Two Sequence Kernels for the Classification of Speech Data

J. Goddard[1] , *A. E. Martínez*[1] , *F. M. Martínez*[1] , *H.L. Rufiner*[2,3]

[1] Depto. de Ingeniería Eléctrica, U.A.M-I, Mexico

[2] Lab. Cibernética, F.I.-U.N.E.R., Argentina

[3] SINC, F.I.C.H.- U.N.L., Argentina

Abstract

The most successful classifiers for automatic speech recognition are those which can handle sequential data, such as sequences of fixed dimensional vectors. In the last few years, alternatives to hidden Markov models, the *de facto* classifier, have been considered. The search for alternative classifiers has to do with deficiencies encountered in the standard scheme, such as a degradation in performance with noise. One particular classifier that has been successfully applied to sequential data is the support vector machine. In this paper, two sequence kernels are considered, and their performance compared to various other classifiers, on an isolated digit speech recognition task.

1. Introduction

The majority of classifiers tend to work with non-sequential data. This can cause problems in such applications as automatic speech recognition, biosequence classification, and text categorization, where the data is naturally variable-lengthed and sequential in nature. Different methods have been proposed to handle sequential data, either by adapting an existing classifier, e.g. using a fixed-sized feedforward neural net and windowing the input, or designing the classifier from the outset to handle sequential data, e.g. dynamic time warping and hidden Markov models (hmm). The field is one of active research [1].

Since their creation, about fifteen years ago, support vector machines (svm) have given excellent classification results on a wide variety of tasks, and have an impressive theoretical foundation, which goes someway to showing why they are able to generalize so well. Initially, their application was essentially to non-sequential data, however in recent years ways have been found to extend their application to sequential data.

Perhaps the first example of this was in [2], where the Fisher kernel was introduced and defined from a generative model, such as a hmm. It was also shown, than under certain conditions, the svm using this kernel was guaranteed to give results that were no worse than the original generative model. Since then, different svm have been applied to automatic speech recognition, some adapting an svm, such as [3,4,5], and others by introducing new kernels to work with the svm [6,7,8]. Usually the application has been to phoneme classification.

Further research is required in order to decide whether these methods will provide a viable alternative to the standard hmm classifier for sequential data tasks. In the present paper, two sequence kernels are considered and their performance is compared to various other classifiers, on an isolated digit speech recognition task. The sequence kernels are also derived from generative models, specifically Gaussian mixture models (gmm), and were motivated by [8].

The paper is organized as follows: in the next two sections brief discussions of the data and the methods used in the paper are given. The results are then presented followed by some conclusions.

2. Data

The data used for the experiments reported in the paper consists of isolated Spanish digits obtained from 6 young Mexican adults, 5 male and 1 female, and were previously employed in [9]. The recordings were made in a normal office environment and the recording quality varied amongst the utterances. A sampling frequency of 16 kHz was used and the waveforms were converted to vectors with 13 coefficients, 12 mfccs and log energy, using frames of length 256 with an overlap of size 128. Roughly 66% of the data was used for training and the rest for testing. In the training set there were 396 digits containing a total of 19,718 vectors, and the test set had 197 digits with 9,896 vectors.

3. Methods

Gmm and svm were used as the classifiers, and were applied both in a framewise fashion and using the sequential vectors corresponding to each digit. In the case of svm classifiers, three different kernels were used, two of them derived from the gmm. The classifiers considered are now described.

3.1. Gmm classifiers

A mixture model, M, consists of a weighted sum of N Gaussian densities N(μ,Σ), each parameterized by a mean vector μ and a diagonal covariance matrix Σ. The gmm therefore had the form:

$$P(x \mid M) = \sum_{i=1}^{N} a_i \frac{1}{(2\pi)^{13/2}\left|\Sigma_i\right|^{1/2}} \exp[-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)] \quad (1)$$

where *P(x / M)* is the likelihood of input vector $x \in \Re^{13}$ given the mixture model, M. The coefficients, a_i, are the mixture weights, which are constrained to be nonnegative and must sum to one.

The parameters a_i, μ_i, and Σ_i for i = 1,..., N of M were estimated using the netlab software [10**] on suitable data. Ten separate gmm were estimated, one for each digit class, using training data from the corresponding class.** The gmm all had the same number N of Gaussian densities.

For the case of framewise classification, a vector x was assigned to the digit class whose mixture model, M, had a likelihood, *P(x / M),* with maximum value.

For a given vector sequence $X = \{x_1, x_2, ..., x_m\}$, the likelihood of the sequence X, given the mixture model M, was calculated under the assumption of their independence as:

$$P(x \mid M) = \prod_{k=1}^{m} P(x_k \mid M) \quad (2)$$

A vector sequence X was assigned to the digit class whose likelihood was maximum.

3.2. Svm classifiers

The basic idea of svm classifiers, for a two class classification problem, is to map the data in each class into linearly separable sets in a higher dimensional inner product vector space, called the feature space. A separating hyperplane is then found in feature space which maximizes the minimal distance, known as the margin, between the hyperplane and the closest points of the classes to it. New patterns are then classified according to the side of the hyperplane they are mapped to. If ϕ is the transformation, then the function k, is defined using the inner product <,> in feature space by:

$$k(x,y) = <\phi(x),\phi(y)> \quad (3)$$

k is called a kernel. The kernel is in fact defined on the original data, and in earlier applications of svm classifiers was often defined implicitly, without further mention of the transformation; conditions can, in fact, be given on a function which assure that the function satisfies (3) i.e. is a kernel (c.f. [11]). Although the explicit use of the transformation ϕ is not required, it does enable us to gain a geometrical understanding into the svm method. Further, the two sequence kernels used in the paper are defined with explicitly formulated transformations.

One way to formulate the two-class svm classifier mathematically is through the following quadratic optimization problem [9, 12]: let D = $\{x_1,x_2,\ldots,x_m\}$ be a data set where each pattern x_i belongs to one of two classes $y_i \in \{-1, +1\}$, and k a kernel. It can be shown that by maximizing the following objective function:

$$W(\alpha_1,\ldots,\alpha_m) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j k(x_i, x_j) \qquad (4)$$

subject to the constraints

$$0 \le \alpha_i \le C, \ \sum_{i=1}^{m}\alpha_i y_i = 0$$

a suitable hyperplane can be found in feature space which is used to perform the classification. The constraint C is introduced in case the data is not linearly separable in feature space, perhaps due to the presence of noise or outliers, and is known as a soft margin. It represents a trade-off between maximizing the margin (as explained above) and minimizing the classification error on the training set. For an optimal solution $\{\alpha^0_i\}$ to (4), it is found that for the points which lie on or within the margin, or are incorrectly classified, $\alpha^0_i > 0$. These points are called support vectors. The rest of the points have $\alpha^0_i = 0$.

Classification of a test pattern x is given by designating its class to be the sign of:

$$\sum_{i=1}^{m} y_i \alpha^0_i k(x, x_i) + b \qquad (5)$$

where b is a constant whose value can be found from the optimal solution to (4) (c.f. [12]).

The svm classifiers above are defined for a two-class classification problem. One way to extend this to a ten-class classification problem is using a '1 vs. rest' strategy; that is, ten different two-class svm classifiers are found by successively solving (4) for the training data in one of the original classes and grouping the rest of the training data from the other classes. In this way, a different function in (5) is obtained for each class. A test pattern is then classified as belonging to the class with the maximum function value.

The software used for configuring the svm classifiers was libsvm [13].

3.3. The kernels

Three kernels were used for svm classifiers. The first was for framewise classification of the individual vectors, where a radial basis kernel (rbk) was employed and defined by:

$$k(x,y) = \exp(-\sigma||x-y||^2) \qquad (6)$$

where $x,y \in \Re^{13}$ and $\sigma \in \Re$.

For each of the sequence kernels, the following strategy was employed, based originally on the definition of the Fisher kernel [2], and specifically on that using a generative gmm model given in [8].

Firstly, ten gmm, M_i, for i=1,…,10, were trained for each digit class, using the training mfcc vectors of the class in question. Each gmm had the same number, N, of Gaussians and all had diagonal covariances. Given a vector sequence $X = \{x_1,x_2,\ldots,x_m\}$, from (2) we have:

$$\log P(X \mid M_i) = \sum_{k=1}^{m} \log P(x_k \mid M_i) \tag{7}$$

The kernel makes use of the first derivative of the log likelihoods with respect to the parameters of a generative model, which in turn is a function on the space of sequences X. Following [8], we take the generative model as the gmm M_i, and, in this case, the parameters are the mixture weights, means, and diagonal covariances. Hence, for each gmm M_i, we initially define this function, $\psi_i(X)$, on the space of sequences X by:

$$\psi_i(X) = \left[\frac{d}{da_{j^*}}, \ldots, \frac{d}{d\mu_{j^*}^{l^*}}, \ldots, \frac{d}{d\sigma_{j^*}^{l^*}} \right]^T \log P(X \mid M_i) \tag{8}$$

for j*=1,…,N and l*=1,…,13. Wan [8] gives the following formulas for calculating these values, which we shall include for completeness (note that the corresponding code can be obtained from his website): if M is a gmm with diagonal covariances, then defining R(i,j) as:

$$R(i,j) = \prod_{l=1}^{13} \frac{1}{\sigma_j^l \sqrt{2\pi}} \exp\left\{ -\frac{1}{2} \left(\frac{x_i^l - \mu_j^l}{\sigma_j^l} \right)^2 \right\} \tag{9}$$

so that the diagonal covariance gmm likelihood is:

$$P(x_i \mid M) = \sum_{j=1}^{N} a_j R(i,j) \tag{10}$$

Wan obtains the following:

$$\frac{d}{da_{j^*}} \log P(X \mid M) = \sum_{i=1}^{m} \frac{R(i,j^*)}{\sum_{i=1}^{N} a_j R(i,j)} \tag{11}$$

$$\frac{d}{d\mu_{j^*}^{l^*}} \log P(X \mid M) = \sum_{i=1}^{m} \frac{a_{j^*} R(i,j^*)}{\sum_{i=1}^{N} a_j R(i,j)} \cdot \frac{1}{\sigma_{j^*}^{l^*}} \left(\frac{x_i^{l^*} - \mu_{j^*}^{l^*}}{\sigma_{j^*}^{l^*}} \right) \tag{12}$$

$$\frac{d}{d\sigma_{j^*}^{l^*}} \log P(X \mid M) = \sum_{i=1}^{m} \frac{a_{j^*} R(i,j^*)}{\sum_{i=1}^{N} a_j R(i,j)} \cdot \left(\frac{(x_i^{l^*} - \mu_{j^*}^{l^*})^2}{(\sigma_{j^*}^{l^*})^3} - \frac{1}{\sigma_{j^*}^{l^*}} \right) \tag{13}$$

Finally, the transformation used to define the first sequence kernel is:

$$\phi(X) = (\psi_1(X), \psi_2(X), \ldots, \psi_{10}(X)) \tag{14}$$

and the kernel is defined as in (3) using the usual Euclidean inner product. In this case, as each $\psi_i(X)$ has dimension 27N, the Euclidean space is of dimension 270N.

The second kernel is similar to the above, but instead of using all the values for each function, $\psi_i(X)$, only the components corresponding to the partial derivatives with respect to the means are included. This reduces the dimension of the Euclidean space to 130N.

4. Results

Two framewise classifiers, one a gmm and the other a svm with rbk, where configured on the 19,718 mfcc vector training data, and tested on the remaining 9,896 vectors. Table 1 shows the results obtained choosing a different number of Gaussians for each model. The results for the framewise gmm on the test vectors is shown in the second column. These are averaged over 10 repetitions.

The task is digit classification, so although the classifier assigns a particular class to a test vector, some way has to be chosen to decide what digit a sequence of vectors represents. Both here, and for the framewise svm, the class assigned is that which is assigned to the majority of the test vectors in the sequence; this is shown in third column of Table 1. Although it is not shown in the table, the best classification rate achieved was with N = 30 giving 98.98% for the majority vote gmm and 59.33% with the framewise gmm.

In the case of the framewise svm, after some experimentation, C=10 and σ = 0.005 were the values selected for the parameters. With these values, the framewise classification rate was 66.94% , and the majority classification rate was 98.98%.

Sequential classifiers with gmm, and two svm with the two sequence kernels described in the previous section, were trained on the sequences of mfcc vectors corresponding to the 396 digits, and tested on the vector sequences of the 197 digits. This was repeated 10 times each, and the results shown in Table 2 are the averaged values. Each row corresponds to gmm with different numbers, N, of Gaussians. The third and fourth columns give the results for the svm using the first and second kernel, respectively, and taking the gmm with the corresponding number of Gaussians as its generative model.

5. Conclusions

In the present paper two sequence kernels were considered and applied to a problem of isolated digit classification. The results were also compared to a variety of other classifiers.

For the framewise classifiers, it is interesting to see how their results, in column 2 of Table 1, improve when even a simple scheme is chosen for extending them to the sequential data. The same is true for the sequential gmm of Table 2, although here for the same parameters (i.e. values of N) the results are significantly better. The framewise svm, in particular, achieves the best result with the majority voting scheme.

Table 1. Results of framewise gmm on the test digit data, averaged over 10 repetitions

Number of Gaussians N	% framewise gmm	% majority vote gmm
1	31.56	50.25
2	35.16	62.44
3	35.59	64.87
4	38.36	72.79
10	47.56	91.88

Table 2. Results of sequential classifiers on the test digit data, averaged over 10 repetitions

Number of Gaussians N	% gmm	% svm full	% svm means
1	69.54	90.86	87.81
2	81.22	93.91	94.26
3	83.96	95,99	96.19
4	88.68	94.92	96.19

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
J. Goddard, A. E. Martínez, F. M. Martínez & H. L. Rufiner; "Two Sequence Kernels for the Classification of Speech Data"
Proc. of the XII International Conference Speech and Computer (SPECOM 2007), oct, 2007.

For the sequential classifiers, good results are obtained for both of the svm when using gmm with a small number of Gaussians (less than four). The idea for using the second kernel was to see if the complete transformation was really required in order to obtain good classification rates; Table 2 indicates that the second kernel has a similar performance to that of the first.

Acknowledgements

The authors would like to thank CONACyT, Mexico, SECYT, Argentina and the S.E.P. (through their backing of cuerpos academicos), Mexico, for financial support.

References

1. *Thomas G. Dietterich.* Machine Learning for Sequential Data: A Review In T. Caelli (Ed.) Lecture Notes in Computer Science. Springer-Verlag. 2002

2. *T. Jaakkola and D. Haussler.* Exploiting generative models in discriminative classifiers. In Advances in Neural Information Processing Systems 11, 1998

3. *P. Clarkson, P.J. Moreno.* On the Use of Support Vector Machines for Phonetic Classification. In Proceedings of ICASSP'99, pp. 585-588, 1999

4. *Ganapathiraju, J. Hmaker, and J. Picone.* Hybrid SVM/HMM architectures for speech recognition, in Proc. of the International Conference on Spoken Language Processing, vol. 4, pp. 504–507. 2006

5. *Jesper Salomon, Simon King, and Miles Osborne.* Framewise phone classification using support vector machines. In *Proceedings International Conference on Spoken Language Processing,* Denver, 2002

6. *H. Shimodaira, K. Noma, M. Nakai, and S. Sagayama.* Support vector machine with dynamic time-alignment kernel for speech recognition, in Proc. of the Eurospeech, pp. 1841–1844, 2001

7. *N. Smith and M. Gales.* Speech recognition using SVMs, Advances in Neural Information Processing Systems 14. MIT Press, 2002

8. *Vincent Wan and Steve Renals.* Speaker Verification using Sequence Discriminant Support Vector Machines, IEEE Transactions on Speech and Audio Processing, vol. 13, no. 2, pages 203-210, 2005

9. *J. Goddard, A.E. Martínez, F.M. Martínez, H.L. Rufiner.* A Comparison of String Kernels and Discrete Hidden Markov Models on a Spanish Digit Recognition Task, in Proc. 25th Annual International Conference of the IEEE Engineering in Medicine And Biology Society, pp.2962-2965, Cancun, 2003

10. *Ian T. Nabney.* NETLAB: Algorithms for Pattern Recognition, Springer, 3rd printing, 2004

11. *B. Schölkopf.* The Kernel Trick for Distances, Microsoft Research Technical Report MSR-TR- 2000-51, 2000

12. *N. Cristianini and J. Shawe-Taylor.* Support Vector Machines, Cambridge University Press, 2000

13. *Chih-Chung Chang and Chih-Jen Lin.* LIBSVM : a library for support vector machines, 2001 Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm