Fusion of Spectral Feature Sets for Accurate Speaker Identification

Tomi Kinnunen, Ville Hautamäki, and Pasi Fränti
Department of Computer Science
University of Joensuu, Finland
{tkinnu,villeh,franti}@cs.joensuu.fi

13th August 2004

Abstract

Several features have been proposed for automatic speaker recognition. Despite their noise sensitivity, low-level spectral features are the most popular ones because of their easy computation. Although in principle different spectral representations carry similar information (spectral shape), in practice the different features differ in their performance. For instance, LPC-cepstrum picks more "details" of the short-term spectrum than the FFT-cepstrum with the same number of coefficients. In this work, we consider using multiple spectral presentations simultaneously for improving the accuracy of speaker recognition. We use the following feature sets: mel-frequency cepstral coefficients (MFCC), LPC-cepstrum (LPCC), arcus sine reflection coefficients (ARCSIN), formant frequencies (FMT), and the corresponding delta-parameters of all feature sets. We study the two ways of combining the feature sets: feature-level fusion (feature vector concatenation), score-level fusion (soft combination of classifier outputs), and decision-level fusion (combination of classifier decision).

1 Introduction

Front-end or *feature extractor* is the first component in an automatic speaker recognition system. Feature extraction transforms the raw speech signal into a compact but effective representation that is more stable and discriminative than the original signal.

Speaker differences in the acoustic signal are coded in complex way in both *segmental* (phoneme) level, *prosodic* (suprasegmental) level and *lexical* level. Modeling of prosody and lexical features has shown great promises in automatic speaker recognition systems lately [19]. However, the segmental features are still the most popular approach because of their easy extraction and modeling.

In most automatic speaker and speech recognition systems, segmental features are computed over a short time window (around 30 ms), which is shifted forward by a constant amount (around 50-70 % of the window length). Two most popular features are *mel-frequency cepstral coefficients* (MFCC) and *linear predictive cepstral coefficients* (LPCC) [9]. These features are often augmented with the corresponding *delta features*. The delta features give an estimate of the time derivative of each feature, and therefore they are expected to carry information about vocal tract dynamics. Sometimes, the delta parameters of the delta parameters (*double-deltas*) are also used, as well as the *fundamental frequency* (F0). For each time window, the different features are simply concatenated into a one higher dimensional (around $d = 40$) feature vector.

Augmenting the static parameters with the corresponding delta parameters can be seen as one way to perform *information fusion* by using different information sources, in the hope that the recognition accuracy will be better. The vector level feature augmentation is denoted here as *feature-level fusion*.

Although feature-level fusion may improve recognition accuracy, it has several shortcomings. First, fusion becomes difficult if a feature is missing (e.g. F0 of unvoiced sounds) or the frame rates of the features are different. Second, the number of training vectors needed for robust density estimation increases exponentially with the dimensionality. This phenomenon is known as the *curse of dimensionality* [2].

An alternative to feature-level fusion is to model each different feature set separately, design a specialized classifier for this feature set, and combine the classifier output scores. Each of the different feature sets acts as an independent "expert", giving its opinion about the unknown speaker's identity. The *fusion rule* then combines the individual experts' match scores. This approach is referred here as *score-level fusion*.

Score-level fusion strategy can also be abstracted by hardening the decisions of the individual classifiers. In other words, each of the experts produces a speaker label, and the fusion rule combines the individual decisions e.g. by majority voting. We call this fusion strategy *decision-level fusion*.

In a previous work [11], we documented our implementation of an score-level fusion system that uses vector quantization (VQ) based classifiers. The system can be used for combining an arbitrary number of diverse feature sets varying in scale, dimensionality and the number of vectors. For each speaker and feature set, a codebook is trained using a clustering algorithm. In the recognition phase, features extracted from the unknown speaker are presented to the corresponding classifiers. Each vec-

tor quantizer computes average quantization distortion of the unknown sequence. Within each quantizer, the distortions are scaled so that they sum up to unity over different speakers. The scaled distortions are then weighted and summed to give the final combined match score. The weights are feature set depended, but same for all speakers.

Extensive experiments in [10] were carried out on two corpora, a 100 speaker subset of the American English TIMIT corpus [16] and a corpus of 110 native Finnish speakers, documented in [6]. There were some differences between the two corpora and feature sets, but these were relatively small; many of the feature sets reached error rates close to zero. Therefore, it seemed unnecessarily to experiment with different fusion strategies with these features since the individual features already performed so well. The reason for this is that the both corpora were recorded in unrealistic laboratory conditions. We have found out that the performance decreases radically in real-world conditions.

In this study, we have selected the spectral parameters that seem most promising in the light of the findings of [10]. We study these on a more realistic corpus, a subset of the 1999 Speaker Recognition Evaluation Corpus. We aim at studying whether different spectral feature sets can complement each other, and which one of the fusion strategies (feature, score, and decision-level) is most appropriate for VQ-based classification in practice.

2 Selected Spectral Features

2.1 Mel-Frequency Cepstral Coefficients

Mel-frequency cepstral coefficients (MFCC) are motivated by studies of the human peripheral auditory system. First, the pre-emphasized and windowed speech frame is converted into spectral domain by the fast Fourier transform (FFT). The magnitude spectrum is then smoothed by a bank of triangular bandpass filters that emulate the critical band processing of the human ear. Each of the bandpass filters computes a weighted average of that subband, which is then compressed by logarithm. The log-compressed filter outputs are then decorrelated using the discrete cosine transform (DCT). The zeroth cepstral coefficient is discarded since it depends on the intensity of the frame.

There are several analytic formulae for the mel scale used in the filterbank design. In this study, we use the following mapping [7]:

$$f_{\mathrm{mel}}(f_{\mathrm{Hz}}) = \frac{1000}{\log_{10} 2} \log_{10}\left(1 + \frac{f_{\mathrm{Hz}}}{1000}\right), \qquad (1)$$

having the inverse mapping

$$f_{\mathrm{Hz}}(f_{\mathrm{mel}}) = 1000\left(1 + 10^{\frac{\log_{10} 2}{1000} f_{\mathrm{mel}}}\right). \qquad (2)$$

First, the number of filters (M) is specified. Filter center frequencies are then determined by dividing the mel axis into M uniformly spaced frequencies and computing the corresponding frequencies in the hertz scale with the inverse mapping. The filterbank itself is then designed so that the center frequency of the mth filter is the low cutoff frequency of the $(m+1)$th filter. The low and high cutoff frequencies of the first and last filters are set to zero and Nyquist frequencies, respectively.

2.2 LPC-Derived Features

In addition to the MFCC coefficients, we consider the following representations that are computed via linear prediction analysis: *arcus sine reflection coefficients* (ARCSIN), *linear predictive cepstral coefficients* (LPCC), and formant frequencies (FMT).

The linear predictive model of speech production [17, 5] is given in the time domain:

$$s[n] \approx \sum_{k=1}^{p} a[k] s[n-k], \qquad (3)$$

where $s[n]$ denotes the speech signal samples, $a[k]$ are the *predictor coefficients* and p is the *order* of the predictor. The total squared prediction error is:

$$E = \sum_{n} \left(s[n] - \sum_{k=1}^{p} a[k] s[n-k]\right)^2. \qquad (4)$$

The objective of linear predictive analysis is to determine the coefficients $a[k]$ for each speech frame so that (4) is minimized. The problem can be solved by setting the partial derivatives of (4) with respect to $a[k]$ to zero. This leads to so called *Yule-Walker equations* that can be efficiently solved using so-called *Levinson-Durbin recursion* [8].

The Levinson-Durbin recursion generates as its side product so-called *reflection coefficients*, denoted here as $k[i], i = 1, \ldots, p$. The name comes from the multitube model, each reflection coefficient characterizing the transmission/reflection of the acoustic wave at each tube junction. Instead of using the reflection coefficients, we use instead the numerically more stable *arcus sine reflection coefficients* [3].

In the frequency domain, the linear predictive coefficients specify an IIR filter with the transfer function:

$$H(z) = \frac{1}{1 - \sum_{k=1}^{p} a[k] z^{-k}}. \qquad (5)$$

The *poles* of the filter (5) are the zeroes of the denominator. They are denoted here as $z_1, z_2, \ldots, z_p$, and they can be found by numerical root-finding techniques. The coefficients $a[k]$ are real, which restricts the poles to be either real or occur in complex conjugate pairs.

If the poles are well separated in the complex plane, they can be used for estimating the formant frequencies [5]:

$$\hat{F}_i = \frac{F_s}{2\pi} \tan^{-1}\left(\frac{\mathrm{Im}\ z_i}{\mathrm{Re}\ z_i}\right). \qquad (6)$$

Table 1: Summary of the NIST-1999 subset

Language	English
Speakers	230
Speech type	Conversational
Quality	Telephone
Sampling rate	8.0 kHz
Quantization	8-bit μ-law
Training speech (avg.)	119.0 sec.
Evaluation speech (avg.)	30.4 sec.

Given the LPC coefficients $a[k], k = 1, \ldots, p$, the LPCC coefficients are computed using the recursion [1]:

$$c[n] = \begin{cases} a[n] + \sum_{k=1}^{n-1} \frac{k}{n} c[k]a[n-k], & 1 \leq n \leq p \\ \sum_{k=n-p}^{n-1} \frac{k}{n} c[k]a[n-k], & n > p. \end{cases} \quad (7)$$

2.3 Delta Features

There are two different ways for computing the delta features: (1) differentiating, and (2) fitting a polynomial expansion. We have found out that the differentiator method works systematically better than the first order polynomial, i.e. the linear regression method [10]. Let $f_k[i]$ denote the ith feature in the kth time frame. The differentiator method estimates the time derivative of the feature as follows [5]:

$$\Delta f_k[i] = f_{k+M}[i] - f_{k-M}[i], \quad (8)$$

where M is typically 1-3 frames.

3 Experiments

3.1 Speech Material and Parameter Setup

For the experiments, we used a subset of the *NIST 1999 speaker recognition evaluation corpus* [18] (see Table 1). We decided to use the data from the male speakers only. For training, we used both the "a" and "b" sessions. For identification, we used the one speaker test segments from the same telephone line. In general it can be assumed that if two calls are from different lines, the handsets are different, and if they are from the same line, the handsets are the same [18]. In other words, the training and matching conditions have very likely the same handset type (electret/carbon button) for each speaker, but different speakers can have different handsets. The total number of test segments for this condition is 692.

The parameters for different feature sets and training algorithm were based on our previous experiments with the NIST corpus [12]. The frame length and shift were set to 30 ms and 20 ms, respectively, and the window function was Hamming. For MFCC computation, the number of filters was set to 27, and the number of coefficients was 12. For LPCC, ARCSIN and FMT, we used LPC predictor of order $p = 20$. We selected 12 LPCC and ARCSIN coefficients, and 8 formant frequencies. The delta features were computed using the differentiator method with $M = 1$. Throughout the experiments, codebook size was fixed to 64, and the codebooks were trained using the Linde-Buzo-Gray (LBG) clustering algorithm [15].

3.2 Individual Feature Sets

The identification error rates of the individual feature sets are reported in Table 2. The static features (MFCC, LPCC, ARCSIN, FMT) all give good results. The delta features, on the other hand, are worse than the static features. The error rate of delta formants is very high.

Table 2: Accuracies of the individual feature sets

Static features		*Dynamic features*	
Feature set	Error rate (%)	Feature set	Error rate (%)
MFCC	16.8	ΔMFCC	21.2
LPCC	16.0	ΔLPCC	25.1
ARCSIN	17.1	ΔARCSIN	28.6
FMT	19.4	ΔFMT	70.5

3.3 Fusion Results

Next, we experimented by fusing the static parameters and their corresponding delta features using all the three strategies. We also combined all the 8 feature sets. For the feature-level fusion, each feature vector was normalized by its norm, and the normalized vectors were then concatenated. For the score-level fusion, we used the normalized VQ distortions giving unity weights to all feature sets [11]. For the decision-level fusion, we use majority voting, by selecting speaker label that is voted most by all classifiers. If no speaker received majority, then speaker label is selected randomly from the highest number of votes.

The fusion results are shown in Table 3, along with the best individual performance from the pool. The score-level fusion gives the best result in all cases fusing feature with it's delta parameters, except with the formant data for which fusion is not succesfull. The reason for poor performance in this case is the poor performance of delta formants. Situation could be alleviated by de-emphasizing the delta formants.

It can be seen that the feature-level fusion improves the performance over the individual classifier in the case of MFCC and its delta features. However, in all other cases it degrades the performance. The decisionl-level fusion is the best fusion strategy, when all feature sets are used. Majority voting is not applicable for only two classifier system as seen for all other cases, where performance is degraded.

In the case, when user has only feature set and its delta parameters, results show that the score-level fusion seems to be the method to be preferred in the case of reliable experts. However, if some of the "experts" produces a lot of classification errors (ΔFMT), the weight for the unreliable features or feature sets should be set small. In this study, we did not attempt to weight individual features or

Table 3: Accuracies of the fused systems.

Combination	Best individual	Feature-level	Score-level	Decision-level	Oracle
MFCC + ΔMFCC	16.8	15.8	**14.6**	19.0	12.3
LPCC + ΔLPCC	16.0	19.8	**14.7**	20.5	12.6
ARCSIN + ΔARCSIN	17.1	18.2	**16.8**	22.8	15.0
FMT + ΔFMT	**19.4**	29.9	52.0	44.9	18.5
All feature sets	16.0	21.2	15.2	**12.6**	7.8

Table 4: Q statistic between all classifier pairs.

	MFCC	ΔMFCC	LPCC	ΔLPCC	ARCSIN	ΔARCSIN	FMT	ΔFMT
MFCC		0.916	0.976	0.861	0.953	0.875	0.925	0.594
ΔMFCC			0.909	0.934	0.869	0.847	0.838	0.527
LPCC				0.907	0.984	0.929	0.952	0.637
ΔLPCC					0.866	0.898	0.854	0.517
ARCSIN						0.948	0.956	0.753
ΔARCSIN							0.921	0.505
FMT								0.842

feature sets. In the case of feature-level fusion, it is not obvious how the individual features should be weighted.

3.4 Feature Set Diversity

Although the fusion improves performance in most cases, the gain is rather low. Intuitively, if the different classifiers misclassify the same speech segments, we do not expect as much improvement as in the case where they complement each other. There are several indices to assess the interrelationships between the classifiers in a classifier ensemble [4].

Given classifiers i and j, we compute the *Q statistic* [4]:

$$Q_{i,j} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}, \qquad (9)$$

where N^{00} is the number of test segments misclassified by both i and j; N^{11} is the number of segments correctly classified by both; N^{10} and N^{01} are the numbers of segments misclassified by one and correctly classified by the other. It can be easily verified that $-1 \leq Q_{i,j} \leq 1$. The Q value can be considered as a correlation measure between the classifier decisions.

The Q statistics between all feature set pairs are shown in Table 4. It can be seen that all values are positive and relatively high, which indicates that the classifiers function essentially the same way. In other words, the classifiers are *competitive* instead of *complementary* [13]. This partially explains why the performance is not greatly improved by fusion. Interestingly, although the performance of delta formants is very poor, it has lowest Q values on average. This means that delta formants make different decisions compared to other feature sets.

Table 5: Distribution of the number of correct votes.

8	7	6	5	4	3	2	1	0
155	269	72	39	43	23	22	15	54

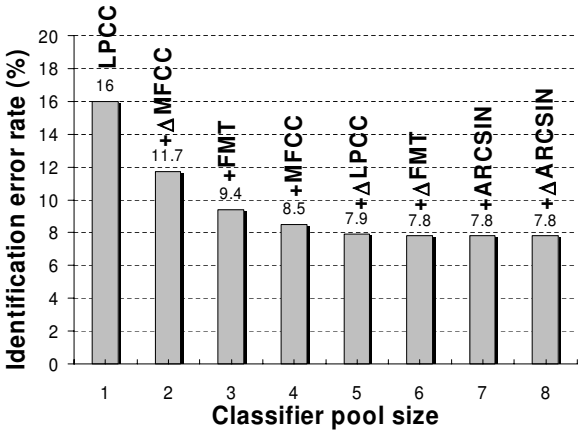

Figure 1: Performance of the "Oracle" classifier.

We can also analyze the difficulty of the test segments. Table 5 shows how many classifiers voted correctly on the same test segments out of 692. Interestingly, most test segments are voted correctly by 6,7 or 8 classifiers (72 %), which means that most of the test segments are relatively "easy". However, in the other end, there were 54 test segments (8 %) that no classifier voted correctly. This shows that some speakers are more difficult to recognize.

3.5 "Oracle" Classifier

We can estimate the lower limit of the identification error rate using an abstract machine called *Oracle classifier* [14]. The Oracle assigns correct class label to the test segment if at least one feature set classifies it correctly. Figure 1 shows the performance of this abstract classifier as a function of the classifier pool size. New classifiers are added to the pool in a greedy manner, starting from the best individual feature set (LPCC) and adding the fea-

ture set that decreses the error rate most. The lowest error rate (7.8 %) is reached by using six feature sets. The test segments classified correctly by the ARCSIN and ΔARCSIN feature sets are already classified correctly by some of the other feature sets. It must be emphasized that this is only a theoretical classifier, giving an idea of the lowest possible error rate if the diversity of the feature sets was taken fully into account.

4 Conclusions

We have compared and analyzed different ways of using several spectral feature sets for speaker identification. From the individual feature sets considered, linear predictive cepstral coefficients performed the best giving an error rate of 16.0 %. The best fusion result reduced this to 12.6 %, and it was obtained by decision-level fusion with all feature sets. If many different feature sets are availeble we recommend to use majority voting, otherwise in more traditional setting score-level fusion is the best.

Although fusion improves performance, the difference is not big. The analysis of the classifier diversities showed that the different feature sets classify speakers essentially in the same way. It is possible to reduce the error rate further by setting feature set depended weights reflecting the relative importances of the feature set. In future, we plan to use speaker-dependent weights and recent advances in information fusion, e.g. *decision templates* and *consensus classification* [13].

5 Acknowledgements

The work of V. Hautamäki was supported by the National Technology Agency of Finland (project "Puheteknikan uudet menetelmät ja sovellukset", TEKES dnro 8713103).

References

[1] B. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *Journal of the Acoustic Society of America*, 55(6):1304–1312, 1974.

[2] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1996.

[3] J. Campbell. Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997.

[4] C.A.Shipp and L.I.Kuncheva. Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion*, 3:135–148, 2002.

[5] J.R. Jr. Deller, J.H.L. Hansen, and J.G. Proakis. *Discrete-Time Processing of Speech Signals*. IEEE Press, New York, second edition, 2000.

[6] P. Eskelinen-Rönkä. Report on the testing of *Puhujan Tunnistaja* database software. MSc Thesis, Department of General Phonetics, University of Helsinki, Helsinki, Finland, 1997. (in finnish).

[7] G. Fant. *Acoustic Theory of Speech Production*. The Hague, Mouton, 1960.

[8] J. Harrington and S. Cassidy. *Techniques in Speech Acoustics*. Kluwer Academic Publishers, Dordrecht, 1999.

[9] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: a Guide to Theory, Algorithm, and System Development*. Prentice-Hall, New Jersey, 2001.

[10] T. Kinnunen. *Spectral Features for Automatic Text-Independent Speaker Recognition*. Licentiate's thesis, University of Joensuu, Department of Computer Science, Joensuu, Finland, 2004.

[11] T. Kinnunen, V. Hautamäki, and P. Fränti. On the fusion of dissimilarity- based classifiers for speaker identification. In *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, pages 2641–2644, Geneva, Switzerland, 2003.

[12] T. Kinnunen, E. Karpov, and P. Fränti. Real-time speaker identification. In *Proc. Int. Conf. on Spoken Language 2004 (ICSLP 2004)*, Jeju Island, Korea, 2004. (to appear).

[13] L.I.Kuncheva. *Fuzzy Classifier Design*. Physica Verlag, Heidelberg, 2000.

[14] L.I.Kuncheva, C.J.Whitaker, C.A.Shipp, and R.P.W.Duin. Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis and Applications*, 6:22–31, 2003.

[15] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, 1980.

[16] Linguistic data consortium. WWW page, December 2004. `http://www.ldc.upenn.edu/`.

[17] J. Makhoul. Linear prediction: a tutorial review. *Proceedings of the IEEE*, 64(4):561–580, 1975.

[18] A. Martin and M. Przybocki. The NIST 1999 speaker recognition evaluation - an overview. *Digital Signal Processing*, 10(1-18):1–18, 2000.

[19] D.A. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang. The SuperSID project: exploiting high-level information for high-accuracy speaker recognition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, pages 784–787, Hong Kong, 2003.