FUSION OF CROSS STREAM INFORMATION IN SPEAKER VERIFICATION

F. Alsaade, A. Malegaonkar and A. Ariyaeeinia

{F.Alsaade, A.Malegaonkar, A.M.Ariyaeeinia}@.herts.ac.uk

ABSTRACT

This paper addresses the performance of various statistical data fusion techniques for combining the complementary score information in speaker verification. The complementary verification scores are based on the static and delta cepstral features. Both LPCC (Linear prediction-based cepstral coefficients) and MFCC (mel-frequency cepstral coefficients) are considered in the study. The experiments conducted using a GMM-based speaker verification system, provides valuable information on the relative effectiveness of different fusion methods applied at the score level. It is also demonstrated that a higher speaker discrimination capability can be achieved by applying the fusion at the score level rather than at the feature level.

1. INTRODUCTION

The fusion of the complementary information obtained from the biometric data has been a research area of considerable interest. The efforts in this area are mainly focussed on fusing the information obtained using various independent modalities. For instance, a popular approach is to combine face and voice modalities to achieve a better recognition of individuals. The motivation behind this approach is that the independent information obtained using different modalities is thought to possess complementary evidences about the identity attributes of a particular person [1]. Hence combining such complementary information should be more beneficial than using a single modality. Various statistical fusion techniques have been developed for this task [2]. These range from using different weighting schemes that assign weights to the information streams according to their information content, to support vector machines which use the principle of obtaining the best possible boundary for classification, according to the training data.

Speaker verification is the task of matching the information obtained from a given test utterance against the model associated with the claimed identity. The process involves a binary decision depending on whether or not the match score exceeds a preset threshold. It is therefore desired that the metric adopted for this purpose can effectively discriminate between each true claimant and impostors. The most common approach to representing the registered speaker information is through training the Gaussian Mixture Models (GMM) on the speech feature data [3]. In GMM-based speaker verification, likelihood scores are used as matching metrics. Most of the verification systems use cepstral features to represent the speaker information. Static and delta cepstra obtained from speech represent two distinctive aspects of human vocal tract. Static cepstra represent the coarse aspects of vocal tract configuration under the assumption of being stationary, while delta coefficients represent the time varying (dynamic) information such as speaking style, and speaking rate [4]. This information can be derived from cepstra based on the linear prediction analysis (LPCC), or based on the perceptual processing on filter bank analysis (MFCC). Though delta coefficients are derived from static coefficients using a polynomial fit method, they represent a completely different level of information about the speaker and hence can be considered independent in terms of the information content.

Usually, static and delta cepstra are concatenated to represent a single feature vector for the task of speaker recognition. This is referred to as fusion at the feature level [5]. It is, however, reported in the literature that the fusion strategies work best at the score level [2]. Hence in this study, the fusion of the information obtained from static and delta cepstra is considered at the score level.

Various score level fusion schemes are evaluated in this study. Amongst these, the Support Vector Machine (SVM) is of particular interest. The use of Support Vector Machines in speaker verification has been considered relatively recently. To date, however, SVM have only been implemented at the feature level for speaker verification [6]. In this approach, the feature space is projected into some different hyperspaces so that the discrimination between the true and impostor speaker utterances is maximised. It has also been shown that combining SVM and GMM would lead to improvement in discrimination capability. [6]. In the present work, SVM are used at the score level (to combine the likelihood scores obtained from the static and delta cepstra) with the aim to maximise the separation of the true and impostor speakers. The rest of the paper is structured as follows. Section 2 gives the theory of various fusion schemes. Section 3 details the experimental setup. Section 4 discusses the results, whilst Section 5 presents the overall conclusions

2. FUSION TECHNIQUES

2.1. Weighted Average Fusion

In weighted average schemes, the fused score for each class (e.g. *j*) is computed as a weighted combination of the scores obtained from *N* matching streams as follows.

$$f_j = \sum_{i=1}^{N} w_i x_{ij} \quad , \qquad (1)$$

where, f_j is the fused scores for j^{th} class, x_{ij} is the normalised match score from the i^{th} matcher and w_i is the corresponding weight in the interval of 0 to 1, with the condition

$$\sum_{i=1}^{N} w_i = 1 \ , \qquad (2)$$

There are three sub-classes of this scheme, which primarily differ in the method used for the estimation of weight values.

2.1.1. Brute Force Search (BFS)

This approach is based on using the following equation [5].

$$f_j = x_j^1 * a + x_j^2 * (1-a) \quad , \qquad (3)$$

where f_j is the j^{th} fused score, x_j^p is the j^{th} normalized score of the p^{th} matcher, $p = 1{,}2$ and $0 \le a \le 1$.

2.1.2. Matcher Weighting using FAR and FRR (MW – FAR/FRR)

In this technique the performance of the individual matchers determines the weights so that smaller error rates result in larger weights. The performance of the system is measured by False Acceptance Rate (FAR) and False Rejection Rate (FRR). These two types of errors would be computed at different thresholds. Threshold that minimises the absolute difference between FAR and FRR on the development set is then taken into consideration. The weights for the respective matchers are computed as follows [7].

$$w_u = \frac{1-(FAR_u + FRR_u)}{2-(FAR_v + FRR_v + FAR_u + FRR_u)} \quad , \qquad (4)$$

where u=1, 2, v =1, 2 and u is not equal to v with the constraint $w_u + w_v = 1$

The fused score using different matchers is given as

$$f_j = w_u * x_j^u + w_v * x_j^v \qquad (5)$$

where, w_k is the weight from the k^{th} matcher, x_j^p is the jth normalised score of matcher p and f_j is the fused score.

2.1.3. Matcher Weighting based on EER (MW - EER)

The matcher weights in this case depend on the Equal Error Rates (EER) of the intended matchers for fusion. EER of matcher m is represented as E^m , m=1, 2 and the weight w_m associated with matcher m is computed as [8].

$$w_m = \frac{1}{E^m \left(\sum_{m=1}^{M} \frac{1}{E^m} \right)} \qquad (6)$$

Note that $0 \le w_m \le 1$, with the constraint given in (2). It is apparent that the weights are inversely proportional to the corresponding errors in the individual matchers. The weights for less accurate matchers are lower than those of more accurate matchers. The fused score is calculated in the same way as in equation (1).

2.2. Fisher Linear Discriminant (FLD)

In FLD, the linear boundary between the data from two classes is obtained by projecting the data onto the one dimensional space [9].

For data x, the equation of the boundary can be given as

$$h(x) = w^T x + b \ , \qquad (7)$$

where, w is a transformation matrix obtained on the development data and b is a threshold determined on the development data to give the minimum error of classification in respective classes. The rule for class allocation of any data vector is given by

$$x \in \left\{ \begin{matrix} \omega_1 \\ \omega_2 \end{matrix} \right. \ \text{if} \ w^{\mathrm{T}} x + b \left\{ \begin{matrix} > \\ < \end{matrix} \right. \ 0 \quad , \qquad (8)$$

2.2.1. Training the FLD

Given a range normalised data x_i from class Ci having a multivariate Gaussian distribution with the statistics $[m_i, S_i], i \in 1 \, and \, 2$, where S_i and m_i are a scatter matrix and mean for the particular class i. The scatter matrix is given as [9]

$$S_i = \sum_{k \in C_i} (x_k - m_i)(x_k - m_i)^T \quad , \qquad (9)$$

where, T is a transpose operation.

The overall within class scatter matrix S_W and the between class scatter matrix S_B are given by

$$S_W = \sum_{i=1}^{2} S_i \qquad (10)$$

$$S_B = (m_2 - m_1)(m_2 - m_1)^T \qquad (11)$$

The transformation matrix w is obtained using the equation

$$w = S_W^{-1} (m_2 - m_1) \qquad (12)$$

2.3. Quadratic Discriminant Analysis (QDA)

This technique is the same as FLD but is based on forming a boundary between two classes using a quadratic equation given as [10]

$$h(x) = x' A x + b' x + c \qquad (13)$$

For training data 1 and 2 from two different classes, which are distributed as $N[m_i, \Sigma_i], i \in 1 \, and \, 2$, the transformation parameters A and b can be obtained as

$$A = -\frac{1}{2}\left(\Sigma_1^{-1} - \Sigma_2^{-1}\right) \qquad (14)$$

$$b = \Sigma_1^{-1} m_1 - \Sigma_2^{-1} m_2 \qquad (15)$$

The classification rule in QDA is of the same nature as in FLD, only the equation is replaced appropriately.

2.4 Logistic Regression (LR)

The assumption in this technique is that the difference between log likelihood functions from two classes in data *x* is linear in *x* [9].

$$\log\left(\frac{p(x/\omega_1)}{p(x/\omega_2)}\right)=\alpha+\beta^T x \qquad (16)$$

Parameters in the above equation can be calculated with the maximum likelihood approach with an iterative optimisation scheme on some development data. Details can be found in [9].

The allocation rule for the test data is given as

$$x\in\begin{cases}\omega_1\\ \omega_2\end{cases} if \quad \alpha_0+\beta^T x \begin{cases}>\\ <\end{cases} 0 \qquad (17)$$

2.5 Support Vector Machines (SVM)

SVM is a classification technique based on forming a hyper plane that separates data from two classes with a maximum possible margin. SVM is based on the principle of Structural Risk Minimization (SRM) [11]. SRM principle states that better generalization capabilities are achieved through a minimization of the bound on the generalization error. The SVM uses the following function to map a given vector to its label space (i.e., -1 or +1)

$$f(x)=\mathrm{sign}\left(\sum_{i=1}^{l} a_i y_i k(x,x_i)+b\right) \qquad (18)$$

where $k(x,x_i)$ is a kernel function that defines the nature of the decision surface that separates the data, *x* is the input vector of a test set, x_i is the input vector of the ith training example, *l* is the number of training examples, *b* is a bias estimated on the training set, y_i is the class specific mapping label and a_i are the solutions of the following Lagrangian in the quadratic programming problem.

$$\mathrm{Q}(a)=\sum_{i=1}^{l} a_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} a_i a_j y_i y_j k(x_i,x_j) \qquad (19)$$

with the constraints,

$$\sum_{i=1}^{l} a_i y_i = 0 \qquad (20)$$

More details of this equation are given in [11]. In the resulting solution, most a_i are equal to zero, which refer to the training data that are not on the margin. The training examples with non-zero a_i are called support vectors, which are the input vectors that lie on the edge of the margin. Introducing new data outside of the margin will not change the hyper plane as long as the new data are not in the margin or misclassified. Therefore, the classifier must remember those vectors which define the hyper plane.

The kernel function $k(x,x_i)$ can have different forms. More details can be found in [11]. In this work, linear and polynomial kernel functions with a degree of 2 (quadratic) are used. These are given by following equations,

$$Linear{:}\,k(x,x_i)=x^T x_i \quad , \qquad (21)$$

$$Quadratic{:}\,k(x,x_i)=\left(x^T x_i+1\right)^2, \qquad (22)$$

2.6. Range-Normalisation Techniques

Range-normalisation is the task of bringing raw scores from different matchers to the same range. This is a necessary step in any fusion system as fusing the scores without such normalisation would de-emphasise the contribution of the matcher having a lower range of scores. Two different normalisation techniques have been evaluated in this paper [8].

2.6.1 Min-Max Normalisation (MM)

This method uses the following equation

$$x=\frac{n-\min(n)}{\max(n)-\min(n)} \quad , \qquad (23)$$

where, *x* is the normalised score, *n* is the raw score, and max and min functions specify the maximum and minimum end points of the score range respectively.

2.6.2. Z-score Normalisation (ZS)

This method transforms the scores having some Gaussian distribution to a standard Gaussian distributional form. It is given as

$$x=\frac{n-mean(n)}{std(n)}, \qquad (24)$$

Where, *n* is any raw score, and mean and std are the statistical mean and standard deviation operations.

3. EXPERIMENTAL SETUP

3.1. Speech Data

The speech data used in this work is from the TIMIT database. Material from all the 630 speakers is used. For each speaker, the utterances ‘sa1’ and ‘sa2’ are used for the development and testing respectively. The rest of the 8 utterances for each speaker are used for developing the speaker representation as a Gaussian Mixture Model (GMM) with 32 components.

3.2. Feature Extraction

The extraction of cepstral parameters is based on first pre-emphasising the input speech data using a first order digital filter with a coefficient of 0.95 and then segmenting it into 20 ms frames at intervals of 10 ms using a Hamming window. 16 LPCC coefficients are then obtained via a linear prediction analysis. For obtaining MFCC, speech spectrum for each frame is weighted by a Mel scale filter bank. The discrete cosine transformation of the log magnitude outputs of these filters gives the MFCC for that speech frame. For each type of cepstra, a polynomial fit method is used to obtain the delta coefficients [4].

3.3. Testing

The scores generated with the development utterances are first used to obtain the training parameters in various fusion techniques. True and impostor scores from static and delta streams are pooled and then normalised according to the chosen range-normalisation scheme. Parameters obtained in the fusion schemes are then used in the test phase to transform the normalised test scores according to the fusion scheme. The verification performance is then obtained on the transformed scores in terms of equal error rates (EER) via the DET curves

.

4. RESULTS AND DISCUSSIONS

The experimental results are presented in the following tables. It can be seen that (in most cases) the ZS normalisation is exhibiting more effectiveness than the MM normalisation. In some cases though, the two approaches provide comparable performance.

It can be observed that the way fusion techniques work for combining the static and delta features is not identical in the two considered cases of LPCC and MFCC. In the case of LPCC features, improvements are seen in majority of the fusion cases by fusing the scores from static and delta features as compared to the feature level concatenation. In some cases such as Linear SVM and LR, the results are even better than using the individual feature streams. Thus under this experimental setup, LR and Linear SVM give the best results for LPCC features. In the case of MFCC data, all of the fusion techniques except MW-EER indicate that score level fusion can give better performance than the feature level concatenation. But no fusion techniques for MFCC are seen to exceed the performance of the baseline MFCC static features. Thus the best results obtained in this case are still with MFCC static features.

Thus it can be said that the speaker verification systems can benefit through the score level fusion, but this depends on the types of feature as well as the normalisation method used.

Cepstra	EER %	Cepstra	EER %
LPCC static (s)	1.76	MFCC static (s)	2.06
LPCC delta (d)	39.64	MFCC delta (d)	38.89
LPCC (s + d)	2.44	MFCC (s + d)	3.23

Table 1: Baseline Results

EER %	BFS	MW (FAR/FRR)	MW - EER	FLD
LPCC (s + d)	2.17	3.17	2.16	10.64
MFCC (s + d)	2.21	4.45	2.45	2.27

EER %	QDA	LR	SVM Linear	SVM Poly
LPCC (s + d)	4.60	3.87	4.20	4.39
MFCC (s + d)	2.27	2.73	2.32	3.17

Table 2: Score Level Fusion (MM Normalisation)

EER %	BFS	MW (FAR/FRR)	MW - EER	FLD
LPCC (s + d)	1.74	2.70	2.06	2.85
MFCC (s + d)	2.22	3.83	2.30	2.27

EER %	QDA	LR	SVM Linear	SVM Poly
LPCC (s + d)	1.75	1.14	1.08	1.71
MFCC (s + d)	2.27	2.79	2.34	2.65

Table 3: Score Level Fusion (ZS Normalisation)

5. CONCLUSIONS

It can be concluded from this study that the combination of complementary information from the speech static and delta cepstra can improve the performance in the speaker verification. Improvements are of greater extent in the case of LPCC features. In this case, the fusion of the information at the score level is more effective than that at the feature level. Amongst various fusion methods considered, SVM approach has appeared to provide the best performance in terms of reducing error rates in speaker verification. Finally the ZS normalisation method exhibits better performance than MM normalisation for the fusion task.

6. REFERENCES

1. Fabio Roli et.al. “An Experimental Comparison of Classifier Fusion Rules for Multimodal Personal Identity Verification Systems”, Proc. Multiple Classifier Systems, Springer-Verlag, 2002, pp. 325-336.

2. Conrad Sanderson and Kuldip K. Paliwal, “Identity Verification using Speech and Face Information”, Digital Signal Processing, 2004.

3. D. Reynolds and R. Rose, “Robust Text Independent Speaker Identification using Gaussian Mixture Speaker Models”, IEEE Transactions on Speech and Audio Processing, 3 (1), Jan. 1995.

4. D. O'Shaughnessy, Speech Communication: Human and Machine, Addison -Wesley, 1987

5. Ariyaeeinia A. M. and Sivakumaran P., “Effectiveness of Orthogonal Instantaneous and Transitional Feature Parameters for Speaker Verification”, 29th Annual Carnahan Conference on Security Technology, 1995, pp. 79 – 84.

6. V. Wan and S. Renals, IEEE International Workshop on Neural Networks for Signal Processing 17 - 19 September 2003.

7. Y. Wang, T. Tan and A. K. Jain, “Combining Face and Iris Biometrics for Identity Verification”, Proceedings of Fourth International Conference on AVBPA, (Guildford, U. K.), pp. 805-813, June 2003

8. M. Indovina et.al., “Multimodal Biometric Authentication Methods: A COTS Approach”, Proc. MMUA 2003, Workshop on Multimodal User Authentication, Santa Barbara, CA, December 11-12, 2003.,pp. 99-106

9. C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, New York, 1996.

10. B. Flury, Common Principle Components and Related Multivariate Models, John Wiley and Sons, USA, 1988.

11. C.J.C. Burges, “A tutorial on support vector machines for pattern recognition”. Data Mining and Knowledge Discovery, 2(2),pp. 955-974, 1998