EVALUATION OF KERNEL METHODS FOR SPEAKER VERIFICATION AND IDENTIFICATION

Vincent Wan, Steve Renals

Department of Computer Science,
University of Sheffield,
211 Portobello Street,
Sheffield S1 4DP, UK.
{*v.wan, s.renals*}*@dcs.shef.ac.uk*

ABSTRACT

Support vector machines are evaluated on speaker verification and speaker identification tasks. We compare the polynomial kernel, the Fisher kernel, a likelihood ratio kernel and the pair hidden Markov model kernel with baseline systems based on a discriminative polynomial classifier and generative Gaussian mixture model classifiers. Simulations were carried out on the YOHO database and some promising results were obtained.

1. INTRODUCTION

Speaker verification is concerned with determining whether an utterance has been spoken by the claimant or an imposter. The related task of speaker identification is concerned with labelling an unidentified speaker as one of a pool of enrolled speakers. Current standard approaches for text-independent speaker verification and identification are based on Gaussian mixture models (GMMs) [1, 2]. Discriminative classifiers have also proven to be successful for speaker verification [3]. In this paper we are concerned with using support vector machine (SVM) methods for text-independent speaker verification and identification.

SVMs seem well-suited to the classification-oriented tasks of speaker verification and identification. In particular, speaker verification may be posed as a binary classification problem. Some early work using the Switchboard database of conversational telephone speech was reported by Schmidt and Gish [4]. Fine *et al.*[5] recently applied GMMs and Fisher kernels to a speaker identification problem (using error correcting output codes to perform multiclass classification). Bengio and Mariéthoz [6] postprocessed the output of a GMM-based system using an SVM trained on the scores of a user model and an impostor model.

We present kernel-based methods for speaker verification and identification, comparing them with baseline systems based on discriminative polynomial classifiers and generative Gaussian mixture models. We have employed frame-based polynomial kernels and utterance-based dynamic kernels. The utterance-based approaches, using kernels for variable length sequences [7, 8], exploit an underlying generative model (GMM). We have carried out verification and identification experiments, using the YOHO database [9], and we report comparative results for the polynomial classifier, the GMM and the various SVM approaches.

2. BASELINE SYSTEMS

We have used two baseline approaches with which to compare the SVM. The polynomial classifier of Campbell and Assaleh [3] has been demonstrated to yield state of the art performance for speaker verification on the YOHO database. The GMM is a standard approach to both verification and identification, for example [1].

2.1. Polynomial classifier

The polynomial classifier method operates in a similar way to the SVM with a polynomial kernel. In both cases a linear boundary separates the data in a high dimensional feature space that is induced by a polynomial expansion.

The observed utterance, X, is denoted as a sequence of acoustic feature vectors (frames) $X = \{\mathbf{x}_1 \ldots \mathbf{x}_N\}$. Each frame is mapped explicitly into a high dimensional space via a polynomial expansion. For example, a second order polynomial would map the vector $\mathbf{x}$ onto

$$\Phi(\mathbf{x}) = \begin{bmatrix} 1 & x_1 & x_2 & \cdots & x_1x_2 & \cdots & x_1^2 & x_2^2 & \cdots & x_N^2 \end{bmatrix}^T \tag{1}$$

where x_i is the i^{th} component of $\mathbf{x}$ and the components of $\Phi(\mathbf{x})$ are the coefficients of a quadratic expansion. A linear classifier is constructed in the high dimensional space that best separates the features of the user, $\Phi(\mathbf{x}^{\text{user}})$, from those of the impostors, $\Phi(\mathbf{x}^{\text{imp}})$ by minimising the mean square error,

$$\sum_{j\in\{\text{user}\}} \sum_i (f(\Phi(\mathbf{x}_i^j)) - 1)^2 + \sum_{j\in\{\text{imp}\}} \sum_i f(\Phi(\mathbf{x}_i^j))^2 \,. \tag{2}$$

A general linear boundary may be expressed as $f(\Phi(\mathbf{x})) = \Phi(\mathbf{x}).\mathbf{w}$, and the adjustable parameters $\mathbf{w}$ are optimised by a method involving matrix decomposition.

The learned function $f(\Phi(\mathbf{x}_i))$ assigns a score to each frame. The score of a complete utterance, $S(X)$, is mean of the frame scores computed over the whole sequence

$$S(X) = \frac{1}{N} \sum_{j=1}^{N} f(\Phi(\mathbf{x}_j)). \tag{3}$$

The score is compared to a threshold, T. If $S(X) > T$ then the speaker is accepted as genuine, otherwise the speaker is an impostor.

Vincent Wan was sponsored by a Motorola Studentship.

2.2. Gaussian mixture models

We have used diagonal covariance GMMs as a baseline generative model. Additionally, GMMs are used as the underlying generative model in the dynamic kernels discussed in section 4.

The probability, $P(X|M)$, that the observation sequence, X, is generated by the model of the claimed speaker, M is used as the utterance score. It is estimated by the mean log likelihood over the sequence,

$$S(X) = \log P(X|M) = \frac{1}{N}\sum_{i=1}^{N} \log P(\mathbf{x}_i|M). \tag{4}$$

A simple GMM classifier compares the mean log likelihood to a threshold value to make its decision. More sophisticated variants use the likelihood ratio of the probability that the sequence is from the claimed speaker, $P(X|M)$, to the probability that the sequence was generated by a speaker independent (or global) model, $P(X|\Omega)$. Reynolds [2] took this approach by estimating $P(X|\Omega)$ using a pooled impostor model constructed from a set of background speakers selected individually for each claimant speaker using a log likelihood ratio distance measure.

3. POLYNOMIAL KERNELS

We have used SVMs with polynomial kernels [10, 11] as discriminative models for frame based speaker verification and identification. The polynomial kernel is given by

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i.\mathbf{x}_j + 1)^n \,. \tag{5}$$

In previous work [12] we described how this kernel can lead to an optimisation problem that has an ill-conditioned Hessian. The difficulties arise when the numerical value of the kernel becomes excessively large, for example, with large n or very high dimensional input data. The problem was overcome by a normalisation process where the data was mapped onto the surface of a unit hypersphere embedded in a space of higher dimension than the dimensionality of the feature vectors. Since the vectors in that space are of unit length then the dot products between them are constrained to the range ±1 preventing the kernel function generating extremely large values for any degree of polynomial.

One of the forms of the normalised polynomial kernel is,

$$K_{\text{norm}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2^n}\left(\frac{\mathbf{x}_i.\mathbf{x}_j + d^2}{\sqrt{(\mathbf{x}_i^2 + d^2)(\mathbf{x}_j^2 + d^2)}} + 1\right)^n, \tag{6}$$

which can be generalised to,

$$K_{\text{norm}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2^n}\left(\frac{K(\mathbf{x}_i, \mathbf{x}_j) + d^2}{\sqrt{(K(\mathbf{x}_i, \mathbf{x}_i) + d^2)(K(\mathbf{x}_j, \mathbf{x}_j) + d^2)}} + 1\right)^n, \tag{7}$$

where d is an adjustable parameter to achieve the correct normalisation. The general form of the normalised polynomial kernel can be expressed entirely in terms of dot products enabling the normalisation of any valid kernel that yields an ill conditioned Hessian.

Speaker verification by SVMs is achieved in exactly the same way as with polynomial classifiers in that an utterance score is computed from the mean classifier output for the sequence.

4. DYNAMIC KERNELS

The classification schemes described so far have computed frame-level scores and some form of averaging has been necessary in order to cope with utterances having different lengths. Recently, more principled approaches to dealing with variable length data by exploiting generative models have been developed [7, 8].

Recall that the key to SVM classification lies in the ability to compute the dot products between feature vectors. The methods described in this section allow us to effectively compute the “dot products” between two complete utterances regardless of their relative lengths. This implies that the SVM treats a whole utterance as a single feature instead of a series of fixed-length features. Thus the support vectors that define the decision boundary are complete utterances. The SVM optimisation selects the utterances from the training data that are hardest to classify. This is somewhat analogous to the automatic selection of background speakers in the construction of the pooled impostor model [2] discussed in section 2, although kernel methods do not automatically build global background models.

4.1. Fisher kernels

The method of Fisher kernels [7] encodes the variable length data into a single fixed length feature vector for classification by the SVM. Given a pre-trained generative model, the probability that a model M, parameterised by the vector θ, generates the sequence X is denoted by $P(X|M,\theta)$. A fixed length feature vector, $U_\theta(X)$ can be constructed by computing the derivatives of the log likelihood, $\log P(X|M,\theta)$, with respect to each of the parameters of the model. That is,

$$U_\theta(X) = \nabla_\theta \log P(X|M,\theta) \tag{8}$$

which is known as the Fisher score. Each component of $U_\theta(X)$ is the derivative with respect to one particular parameter. In our case, the generative model is a single state hidden Markov model (HMM) with the state output distribution described by a GMM. Thus the derivatives are with respect to the covariances, means and priors of the Gaussian mixture model.

The newly derived high dimensional features are in a non-Euclidean space and therefore the dot product must take into account the local Riemannian metric. A natural kernel is

$$K(X,Y) = U_\theta(X)^T I^{-1} U_\theta(Y) \tag{9}$$

where $I = E(U_\theta(X)U_\theta(Y)^T)$ is the Fisher information matrix. However, I is often a very large matrix. If there are n parameters in the model then I will have n^2 elements making its inverse difficult to compute. In such a situation we assume a Euclidean metric on the space and use I equal to the identity matrix.

4.2. Likelihood ratios

A variant of the Fisher kernel approach uses the ratio between the speaker model and a background impostor model [13]. In Gaussian mixture model classifiers the standard approach is to build two models: one model of the speaker and a second of a pooled set of impostors. The classifier takes the ratio of the likelihood estimates of the two models to obtain a better performance by increasing the likelihood of the speaker’s acoustics while reducing the likelihood of each impostor’s.

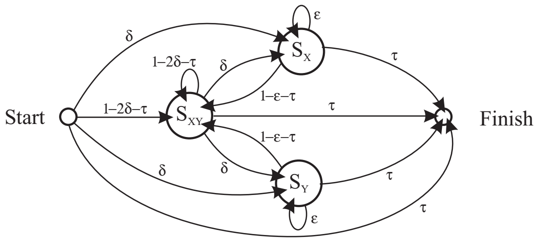

Fig. 1. A simple pair hidden Markov model

Let M_{spkr} denote the model of the speaker and M_{imp} denote the pooled impostor model. Each model is parameterised by θ_{spkr} and θ_{imp} respectively. A fixed length feature vector is obtained by,

$$U_\theta(X) = \nabla_\theta \log \frac{P(X|M_{spkr}, \theta_{spkr})}{P(X|M_{imp}, \theta_{imp})} \tag{10}$$

where θ is the combined set of parameters $\{\theta_{spkr}, \theta_{imp}\}$.

The kernel is computed using (9). Once again I is assumed to be the identity matrix since it is difficult to compute here.

4.3. Pair HMM kernels

The pair HMM is a type of hidden Markov model that has been widely used for biological sequence analysis. It is a model that generates two sequences of symbols simultaneously. The two sequences need not be of the same length. Figure 1 shows an example of a simple pair HMM used in bioinfomatics to construct probabilistic models of DNA sequences [14].

The pair HMM consists of a state S_{XY} that emits symbols for both sequences X and Y, a state S_X that emits for sequence X only (that is insertions in sequence X) and state S_Y that emits for sequence Y only. Just as in normal HMMs, there are start and end states that emit no symbols, matrices of emission and transition probabilities and an alphabet of symbols. The Viterbi alignment algorithm for pair HMMs is identical to that of normal HMMs except for an extra factor in the initialisation and termination of the recursion. The standard HMM assumptions apply. A pair HMM may be interpreted as performing a soft alignment. It may be more appropriate for text-dependent speaker verification when the lexical sequence — hence the state sequence — is known.

Watkins [8] proposed that the likelihood, $P(X,Y|M)$, that this model generates sequences X and Y together could be interpreted as a dot product in some space. In short, we can express the sequence pair likelihood as a dot product, $P(X,Y|M) = \Phi(X) \cdot \Phi(Y)$. The pair HMM kernel is thus simply

$$K(X,Y) = P(X,Y|M), \tag{11}$$

which can be computed by applying either the Viterbi or Baum Welch dynamic programming algorithms. In our case the emission probabilities of the states S_X and S_Y are calculated using a pre-trained Gaussian mixture model. The emission probability of the joint state, S_{XY}, $P(\mathbf{x},\mathbf{y}|M)$ is the product over the two Gaussian likelihoods, $P(\mathbf{x}|M)P(\mathbf{y}|M)$, that each symbol was emitted separately. By assuming independence between the emission probabilities in state S_{XY} we are introducing a strong assumption to simplify calculations.

Although it has been shown to be a dot product in some space, the numerical values generated by this kernel are generally badly scaled. They may vary by many orders of magnitude depending upon the lengths of the two sequences X and Y. This leads to a badly scaled Hessian that makes the optimisation almost impossible. The problem is overcome by applying the normalisation technique described in section 3.

5. EXPERIMENTS

We have evaluated the SVM approaches using text independent speaker verification and identification tasks. The YOHO database [9] was used in these experiments. This database contains clean speech recorded from 138 co-operative speakers. Each utterance takes the form of three double digit numbers read, "twenty-four thirty-seven fifty-one." Each speaker has 95 training utterances and 40 test utterances. The features were derived from the waveforms using 12th order LPC analysis and augmented with deltas. The features were normalised to zero mean and unit variance.

For the speaker verification experiments the database was split into two halves of 69 speakers each. The classifiers were trained on the first set of speakers then tested using the speakers of the second set as the impostors and vice-versa. This meant that classifiers were tested on impostors not seen during training giving a more realistic assessment of the overall performance. There are two types of error that can be made: a false acceptance where an imposter is incorrectly authenticated and a false rejection where the user is incorrectly identified as an imposter. The rates of each type of errors is dependent upon the value of the threshold, T. The equal error rate (EER) occurs when T is set appropriately such that the percentage of each type of error are equal.

It is trivial to perform speaker identification using the classifiers trained for verification. We use a one-versus-others scheme in which each classifier assigns a score to an utterance. The speaker is identified by the classifier with the highest score.

5.1. Polynomial based classifiers

To train the SVMs it was necessary to quantise the feature vectors using the k-means algorithm to create a smaller training set. Reducing the size of the training set reduces the number of support vectors in the final solution to a more manageable number. Each speaker was quantised from approximately 20,000 individual training vectors down to 100 centres resulting in a training set size of 6900 vectors for the SVMs. The parameter, d, in the normalised polynomial kernel was set to 1 for these experiments.

We compare the different polynomial classifiers in table 1. The SVM with the unnormalised kernel is by far the worst performer. The optimiser could not converge to a solution for unnormalised polynomial kernels of a degree higher than 4. Normalising the polynomial kernel without changing the degree of the polynomial yields a relative decrease in the average equal error rate of nearly 40%. Normalised higher order polynomial kernels allow the SVM to achieve a similar EER to the polynomial classifier, despite a significantly reduced amount of training data.

Curiously, as performance improves on the verification task performance on the identification task degrades. We suspect that this may be an artifact of either the vector quantisation or that the classifiers were trained independently so that classification is biased towards the SVMs that have slightly higher average scores.

Table 1. Performance of each classifier on text independent speaker verification and speaker identification tasks. The classifier labelled † was trained and tested under different conditions [2].

	Average	Speaker ID
Classifier	EER %	error rate %
Polynomial classifier, degree 3	0.38	1.01
SVM polynomial kernel, degree 4	1.72	4.24
SVM normalised polynomial, degree 4	1.05	6.12
degree 14	0.41	8.48
Basic GMM	1.08	0.50
SVM pair HMM kernel	1.05	13.50
SVM fisher kernel	0.68	0.96
SVM likelihood ratio kernel	0.43	0.78
GMM (likelihood ratio) † [2]	0.58	—

5.2. GMM-based classifiers

The basic GMM system consisted of a 512-component Gaussian mixture model with diagonal covariance matrices: the same system used to estimate the emission probabilities in the pair HMM, Fisher and likelihood ratio kernels. The result of the GMM likelihood ratio classifier is that obtained by Reynolds [2]. It is possible to build a global GMM of all the impostors in our training scheme. Undoubtedly using a larger training set for the impostors will improve performance. However, this leads to biased results (i.e. artificially low error rate) on small databases since the classifiers will have models for too many of the speakers.

The Fisher kernel maps each utterance to a 25,088 dimensional space prior to linear classification since the number of parameters in the single state HMM (GMM system) is 25,088. The likelihood ratio kernel maps to a space twice this dimensionality since it uses two of these GMMs. In the case of the pair HMM kernel, the Viterbi algorithm was used to compute the joint probability of the sequence pair. The parameters associated with the transition probabilities were determined separately for each utterance pair so as to maximise the joint probability. These parameters are dependent upon the lengths of the two sequences. Normalisation of the pair HMM kernel was achieved using (7) with $n = 1$. The parameter d was determined by approximately minimising the SVM's objective function on a small randomly chosen subset of the training data.

The performance of the dynamic kernels is mixed. The likelihood ratio kernel is clearly the best performer achieving an error rate in speaker verification close to that of the baseline polynomial classifier and the normalised polynomial kernel. The likelihood ratio kernel out performs the Fisher kernel as we would expect. More significantly both the Fisher and the likelihood ratio kernels outperform the basic GMM system from which they are derived on the speaker verification task but not on the speaker identification task.

The result of the pair HMM kernel gives no advantage over the basic GMM system. Closer analysis suggests the underlying model is incorrect for text independent speaker verification. The probability that a pair HMM generates a sequence pair depends upon the sequence lengths. Thus pair HMMs may be more suited to a text dependent speaker verification task where the model topology is predefined. Furthermore, there are some strong underlying assumptions applied to the kernel in order to simplify the calculations, which will undoubtedly affect the performance adversely. Once again improvement in performance is not reflected in the speaker identification task.

6. CONCLUSION

We have applied SVMs to speaker verification and identification, using a variety of kernels. On the YOHO database, we have found that it is possible to achieve state-of-the-art results using both static and dynamic kernels.

7. REFERENCES

[1] A. Martin, M. Przybocki, G. Doddington, and D. Reynolds, "The NIST speaker recognition evaluation - overview, methodology, systems, results, perspectives," *Speech Communication*, vol. 31, pp. 225–254, 2000.

[2] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Communication*, vol. 17, pp. 91–108, 1995.

[3] W. M. Campbell and K. T. Assaleh, "Polynomial classifier techniques for speaker verification," in *Proc. IEEE ICASSP*, 1999, vol. 1, pp. 321–324.

[4] M. Schmidt and H. Gish, "Speaker identification via support vector machines," in *Proc. IEEE ICASSP*, 1996, pp. 105–108.

[5] S. Fine, J. Navrátil, and R. Gopinath, "A hybrid GMM/SVM approach to speaker identification," in *Proc. IEEE ICASSP*, 2001.

[6] S. Bengio and J. Mariéthoz, "Learning the decision function for speaker verification," in *Proc. IEEE ICASSP*, 2001.

[7] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Advances in Neural Information Processing Systems 11*, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds. MIT Press, 1999.

[8] C. Watkins, "Dynamic alignment kernels," Tech. Rep. CSD-TR-98-11, Royal Holloway, University of London, 1999.

[9] J. P. Campbell Jr., "Testing with the YOHO CD-ROM voice verification corpus," in *Proc. IEEE ICASSP*, 1995, vol. 1, pp. 341–344.

[10] C. J. C. Burges, "A tutorial on suport vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 1–47, 1998.

[11] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

[12] V. Wan and W. M. Campbell, "Support vector machines for speaker verification and identification," in *Proc. Neural Networks for Signal Processing X*, 2000, pp. 775–784.

[13] Nathan Smith, Mark Gales, and Mahesan Niranjan, "Data-dependent kernels in SVM classification of speech patterns," Tech. Rep. CUED/F-INFENG/TR.387, Cambridge University Engineering Dept., 2001.

[14] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis*, Cambridge University Press, 1998.