In design of time aligned kernel we impose the following constraints.

- The kernel should map two vectors of different lengths into a real value, similar vectors should get high value.
- The kernel should incorporate time alignment to account for local misalignment in feature vectors.
- The real value obtained , should represent the total score over the interval of comparison.

To perform local match between two feature vectors x_i and y_i we use a normalized gaussian kernel. If the norm of the two input feature vectors are kept constant then the gaussian kernel results in exponentiation of a dot product kernel given by the following equation

$$K(x_i, y_i) = exp(\frac{-(x_i - y_i)^T(x_i - y_i)}{\sigma}) \tag{10}$$

$$= exp(\frac{-x_i^T x_i}{\sigma})exp(\frac{-y_i^T y_i}{\sigma})exp(\frac{2x_i^T y_i}{\sigma}) \tag{11}$$

$$= Cexp(\frac{2x_i^T y_i}{\sigma}) \tag{12}$$

The primary advantage of using the exponentiation of the dot product kernel is that it reduces the noise floor by boosting the higher values.

To incorporate alignment into the kernel we use a shift windowing technique as follows

- Given two sequence of vectors $\overline{X}$ and $\overline{Y}$ we first choose the smallest length vector of the two. Without loss of generality denote this vector by $\overline{X}$.
- Taking the first feature vector x_1 of the chosen sequence $\overline{X}$, and compute the best local match of this vector in sequence $\overline{Y}$ over a window of predetermined size denoted

by p units. Initially the window starts from y_1. Let the best match be denoted by y_k where $1 \leq k \leq p$.

- Select the next feature vector x_2 and repeat the above procedure except that the match window now starts from index k instead of 1 as previously.

- Repeat the above procedure till we have obtained two vectors $\overline{X}$ and $\overline{Y}^{new}$ which are of the same length. We then sum up their local scores and normalize them using equation (13).

$$K_t(\overline{X}, \overline{Y}) = C/N \sum_{m}^{N} exp(\frac{2x_i^T y_i^{new}}{sigma}) \quad (13)$$

V. SVM DIGIT RECOGNIZER DESIGN USING TA-KERNEL

We followed a one-vs-one approach to design classifiers that are trained to discriminate between any two digits ,hence resulting in 45 SVMs . We chose this approach because

- Different digits require different kernel parameters , σ and window size for better discrimination. For example to discriminate digit one and four, the detection of the fricative /f/ carries lot of importance. Hence we need to choose a smaller window size for this particular machine. This results in designing better kernels for each machine which in turns reduces the number of support vectors and hence training time and generalization error.

- Each individual SVM is trained on a small subset of total data. Hence training is much faster.

- Each of the classifier performance can be boosted by using a voting algorithm at the output stage.

We then combined the 45 machines into a voting stage as shown in (2). The voting algorithm that we use assigns fixed weights to the output of each SVM based on generalization performance of individual machines. For simplicity we chose the generalization error proportional to the fraction of support vectors $(S_{i,j})$ for a machine trained to discriminate between digit i and j. The output D of the output stage is given by

$$D = \arg max_i O_i \quad (14)$$

$$O_i = \sum_j w_{i,j} f_{i,j} \quad (15)$$

$$w_{i,j} = sgn_{i,j} \frac{exp(-S_{i,j}/T)}{\sum_{k->i} exp(-S_{i,j}/T)} \quad (16)$$

where T is the total data used to train each machine and $k- > i$ denotes the machines that have been trained with digits corresponding to i. $sgni, j$ denotes the polarity of the output i and $f_{i,j}$ corresponds to the SVM output.

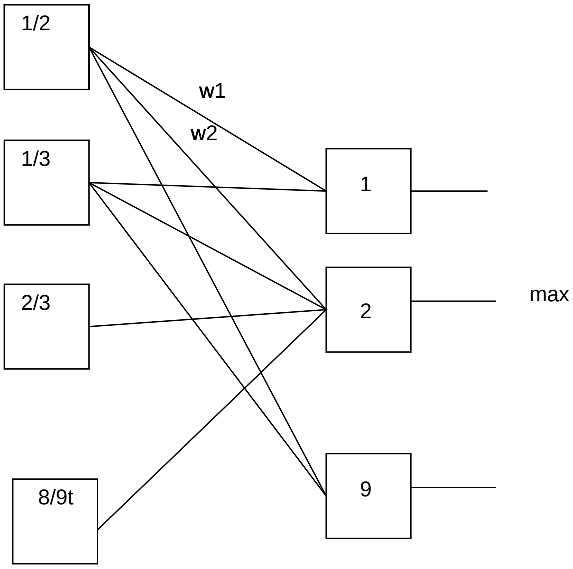

Fig. 2. *Recognizer architecture used for, where each one-vs-one classifer is connected to the output stage by weights that are determined by the generalization performance of each machines.*

VI. EVALUATION AND RESULTS

For our experiments we chose 100 utterances of each digit from one to nine. We then obtained 12 mel-cepstra coefficients for a windowed speech segment of 25ms duration. The data was then separated into 80 instances of training data and 20 instances of test data. Out of the 80 instances of training data 40 of the utterances were from men and rest from women. We then obtained 12 mel-cepstra coefficients for a windowed As we will show that the time aligned kernel is able to distinguish between these male and feminine utterances. Figure (3) shows the kernel matrix used for training the SVM. The first 40 digits were from utterance of one and the rest were from utterance *two*. Moreover the gray scale used for depicting the matrix assigns lighter color to higher values and darker for lower values. As we can see that the matrix is approximately in a block diagonal form, which means that the kernel is able to project the utterances onto two opposite sides of a hyper-plane and it is upto the maximal margin property of the SVM formulation to find the optimal one.

Table I and II gives the individual performance of each one-vs-one classifier. For this preliminary work we chose the tuning parameter (C , σ) to be constant amongst all the classifier. Hence these figures can be greatly improved.

Once all the individual one-vs-one classifier is combined through an output voting stage we get a higher recognition error rate as expected. However we believe by careful tuning of individual machines these figures can be greatly improved. Currently without proper tuning of the kernel parameters we are able to achieve 78performance.

TABLE I

Respective estimated generalization performance for individual one-vs-one classifiers. Note the generalization performance is the fraction of support vectors for that machine

1	2	3	4	5	6	7	8	9
x	0.26	0.3	0.45	0.44	0.31	0.29	0.27	0.4
x	x	0.33	0.3	0.27	0.42	0.32	0.32	0.23
x	x	x	0.35	0.31	0.35	0.35	0.34	0.35
x	x	x	x	0.32	0.32	0.32	0.33	0.3
x	x	x	x	x	0.22	0.35	0.25	0.45
x	x	x	x	x	x	0.4	0.31	0.26
x	x	x	x	x	x	x	0.25	0.33
x	x	x	x	x	x	x	x	0.26

TABLE II

Respective Error rates on test set for individual one-vs-one classifiers. Note the high error rates for some of the classifiers which is due to improper choice of tuning parameters

1	2	3	4	5	6	7	8	9
x	0.0	0.05	0.65	0.7	0.15	0.15	0.1	0.2
x	x	0.25	0.1	0	0.65	0.2	0.55	0
x	x	x	0.15	0.05	0.1	0.1	0.2	0.15
x	x	x	x	0.25	0.3	0	0.2	0.1
x	x	x	x	x	0.05	0.2	0	0.8
x	x	x	x	x	x	0.75	0.25	0.35
x	x	x	x	x	x	x	0.25	0.15
x	x	x	x	x	x	x	x	0.15

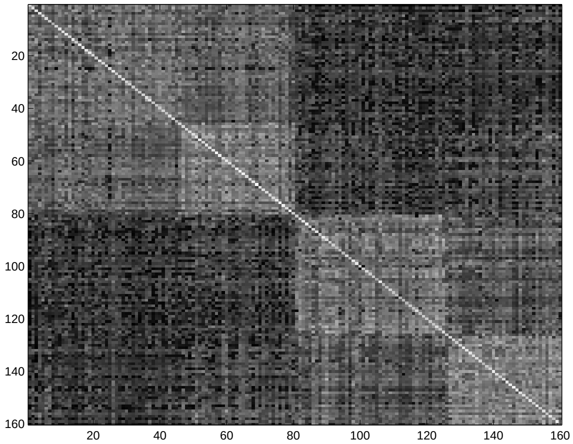

Fig. 3. *Kernel matrix plot for 80 utterances of digit 1 and 80 utterances of digit 2. Note the block diagonal form of the matrix depicting the discriminative power of the kernel. Also note the sub-blocks in each block diagonal which corresponds to the utterances from males and females.*

VII. Conclusions and Future Work

We proposed a novel approach to design recognition system where in individual elements can be tuned to optimize the overall performance. Since these results are all preliminary, there is lot of scope of improvement. We enlist the following items for future study.

- Optimal way to tune the parameters of individual one-vs-one machine to optimize the overall performance.
- Better voting algorithm that can minimize the effect of machines that have poor generalization performance.
- Apply the above method to phoneme recognition task where the role of the tuning parameters would be more important to discriminate between fricatives , stops and vowels. We believe because we have a bigger training and test set using TIMIT database we would be able to get more reliable estimate of the performance of this approach.

References

[1] F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, Cambridge, 1999.

[2] V. Vapnik, *The Nature of Statistical Learning Theory*, Second Edition, Springer, 1999.

[3] B. Scholkopf, C. Burges and A.Smola, eds., *Advances in Kernel Methods-Support Vector Learning*, MIT Press, Cambridge 1998.

[4] Girosi, F., Jones, M. and Poggio, T. “Regularization Theory and Neural Networks Architectures,” *Neural Computation*, vol. **7**, pp 219-269, 1995.

[5] Osuna, E., Freund, R., and Girosi, F., “Training support vector machines: An application to face detection,” in *Computer Vision and Pattern Recognition*, pp 130-136, 1997.

[6] Boser, B., Guyon, I. and Vapnik, V., “A training algorithm for optimal margin classifier,” in *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pp 144-52, 1992.