SVMSVM: SUPPORT VECTOR MACHINE SPEAKER VERIFICATION METHODOLOGY

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 with the Fisher and score-space kernels are used for text independent speaker verification to provide direct discrimination between complete utterances. This is unlike approaches such as discriminatively trained Gaussian mixture models or other discriminative classifiers that discriminate at the frame-level only. Using the sequence-level discrimination approach we are able to achieve error-rates that are significantly better than the current state-of-the-art on the PolyVar database.

1. INTRODUCTION

Current state-of-the-art speaker verification systems are based on generative models, particularly Gaussian mixture models (GMMs) and hidden Markov models (HMMs), placed in a discriminative framework [1]. In these systems classifiers are trained to discriminate between individual frames of data then the likelihood scores of each frame are combined using an averaging step [2] or via an HMM to give an overall utterance score from which the authenticity of the speaker may be determined.

A discriminative classifier discards information that the objective function considers irrelevant. Discrimination at the frame-level is therefore not optimal for sequence classification since information relevant to sequence discrimination may be discarded inadvertently.

The support vector machine [3, 4] (SVM) coupled with a Fisher kernel [5] or, more generally, a score-space kernel [6, 7] enables direct discrimination between complete utterances. These kernel functions map a complete utterance onto a single point using a generative model. Such a representation enables any suitable classifier to discriminate between complete sequences. However, in order to represent a complete sequence adequately, the single data point must exist in a high dimensional space (determined by the number of parameters in a generative model for the sequence). Classification of high dimensional data is a task well suited to SVMs.

In this paper we extend previous work [8, 9] on applying the SVM to text independent speaker verification. Our experiments were performed on the PolyVar database [10] and we report error rates that are better than the current state-of-the-art by 25%.

Vincent Wan was sponsored by a Motorola Studentship.

2. BASELINE SYSTEMS

2.1. Gaussian mixture models

We used diagonal covariance GMMs as a baseline classifier. The same GMMs are used as the underlying generative model in section 3 to achieve a mapping from a variable length sequence of feature vectors to a single fixed length vector.

The probability, $P(X|M)$, that the observed sequence, $X = \{\mathbf{x}_1 \cdots \mathbf{x}_N\}$, is generated by the model of the client, M, is used as the utterance score, $S(X)$. It is estimated by the mean log likelihood over the sequence,

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

where $P(\mathbf{x}_i|M)$ is the client GMM likelihood for the i^{th} frame. To make a decision, the utterance score is compared to a threshold.

A better system incorporates a GMM of the impostors, Ω. Taking the ratio of the client model likelihood to the impostor model likelihood, the classifier takes into account the probability distribution of the impostors resulting in a more discriminative classifier [2]. The utterance score is the difference between the log likelihoods of the two models,

$$\begin{aligned} S(X) &= \log \frac{P(X|M)}{P(X|\Omega)} \\ &= \log P(X|M) - \log P(X|\Omega). \end{aligned} \tag{2}$$

This forms the basis of many state-of-the-art speaker verification systems.

2.2. GMM/SVM system

The combination of a GMM likelihood ratio (GMM-LR) with an SVM to incorporate a generative model into a discriminant framework was done by Bengio and Mariéthoz [11]. By Bayes decision rule, equation (2) is optimal so long as the client and impostors are well modelled. Bengio and Maritéhoz proposed that the probability estimates are not perfect and that a better version would be,

$$a \log P(X|M) - b \log P(X|\Omega) + c \tag{3}$$

where a, b and c are adjustable parameters. Given a set of training data and labels, these parameters may be estimated using any learning algorithm. Bengio and Mariéthoz used an SVM to learn these parameters. The input to the SVM is the two dimensional vector made up of the log likelihoods of the client and impostor models.

Table 1. Some examples of score operators

Operator	expression
first derivative	$\hat{F} = \nabla_\theta$
first derivative and argument	$\hat{F} = [\nabla_\theta, 1]^T$
first and second derivative	$\hat{F} = [\nabla_\theta, \mathrm{vec}(\nabla^2\theta)^T]^T$

3. DISCRIMINATIVE SEQUENCE CLASSIFICATION

The classifiers described in section 2 are trained to discriminate between individual frames of data. However, the classification of the sequence as a whole is desired. Discriminative classification of a sequence may be achieved if a discriminant classifier is allowed to observe the entire sequence. Classifying sequences directly is difficult since they have different lengths. A fixed-length representation of the sequence may, however, be classified directly.

The method for mapping from an arbitrary length sequence to a fixed length feature vector was first developed by Jaakkola and Haussler [5] and is known as the Fisher kernel. A more general approach known as score-spaces, from which the Fisher kernel may be derived, was developed by Smith *et. al.* [6, 7].

3.1. The score-space approach

The feature space to which utterances are mapped is called the score-space, so named because it is defined by and derived from the likelihood score of a generative model. Given a set of k generative models, $\{p_k(X|\theta_k)\}$, the generic formulation for mapping a sequence, X, to the score-space is,

$$\psi_{\hat{F}}^{f}(X) = \psi_{\hat{F}} f(\{p_k(X|\theta_k)\}) \tag{4}$$

where $f(\{p_k(X|\theta_k)\})$, a function of the scores of a set of generative models, is called the score-argument and $\psi_{\hat{F}}$ is the score-mapping that maps the scalar score-argument to the score-space using the score-operator, $\hat{F}$. The properties of the resulting score-space depends upon the choice of operator and argument that is used. Several options for score-operators were proposed by Smith *et. al.* [6, 7] and have been summarised in table 1.

Almost any function may be used as a score-argument. We shall show two specific cases that lead to the likelihood score-space kernel (more commonly known as the Fisher kernel) and the likelihood ratio score-space kernel.

3.2. The likelihood score-space (Fisher) kernel

By setting the score-argument to be the log likelihood of a single generative model, M, parameterised by θ,

$$f(\{p_k(X|\theta_k)\}) = \log P(X|M,\theta) \tag{5}$$

and choosing the first derivative score-operator from table 1 we obtain the mapping for the likelihood score space.

$$\psi(X) = \nabla_\theta \log P(X|M,\theta) \tag{6}$$

This mapping is also known as the Fisher mapping and was first developed and applied successfully for biological sequence analysis by Jaakkola and Haussler [5].

Using the first derivative and argument score-operator and the same score-argument the mapping becomes

$$\psi(X) = \begin{bmatrix} \nabla_\theta \log P(X|M,\theta) \\ \log P(X|M,\theta) \end{bmatrix} \tag{7}$$

The feature space defined by this mapping is identical to the Fisher mapping with one additional feature dimension: the log likelihood score itself. This mapping has the benefit that no information with respect to the original generative model, M, is lost in the transformation.

3.3. The likelihood ratio score-space kernel

An alternative score-argument is the log likelihood ratio of two generative models, a client model, M, and an impostor model, Ω,

$$f(\{p_k(X|\theta_k)\}) = \log \frac{P(X|M,\theta_1)}{P(X|\Omega,\theta_2)} \tag{8}$$

where $\theta = [\theta_1 \theta_2]$. The corresponding mapping using the first derivative score-operator is,

$$\psi(X) = \nabla_\theta \log \frac{P(X|M,\theta_1)}{P(X|\Omega,\theta_2)} \tag{9}$$

and using the first derivative with argument score-operator,

$$\psi(X) = \begin{bmatrix} \log \frac{P(X|M,\theta_1)}{P(X|\Omega,\theta_2)} \\ \nabla_\theta \log \frac{P(X|M,\theta_1)}{P(X|\Omega,\theta_2)} \end{bmatrix}. \tag{10}$$

The likelihood ratio score-space is motivated by the likelihood ratio GMM (GMM-LR) classifier described in section 2.1. In the same way that the GMM-LR is a more discriminative classifier than a single GMM alone, then so too should the corresponding score-space kernel.

4. NORMALISING SEQUENCE KERNELS

Classifying the features, $\Psi(X)$, derived from the transformation described in section 3 directly will not immediately yield a good performance when using SVMs. We have used two stages of normalisation in order to make this method work. The first stage is to whiten the data in the feature space, which involves normalising the vectors, $\Psi(X)$ to zero mean and unit variance. The second stage may be interpreted as a Hessian preconditioning step and involves making a further nonlinear transformation to a higher dimensional space.

4.1. Feature space whitening

For a given score-space, the metric of the space is defined by the generative models and is non-Euclidean. To correctly compute the dot product in any space requires the Riemannian metric of the space. A kernel constructed from any of the above mappings is,

$$K(X,Y) = \psi(X)^T G \psi(Y) \tag{11}$$

where G defines the metric of the space. In the case of the Fisher mapping, G is the inverse Fisher information matrix.

$$G = (E(\psi(X)\psi(Y)^T))^{-1}. \tag{12}$$

assuming that $E(\psi(X)) = E(\psi(Y)) = 0$. In general, G is the inverse of the covariance matrix of the data in score space. This can be interpreted as a whitening step where the features in score space are normalised to zero mean and unit variance.

Correct whitening of the data relies on the ability to compute a full covariance matrix, which will not only normalise the scaling in each dimension but also make the principal component axes of the space orthogonal. However, the feature space dimensionality, being equal to the number of parameters in the generative model, may be large. If there were 10^5 parameters in the generative model (a small number by many standards) then the full covariance matrix would have 10^{10} elements. A covariance matrix of this size is impractical to compute and invert. As an approximation the data may be normalised by the diagonal covariance matrix so that the scale of each dimension is the same.

4.2. Spherical normalisation

The second stage of normalisation is a preconditioning step that involves a transformation that maps each feature vector onto the surface of a sphere. The technique, which we call spherical normalisation, was first used in [9] to enable high order polynomial kernel SVMs to be trained. It involves mapping each feature vector onto the surface of a unit hypersphere embedded in a space that has one dimension more than the feature vector itself. The specific method adopted in this paper is to augment each vector with an extra component, d, and then normalise by the L2 norm,

$$\Psi_{\text{sph}}(X) = \frac{1}{\sqrt{\Psi(X)^2 + d^2}} \begin{bmatrix} \Psi(X) \\ d \end{bmatrix}. \tag{13}$$

The feature vectors on a unit hypersphere are normalised to unit length but when d is set appropriately then no information is lost in the transformation. The effect is to make the dot product between any two vectors better scaled (dot products between unit vectors lie between minus one and plus one) and to make the Hessian diagonally dominant. Without the normalisation, the magnitudes of the dot products may vary greatly, especially when the dimensionality of the data is high, such as the case when the likelihood ratio score-space kernel is used. These large variations in magnitudes result in an ill-conditioned Hessian for which the quadratic programming (the method for optimising SVMs) solution is more difficult to find. Spherical normalisation alleviates this problem.

5. EXPERIMENTS

The PolyVar database [10] was used in our speaker verification experiments. The database was recorded over a telephone network and consists of 38 client speakers (24 male and 14 female) each with 85 utterances recorded in 5 sessions with 17 utterances per session. There are also 952 impostor utterances from 56 speakers, each contributing 17 utterances in a single session. A strictly defined protocol for training an testing was adhered to. There were approximately 1000 test utterances (including both client and unseen impostors) for each client speaker.

The speech was processed using perceptual linear prediction with a 32ms window and a 10ms time shift to obtain 12 cepstral coefficients and one energy term. These features were augmented by their first and second order derivatives yielding 39 dimensional feature vectors. Cepstral mean subtraction was applied to remove the effects of the communication channel. Silence frames within each utterance were segmented out using a pretrained MLP.

5.1. The GMM baselines

Two GMM baseline systems were used in this evaluation. The GMM and GMM likelihood ratio (GMM-LR) described in section 2.1 are used. The classifier that gives state-of-the-art results is the combined GMM/SVM system described in section 2.2. Results on the PolyVar database for the GMM-LR and the GMM-LR/SVM have already been published by Bengio and Mariéthoz [11]. To make a fair evaluation the baseline models were trained using exactly the same conditions used by Bengio and Mariéthoz.

The impostor Gaussian mixture model contained 1000 Gaussians. This number was obtained by Bengio and Mariéthoz using cross-validation. Models of various sizes were trained using 90% of the training data while the remaining 10% was used to test the models. The model with the highest likelihood on the validation data was chosen.

The number of Gaussians in the client model was found using ten-fold cross-validation. The training data was randomly partitioned into ten equally sized sets. Every client model was trained on nine of the sets and tested on the one remaining set. This was repeated ten times, each time with a different test set. Repeating the whole process using different numbers of Gaussians in the models Bengio and Mariéthoz found the model that yielded the highest overall likelihood had 200 components.

Using the GMM-LR, Bengio and Mariéthoz reported a text independent speaker verification result of 5.55% HTER on the PolyVar test data using 19 speakers. A 4.73% HTER was reported for the GMM-LR/SVM. These results were successfully replicated. The results obtained using 38 speakers are shown in table 2.

5.2. Score-space kernels

The number of training utterances per SVM for the Fisher and LR kernels was 85 client utterances and 952 impostor utterances making a total training size of 1037 utterances per SVM. The Fisher kernel used the derivatives of the client GMMs of the baseline system to achieve the score-space mapping. The likelihood ratio (LR) kernel used both client and impostor models of the baseline system. The feature space dimensionality of the Fisher and LR kernels were 15800 and 94800 respectively. Using the corresponding score-operator with argument includes the log likelihood or log likelihood ratio score of the GMMs making the dimensionalities 15801 and 94801 respectively. The high feature space dimensionality, particularly for the LR kernel, causes computational problems for SVMs.

To compute the Hessian efficiently the training data must be held in memory. The memory needed to store the training feature vectors is $94800 \times 1037 \times 8$ bytes or 750 MB, assuming double precision floating point arithmetic is used. The SVM optimisation matrices were therefore computed in small blocks leaving feature vectors not in use stored on hard disk. This allowed us to use the complete set of features derived from the GMMs for training. Since linear SVMs were used, the resultant vector could always be computed from the support vectors. This saves memory and computation during testing.

In experiments where spherical normalisation was used, the mapping onto the hypersphere of equation (13) was applied explicitly to the feature vectors. Since the components of the feature

Table 2. Results of the PolyVar experiments. The GMM baseline consisted of 200 diagonal covariance Gaussian components modelling the clients. The GMM likelihood ratio (GMM-LR) consists of the above plus a GMM with 1000 diagonal covariance Gaussian components modelling the impostors. The GMM/SVM is the state-of-the-art classifier, which uses the same client and impostor GMMs but computes a weighted log likelihood ratio. The spherical normalisation constant was set to one in instances where it was used. The half total error rates (HTER) and equal error rates (EER) are shown.

Classifier	% HTER	% EER
Baseline		
GMM	11.22	12.07
GMM-LR	5.53	6.12
GMM/SVM	5.37	5.94
Fisher kernel		
without argument	6.54	6.98
without argument & sph. norm	6.50	6.87
with argument	6.50	6.92
with argument & sph. norm	6.47	6.87
LR kernel		
without argument	5.13	5.55
without argument & sph. norm	3.72	4.03
with argument	5.03	5.55
with argument & sph. norm	3.71	4.03

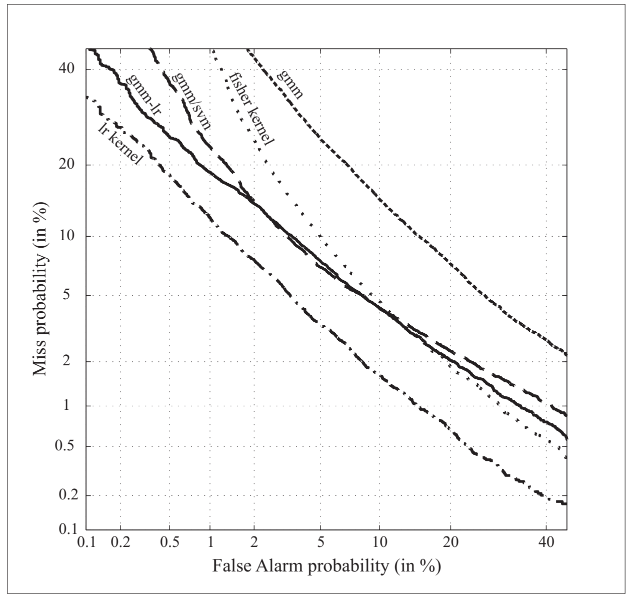

Fig. 1. The DET curves of the classifiers are shown: the baseline Gaussian mixture model (gmm), the baseline likelihood ratio GMM (gmm-lr), the likelihood score-space (without argument) kernel SVM (fisher kernel), the likelihood ratio score-space (without argument) kernel SVM (lr kernel) and the state-of-the-art combined GMM/SVM classifier (gmm/svm).

vectors were normalised to zero mean and unit variance we set the spherical normalisation parameter, $d = 1$.

Table 2 shows a summary of the test results and figure 1 the DET curves. It is clear from the results that sequence level discrimination using the LR kernel is beneficial to speaker verification. It can also be seen that spherical normalisation makes a significant difference when training SVMs on the high dimensional data of the likelihood ratio score-space kernel.

6. CONCLUSION

We have applied SVMs to speaker verification, using sequence kernels such as the Fisher kernel and the likelihood ratio kernel. On the PolyVar database, we reduced error rates by up to 33% relative compared to the state-of-the-art. The use of spherical normalisation was the key to achieving this result.

7. REFERENCES

[1] G. Doddington, M. Przybocki, A. Martin, and D. Reynolds, “The NIST speaker recognition evaluation – overview, methodology, systems, results, perspective,” *Speech Communication*, vol. 31, no. 2-3, 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] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

[4] 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.

[5] 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.

[6] N. Smith and M. J. F. Gales, “Using SVMs and discriminative models for speech recognition,” in *Proc. ICASSP*, 2002.

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

[8] V. Wan and S. Renals, “Evaluation of kernel methods for speaker verificationa and identification,” in *Proc. ICASSP*, 2002.

[9] 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.

[10] A. Constantinescu and G. Chollet, “Swiss polyphone and polyvar: Building databases for speech recognition and speaker verification,” in *Proceedings of The 3rd Slovenian-German and 2nd SDRV Workshop, Speech and Image Understanding*, April 24-26 1996.

[11] S. Bengio and J. Mariéthoz, “Learning the decision function for speaker verification,” in *IEEE International Conference on Acoustic, Speech, and Signal Processing*, 2001.