ROBUST SPEAKER VERIFICATION OVER THE TELEPHONE BY FEATURE RECUPERATION

X. Li[†], *M. W. Mak*[†] *and S. Y. Kung*[‡]

[†]Center for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University
[‡]Department of Electrical Engineering, Princeton University, USA.

ABSTRACT

The performance of speaker verification systems is often compromised under real-world environments. For example, variations in handset characteristics could cause severe performance degradation. This paper presents a novel method to overcome this problem by using a non-linear handset mapper. Under this method, a mapper is constructed by training an elliptical basis function network using distorted speech features as inputs and the corresponding clean features as the desired outputs. During feature recuperation, clean features are recovered by feeding the distorted features to the feature mapper. The recovered features are then presented to a speaker model as if they were derived from clean speech. Experimental evaluations based on 258 speakers of the TIMIT and NTIMIT corpuses suggest that the feature mappers improve the verification performance remarkably.

1. INTRODUCTION

While today's speaker verification systems perform reasonably well under controlled conditions, their performances are often compromised under real-world environments. In particular, variations in handset characteristics are known to be the major cause of performance degradation. Although this problem has been addressed by a number of approaches, such as cepstral weighting [1], adaptive component weighting [2], cepstral mean subtraction [3], relative spectral processing [4], and signal bias removal [5], most of them operate on the assumption that the channel effect can be approximated by a linear filter, which may be a poor approximation. Therefore, a more complex representation of handset characteristics is required. To this end, this paper investigates the non-linear characteristics of telephone handsets and proposes a handset mapper that overcomes the limitations of the conventional approaches.

2. PROBLEMS OF CMS

Cepstral mean normalization (CMN) [3], a popular approach to channel mismatch compensation, is based on two assumptions: (1) the cepstral mean of clean speech is zero and (2) the channel is linear. These assumptions lead to a simple formulation of recovering the clean cepstrum from the channel distorted cepstrum:

$$c_{clean} = c_{distorted} - c_{chan} = c_{distorted} - E\{c_{distorted}\} \quad (1)$$

This project was supported by The Hong Kong Polytechnic University Grant No. 1.42.37.A410.

where $c_{chan}(= E\{c_{distorted}\})$ represents the channel cepstrum and $E\{\cdot\}$ denotes expectation. However, both assumptions are invalid in our applications (e.g. see our previous study [6]). To avoid reliance on the first assumption, we proposed the differential-partial cepstral mean subtraction (DPCMS) in [6]. In DPCMS, the channel cepstrum is given by

$$c_{chan} = E\{c_{distorted}\} - E\{c_{clean}\} \quad (2)$$

where the mean of the clean cepstrum $E\{c_{clean}\}$ is the cepstral mean of the whole TIMIT corpus. This method, however, does not address the problem of channel non-linearity. In this paper, we propose a novel compensation method that does not rely on any of the above assumptions.

3. NON-LINEAR FEATURE MAPPING

We hypothesize that clean cepstra and distorted cepstra are nonlinearly related, i.e.,

$$c_{clean} = f(c_{distorted}, c_{channel}) \quad (3)$$

where $f()$ is a nonlinear function. In our study, we used our recently proposed elliptical basis function (EBF) networks [7] to implement $f()$. An n-input, m-output EBF network can be considered as realizing a multi-dimensional non-linear mapper that maps data from $\mathcal{R}^n$ to $\mathcal{R}^m$. More specifically, the k-th output ($k = 1, \ldots, K$) of an EBF network transforms an input vector $\vec{x}_p$ into

$$w_{k0} + \sum_{j=1}^{J} w_{kj} \exp\left\{ -\frac{1}{2\gamma_j} (\vec{x}_p - \vec{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\vec{x}_p - \vec{\mu}_j) \right\} = y_k(\vec{x}_p) \approx f(x) \quad p = 1, \ldots, N \quad k = 1, \ldots, K \quad (4)$$

where $\vec{\mu}_j$ and $\boldsymbol{\Sigma}_j$ are the mean vector and covariance matrix of the jth basis function respectively, w_{k0} is a bias term, w_{kj} is an output weight connecting the hidden node j to the output node k, and γ_j is a smoothing parameter that controls the spread of the jth basis function. In our study, γ_j was determined heuristically by

$$\gamma_j = \alpha \sum_{l=1}^{L} \|\vec{\mu}_l - \vec{\mu}_j\| \quad j = 1, \ldots, J \quad (5)$$

where $\vec{\mu}_l$ denotes the l-th nearest neighbor of $\vec{\mu}_j$ in the Euclidean sense, L is the number of nearest neighbors, and α is a parameter that controls the spread of the basis functions.

The mean vectors and the covariance matrices of an EBF-based feature mapper can be estimated in three steps

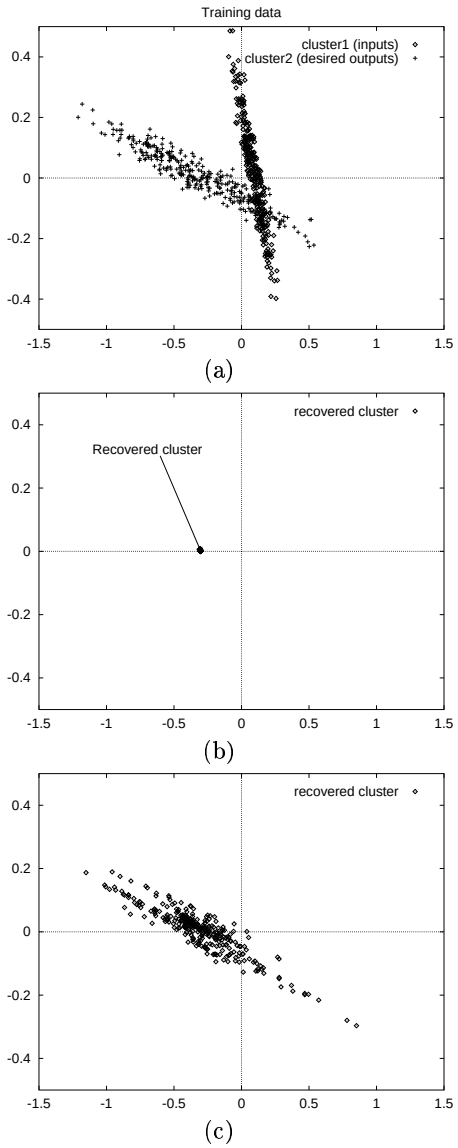

Figure 1: (a) Training data. (b) Data recovered by an EBF network with 1 center. (c) Data recovered by an EBF network with 10 centers.

[7]. In the first step, the K-means algorithm is applied to determine the cluster means of the training data $\mathcal{X}$ in the input feature space. Mathematically, we estimate the function center $\vec{\mu}_j$ by the sample average $\hat{\vec{\mu}}_j$, i.e.,

$$\vec{\mu}_j \approx \hat{\vec{\mu}}_j = \frac{1}{N_j} \sum_{\vec{x} \in \mathcal{X}_j} \vec{x} \tag{6}$$

where $\vec{x} \in \mathcal{X}_j$ if $\|\vec{x} - \hat{\vec{\mu}}_j\| < \|\vec{x} - \hat{\vec{\mu}}_k\| \; \forall j \neq k$, N_j is the number of samples in the cluster $\mathcal{X}_j$, and $\|\cdot\|$ is the Euclidean norm. In the second step, the covariance matrices are approximated by the sample covariance

$$\mathbf{\Sigma}_j \approx \hat{\mathbf{\Sigma}}_j = \frac{1}{N_j} \sum_{\vec{x} \in \mathcal{X}_j} \left(\vec{x} - \hat{\vec{\mu}}_j\right) \left(\vec{x} - \hat{\vec{\mu}}_j\right)^T. \tag{7}$$

Finally, the output weights $\{w_{kj}\}$ can be determined by a least squared method.

4. RECUPERATION OF GAUSSIAN CLUSTERS

We used two Gaussian clusters (with each cluster containing 1000 samples) to demonstrate the idea of feature recuperation. The first cluster had a mean vector at $(0.1, 0.0)$ with covariance matrix $[(0.1, -0.1, -0.1, 0.3)]$, and the second cluster had a mean vector at $(-0.3, 0.0)$ with covariance matrix $[(0.5, -0.1, -0.1, 0.1)]$. Fig. 1(a) shows the two clusters. We trained a mapper that mapped the data from the first cluster onto the second cluster. The values of α and L in (5) were set to 60 and 5 respectively. After training, the mapper was applied to recover the second cluster with the first cluster as input. Fig. 1(b) shows the recovered data (EBF network's outputs) when the network contained one center ($J = 1$) only. Obviously, if the number of centers is too small, the mapper will map all input data onto a very small region in the feature space. This is because the least squares method that determines the output weights will find a least squares solution to satisfy all training data. In this case, the best compromise is to map all inputs onto a single point (which is the center of all desired outputs) in the feature space. Fig. 1(c) demonstrates that the mapper can recover Cluster 2 almost perfectly when there are sufficient function centers.

5. FEATURE RECUPERATION FOR SPEAKER VERIFICATION

Speaker verification is the verification of whether the voice of a claimant matches the voice of the claimed identity. While speaker verification based on clean speech has reached a high level of performance, severe performance degradation still occurs when telephone speech is used. In our study, we propose to apply speaker-specific feature mappers as a means to alleviate this problem.

5.1. Enrollment Procedure

We used 258 speakers (186 males and 72 females) from dialect regions 1, 2, 3, and 4 of the TIMIT and NTIMIT corpuses to evaluate the mappers. TIMIT is a phonetically balanced, continuous speech corpus, and NTIMIT was obtained by playing the speech in the TIMIT corpus through a telephone network, which resulted in a telephone bandwidth corpus [8].

In our study, the speakers were divided into four sets: a speaker set (76 speakers from dialect region 2), an anti-speaker set (38 speakers from dialect region 1), a pseudo-impostor set (68 speakers from dialect region 4), and a impostor set (76 speakers from dialect region 3). The SA and SX sentence sets in the corpora were used as the training set, and the SI sentence set was the test set. This arrangement allowed us to perform text-independent speaker verification experiments. See [7] for the details of regarding the use of these sets.

The feature vectors that characterize the voices of speakers were derived from an LPC analysis procedure. For each sentence, the silent regions were removed by using the information provided by the *.phn* files of the corpus. The remaining signals were pre-emphasized by a filter with transfer function $1 - 0.95z^{-1}$. For every 14 ms, 12th order LP-derived cepstral coefficients were computed by use of a 28 ms Hamming window.

The enrollment procedure that produces the speaker models is adopted from [7]. Each speaker in the speaker

set was assigned a personalized EBF network that modeled the characteristics of his/her own voice. For each network, the feature vectors derived from the SA and SX sentence sets of TIMIT were used for training. Each network was trained to recognize the data derived from two classes—the speaker class and the anti-speaker class. The former was derived from the speaker set while the latter from the anti-speaker set. Therefore each network (speaker model) consisted of 12 inputs, a number of hidden nodes (denoted as function centers), and 2 outputs, with each output representing one class.

Step 1: Apply the K-means algorithm to the cepstral vectors of the speaker being enrolled. The resulting centers are referred to as the speaker centers.

Step 2: Apply the K-means algorithm to the cepstral vectors of all anti-speakers in the anti-speaker set to obtain a pool of function centers. These centers are referred to as the anti-centers.

Step 3: Apply (7) to obtain the function widths corresponding to the speaker centers by using $\vec{x}$ as the cepstral vectors of the speaker and $\vec{\mu}_j$ as the speaker centers obtained from Step 1. Similarly, (7) is applied to the cepstral vectors of the anti-speakers to obtain the widths corresponding to the anti-centers.

Step 4: Compute γ_j in (4) according to (5) and compute the matrix $\mathbf{\Phi}$. Apply singular value decomposition to find the output weights $\mathbf{W}$.

Step 5: Determine the decision threshold as in [9].

Note that this procedure embeds the anti-speaker model in the speaker model, which enables the network to perform scoring normalization during verification [10].

5.2. Constructing Feature Mappers

Besides the speaker model, each speaker was also assigned a personalized feature mapper. These mappers, which consisted of 12 inputs, 12 outputs and a large number of function centers (20 in our study), were trained to recover the clean TIMIT cepstra from the distorted NTIMIT cepstra. The feature vectors (cepstra) derived from the SA and SX sentence sets were used for the training. The procedure of constructing a personalized feature mapper is identical to that of constructing a speaker model, except for the following differences. First, the K-means algorithm and (7) were applied to the distorted cepstra derived from the corresponding speaker and anti-speakers for the case of constructing a feature mapper, whereas in the case of constructing a speaker model, they were applied to the speakers' cepstra and anti-speakers' cepstra independently.[1] Second, the feature mappers were trained to produce clean cepstra, while the speaker models were trained to discriminate the true speaker from the impostors. Therefore, the feature mappers perform function interpolation while the speaker models perform pattern classification.

We built and trained a series of speaker-specific mappers that mapped the NTIMIT cepstra onto their TIMIT counterparts. Although perfect recuperation is impossible, recognition performance might be improved if we could bring the distorted features closer to the clean ones by using the mappers. To this end, we used distorted speech in the NTIMIT corpus as inputs and the clean TIMIT speech as the desired outputs for training. We used 20 centers (10 for speaker's centers and 10 for antispeakers' centers) to construct a speaker-dependent mapper. The values of α and L in (5) were set to 60 and 5 respectively for all mappers.

[1] Since we do not know prior to verification whether a claimant is the true speaker or an impostor, the mappers need to handle both types of speakers.

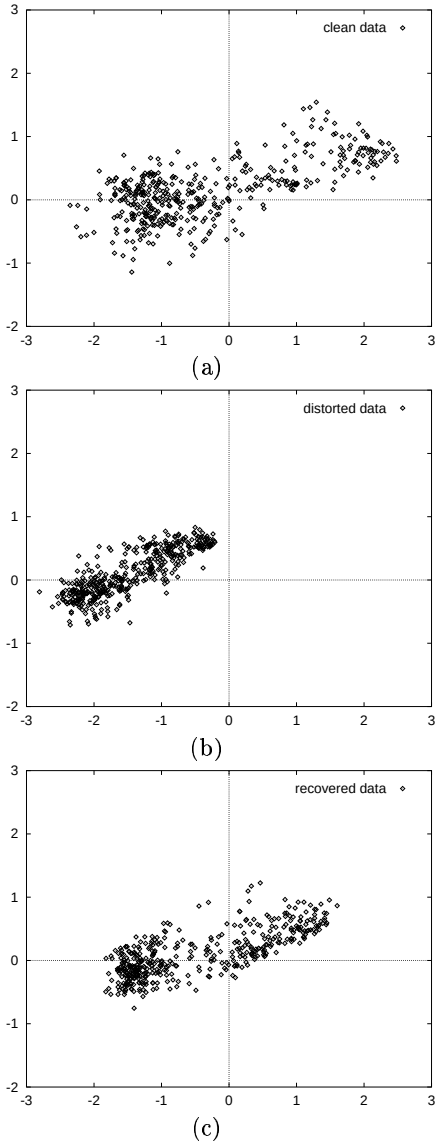

Figure 2: Clean, distorted and recovered data of a speaker. (a) The clean data. (b) The distorted data used to train the mapper. (c) The recovered data.

5.3. Verification and Threshold Determination

Verification sessions were divided into genuine attempts and impostor attempts. For each genuine attempt, utterances were extracted from the SI sentence set of the true speaker; whereas for each impostor attempt, utterances were extracted from the SI sentence set of the impostor set. For each utterance made by the claimant (could be a true speaker or an impostor), the difference between the two scaled average outputs of the EBF network was computed.

More specifically, we computed

$$z_k = \frac{1}{T} \sum_{\vec{x} \in \mathcal{T}} \frac{\exp\{\tilde{y}_k(\vec{x})\}}{\exp\{\tilde{y}_1(\vec{x})\} + \exp\{\tilde{y}_2(\vec{x})\}} \quad k = 1, 2 \qquad (8)$$

where $\mathcal{T} = [\vec{x}_1, \ldots, \vec{x}_T]$ is a vector sequence extracted from the utterance and $\tilde{y}_k(\vec{x})$ is the scaled output, i.e.,

$$\tilde{y}_k(\vec{x}) = \frac{1}{2} \cdot \frac{y_k(\vec{x})}{P(C_k)} \quad k = 1, 2 \qquad (9)$$

where $P(C_k)$ is the prior probability. As a result, we have $\frac{1}{N} \sum_{x \in \mathcal{X}} \tilde{y}_k(\vec{x}) \approx 0.5$, where N denotes the number of patterns in the training set $\mathcal{X}$. A simple way to estimate the prior probability $P(C_k)$ is to divide the number of patterns in class C_k by the total number of patterns in the training set.

Verification decisions were based on the criterion:

$$\text{If } z = z_1 - z_2 \begin{cases} > \zeta & : \text{ accept the claimant} \\ \leq \zeta & : \text{ reject the claimant} \end{cases} \qquad (10)$$

where ζ is the decision threshold corresponding to the claimed identity. Note that the decision threshold was determined during the enrollment phase [9]. More specifically, after a network has been trained, the verification procedure described above was applied. However, instead of using the speech of an unknown speaker, the feature vectors derived from the pseudo-impostor set were used. The threshold was adjusted between the range $[-1, +1]$ until the false acceptance rate (FAR) fell below a predefined value. In our study, the predefined FAR was set to 0.02%.

Once the threshold value was found, the false rejection rate (FRR) corresponding to each speaker was obtained by presenting the SI sentence set of the speaker to his/her own network. The false acceptance rate (FAR) was obtained by feeding the SI sentence set of all impostors (from the impostor set) to the network.

5.4. Results and Discussion

To evaluate the performance of a mapper, we used the distorted speech as inputs and compared the mapper's outputs with the desired clean cepstra. Fig. 2 plots the clean, distorted, and recovered cepstra of a speaker. Since the cepstra are 12-dimensional vectors, only the first 2 components (which have the largest variance) are plotted. Fig. 2 shows that the recovered data are closer to the clean data (in the $c1 - c2$ space) than the distorted data.

Table 1 shows the error rate based on the average of 76 true speakers and 76 impostors. The results show that a very low equal error rate (EER) can be achieved when clean patterns (TIMIT) are used for verification, and that the EER can be as high as 47.56% when NTIMIT speech is used. On the other hand, when mappers are used to recover the clean patterns, the EER is reduced to 0.30%, which is very close to the results obtained by using the clean patterns without the mappers. This suggests that the verification performance can be improved remarkably by using the feature mappers.

6. CONCLUSION

In our study, elliptical basis function (EBF) networks were applied to map the distorted speech vectors to clean speech vectors for speaker verification. The results show that using recovered data as inputs to speaker models leads to lower error rates. By conducting the Gaussian clusters recuperation experiments and the text-independent speaker verification experiments, we confirmed that a remarkable performance can be obtained by using the proposed mappers and recuperation procedures.

Table 1: Error rates based on the TIMIT, NTIMIT and recovered patterns. (FAR: false acceptance rate, FRR: false rejection rate, EER: equal error rate. All results are based on the average of 76 speakers.)

	FAR(%)	FRR(%)	EER(%)
TIMIT	0.95	2.51	0.10
NTIMIT	2.85	96.88	47.56
RECOVERED	18.12	0.00	0.30

7. REFERENCES

[1] Y. Tohkura. A weighted cepstral distance measure for speech recognition. *IEEE Transactions on Acoustics,Speech,and Signal Processing*, Assp-35(10):1414–1422, October 1987.

[2] K. T. Assaleh and R. J. Mammone. New LP-derived features for speaker identification. *IEEE Trans. on Speech and Audio Processing*, 2(4):630–638, 1994.

[3] B. S. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *J. Acoust. Soc. Am.*, 55(6):1304–1312, 1974.

[4] H. Hermansky and N. Morgan. Rasta processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, Oct 1994.

[5] M. G. Rahim and B. H. Juang. Signal bias removal by maximum likelihood estimation for robust telephone speech recognition. *IEEE Transactions on Speech and Audio Processing*, 4(1):19–30, Jan 1996.

[6] T.F. Lo, M.W. Mak, and K.K. Yiu. A new cepstrum-based channel compensation method for speaker verification. In *Eurospeech'99*, volume 2, pages 775–778, Sept. 1999.

[7] M. W. Mak and S. Y. Kung. Estimation of elliptical basis function parameters by the EM algorithm with application to speaker verification. *IEEE Trans. on Neural Networks*, 11(4):961–969, July 2000.

[8] C. Jankowski, A. Kalyanswamy, S. Basson, and J. Spitz. NTIMIT: A phonetically balanced, continuous speech, telephone bandwidth speech database. In *ICASSP'90*, pages 109–112, 1990.

[9] W. D. Zhang, M. W. Mak, C. K. Li, and M. X. He. A priori threshold determination for phrase-prompted speaker verification. In *Eurospeech'99*, volume 2, pages 1023–1026, 1999.

[10] W. D. Zhang, M. W. Mak, and M. X. He. A two-stage scoring method combining world and cohort model for speaker verification. In *Proc. ICASSP'2000*, June 2000.