PROBABILISTIC FEATURE TRANSFORMATION FOR CHANNEL ROBUST SPEAKER VERIFICATION

Man-Wai Mak and Kwok-Kwong Yiu

Center for Multimedia Signal Processing
Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong SAR, China

ABSTRACT

Feature transformation plays an important role in robust speaker verification over telephone networks. This paper compares several feature transformation techniques and evaluates their verification performance and computation time under the 2000 NIST speaker recognition evaluation protocol. Techniques compared include feature mapping (FM), stochastic feature transformation (SFT), and blind stochastic feature transformation (BSFT). The paper proposes a probabilistic feature mapping (PFM) in which the mapped features depend not only on the top-1 decoded Gaussian but also on the posterior probabilities of other Gaussians in the root model. The paper also proposes speeding up the computation of PFM and BSFT parameters by considering the top few Gaussians only. Results show that PFM performs slightly better than FM and that the fast approach can reduce computation time substantially. Among the approaches investigated, the fast BSFT is found to have the highest potential for robust speaker verification over telephone networks because it can achieve good performance without any a priori knowledge of the communication channel. It was also found that fusion of the scores derived from systems using BSFT and PFM can reduce the error rate further.

1. INTRODUCTION

Speaker verification is a biometric trait that is particularly suitable for identifying or authenticating users over the telecommunication and telephone networks, which has high potential for securing financial transactions and remote information access. Although commercial speaker verification systems are now available, the lack of robustness to channel variability and the acoustic mismatch between enrollment and verification conditions remain a major practical challenge. Currently, this problem is addressed by

a technique called channel mismatch compensation. The goal of channel compensation is to achieve performance approaching that of a “matched condition” system. Channel compensation can be applied in feature space [1, 2], model space [3, 4] or score space [5]. One advantage of feature-space compensation is that it is not necessary to modify the speaker models after training.

This paper focuses on feature-based compensation and compares several closely related feature compensation techniques under the 2000 NIST SRE framework [6]. Techniques compared include feature mapping (FM), stochastic feature transformation (SFT), and blind stochastic feature transformation (BSFT). These techniques are closely related in that they all attempt to transform distorted spectral features to fit the clean acoustic models. The paper proposes improving the performance of FM by introducing a probabilistic term in the mapping function so that the mapped features depend not only on the winner mixture but also on the posterior probabilities of other mixtures in the root model. The resulting mapping is referred to as probabilistic feature mapping (PFM). Because both BSFT and PFM require the posterior probabilities of all mixtures in the parameter estimation process, computation time can become a problem when the model size is large. The paper therefore proposes speeding up the computation of FM and BSFT’s parameters by considering the top few components only in the parameter estimation process.

2. FEATURE TRANSFORMATION FOR CHANNEL COMPENSATION

There are two main types of channel compensation: blind and non-blind. The former adapts speaker models or transforms speaker features to accommodate the channel variation without a priori knowledge of the channel characteristics. Non-blind compensation, on the other hand, estimates channel-specific compensation based on a priori knowledge of all possible channels; during recognition, the channel type is identified and is used to select the pre-computed channel compensation to reduce the acoustic mis-

This project was supported by the Hong Kong Polytechnic University the Research Grants Council of Hong Kong SAR (Project Nos. PolyU 5214/04E and PolyU 5230/05E). The authors thank Prof. S.Y. Kung of Princeton University for valuable discussions and suggestions.

match caused by mismatched channels.

2.1. Blind Compensation

Blind compensation can be categorized into four types. The first type exploits the temporal variability of feature vectors. For example, cepstral mean subtraction (CMS) [7] subtracts the cepstral mean of an utterance from each of the cepstral vectors. RASTA [8] applies a bandpass filter to the sequence of cepstral vectors to remove the slow varying components corresponding to the channel. It has been shown that both mean normalization and bandpass filtering can minimize the filtering effect of linear channels [7, 8]. However, these techniques may cause performance degradation when both training and recognition are derived from the same acoustic environment [2].

The second type of blind compensation transforms the distorted features such that acoustic environments have minimum effect on the distribution of the transformed features. For example, in feature warping [9], observed features are mapped to a target distribution (e.g., standard normal) such that they follow the target distribution over a sliding window of feature vectors. Specifically, given a sequence of feature vectors, a sliding window of 3 seconds is applied to the sequence to compute the cumulative distribution function (cdf) of each feature component. For each feature component, the original feature value at the middle of the sliding window is then mapped to a target value such that the cdf at the original feature value is equal to the target cdf at the target value. The warping can be viewed as a nonlinear feature transformation from the original feature to a warped feature. Feature warping has been shown to be robust to different channel and noise effects. In short-time Gaussianization [10], a linear transformation is applied to the distorted feature before mapping them to a normal distribution. The transformation aims to decorrelate the feature vectors, making them more amendable to diagonal covariance GMMs. It was found that short-time Gaussianization is superior to feature warping, especially at low false acceptance rate.

The third type makes use of the statistical difference between the clean acoustic models and the distorted speech to estimate a transformation matrix to map the distorted vectors to fit the clean model. This type of technique include the blind stochastic feature transformation [2] to be detailed in Section 3.

The fourth type, namely discriminative feature design [11], trains a neural network discriminatively to maximize speaker recognition performance on the training set. Because the training set consists of different types of acoustic distortions that the system may encounter during recognition, the neural network is able to “recall” the compensation required during recognition to reduce the effects of handset distortions on speaker discrimination.

2.2. Non-Blind Compensation

One typical property of non-blind techniques is the requirement of channel detection during recognition. Typical examples are feature mapping [12], spectral-magnitude matching [13], and stochastic feature transformation [14].

In feature mapping, the handset type of the testing utterance is identified by a handset detector; feature vectors are then mapped to the channel-independent space based on the closest Gaussian in the channel-dependent GMM. In spectral-magnitude matching [13], a nonlinear polynomial mapper is trained to minimize the mean-squared spectral magnitude error between speech arising from electret and carbon-button handsets. The mapper is shown to be good at minimizing mismatches caused by phantom formants, bandwidth widening, and spectral flattening due to channel nonlinearity. Stochastic feature transformation (SFT) [14] is derived from the stochastic matching method of Sankar and Lee [15], which was originally proposed for robust speech recognition. SFT aims to transform the distorted features to fit the clean speech models by selecting the most appropriate pre-computed transformation matrix. It has been shown that SFT can be extended to non-linear feature transformation to overcome the nonlinear distortion [14].

3. BLIND STOCHASTIC FEATURE TRANSFORMATION

The blind stochastic feature transformation (BSFT) proposed in [2] is a blind approach to channel mismatch compensation. Specifically, given a D -dimensional distorted vector $\mathbf{y}$, the transformed feature vector is

$$\mathbf{x} = f_{\nu}(\mathbf{y}) = A\mathbf{y} + \mathbf{b}, \quad (1)$$

where $A = \text{diag}\{a_1, \dots, a_D\}$ is a transformation matrix, $\mathbf{b} = [b_1, \dots, b_D]^T$ represents a bias vector, $\nu = \{a_i, b_i\}_{i=1}^D$ is the set of transformation parameters, and $f_{\nu}(\cdot)$ denotes the transformation function.

The BSFT parameters A and $\mathbf{b}$ are determined by maximizing the likelihood function of a composite GMM formed by the fusion of a compact speaker model and a compact background model given the distorted feature vectors. This is achieved by maximizing an auxiliary function

$$Q(\nu'|\nu) = \sum_{t=1}^T \sum_{j=1}^{M_c} h_j(f_{\nu'}(\mathbf{y}_t)) \log \left\{ \frac{p(f_{\nu'}(\mathbf{y}_t)|\boldsymbol{\mu}_j, \Sigma_j)}{|J_{\nu'}(\mathbf{y}_t)|} \right\} \quad (2)$$

with respect to ν' . In Eq. 2, ν' and ν represent the new and current estimates of the transformation parameters, respectively. $\Lambda = \{\pi_j, \boldsymbol{\mu}_j, \Sigma_j\}_{j=1}^{M_c}$ (typically $M_c = 128$) is a composite model derived from a compact target-model and a compact background-model (both with $M_c/2$ centers); $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_T\}$ is the distorted features ex-

tracted from a verification utterance; T is the number of distorted vectors; $f_{\nu'}(\cdot)$ denotes the transformation; $|J_{\nu'}(\mathbf{y}_t)|$ is the determinant of the Jacobian matrix, the (r, s) -th entry of which is given by $J_{\nu'}(\mathbf{y}_t)_{rs} = \partial f_{\nu'}(\mathbf{y}_t)_r / \partial y_{t,s}$; and $h_j(f_{\nu}(\mathbf{y}_t)) = \frac{\pi_j p(f_{\nu}(\mathbf{y}_t) | \boldsymbol{\mu}_j, \Sigma_j)}{\sum_{i=1}^{M_c} \pi_i p(f_{\nu}(\mathbf{y}_t) | \boldsymbol{\mu}_i, \Sigma_i)}$, is the posterior probability of the transformed feature $f_{\nu}(\mathbf{y}_t)$ being generated by the j -th mixture of Λ . Maximizing $Q(\nu' | \nu)$ with respect to ν' leads to the following close-form solution for $\{a'_i\}$ and $\{b'_i\}$ for $i = 1, \dots, D$ in the M-step of the EM algorithm:

$$\begin{aligned} b'_i &= (p_i - q_i a'_i) / r_i \\ (s_i - q_i^2 / r_i) a_i'^2 + (q_i p_i / r_i - u_i) a'_i - T &= 0 \end{aligned}$$

where

$$\begin{aligned} p_i &= \sum_{t=1}^T \sum_{j=1}^{M_c} h_j(f_{\nu}(\mathbf{y}_t)) \mu_{ji} \sigma_{ji}^{-2} \\ q_i &= \sum_{t=1}^T \sum_{j=1}^{M_c} h_j(f_{\nu}(\mathbf{y}_t)) y_{ti} \sigma_{ji}^{-2} \\ r_i &= \sum_{t=1}^T \sum_{j=1}^{M_c} h_j(f_{\nu}(\mathbf{y}_t)) \sigma_{ji}^{-2} \\ s_i &= \sum_{t=1}^T \sum_{j=1}^{M_c} h_j(f_{\nu}(\mathbf{y}_t)) y_{ti}^2 \sigma_{ji}^{-2} \\ u_i &= \sum_{t=1}^T \sum_{j=1}^{M_c} h_j(f_{\nu}(\mathbf{y}_t)) \mu_{ji} y_{ti} \sigma_{ji}^{-2}. \end{aligned} \quad (3)$$

To reduce the computational complexity of BSFT, we propose adopting a fast technique to compute the transformation parameters. In the original BSFT, all the posterior probabilities $\{h_j(f_{\nu}(\mathbf{y}_t))\}_{j=1}^{M_c}$ are involved in the maximization of Eq. 2. In the fast BSFT (fBSFT), the top C posterior probabilities in the composite model are determined and the transformation parameters $\nu = \{A, \mathbf{b}\}$ are computed using these top C Gaussians. This is equivalent to maximizing the auxiliary function

$$Q(\nu' | \nu) = \sum_{t=1}^T \sum_{j \in \mathcal{C}} h_j(f_{\nu}(\mathbf{y}_t)) \log \left\{ \frac{p(f_{\nu'}(\mathbf{y}_t) | \boldsymbol{\mu}_j, \Sigma_j)}{|J_{\nu'}(\mathbf{y}_t)|} \right\} \quad (4)$$

with respect to ν' , where $\mathcal{C}$ contains the indexes of the top- C Gaussians. In this work, C was set to 5.

4. PROBABILISTIC FEATURE MAPPING

Feature mapping [12] is a non-blind technique in that it relies on a handset detector that possess the information of all possible channels that the users may use during verification. In feature mapping, the transformation is based on the top-1 Gaussian only. Specifically, let GMM $\Lambda^{\text{CD}_i} = \{\pi_j^{\text{CD}_i}, \boldsymbol{\mu}_j^{\text{CD}_i}, \Sigma_j^{\text{CD}_i}\}_{j=1}^M$, be an M -mixture channel-dependent GMM for channel i and GMM $\Lambda = \{\pi_j^{\text{CI}}, \boldsymbol{\mu}_j^{\text{CI}}, \Sigma_j^{\text{CI}}\}_{j=1}^M$ be an M -mixture channel-independent root model. The mapping of a distorted vector $\mathbf{y}$ in the space

modelled by Λ^{CD_i} to the channel-independent vector $\mathbf{x}$ is given by

$$\mathbf{x} = \left(\mathbf{y} - \boldsymbol{\mu}_k^{\text{CD}_i} \right) \frac{\sigma_k^{\text{CI}}}{\sigma_k^{\text{CD}_i}} + \boldsymbol{\mu}_k^{\text{CI}}, \quad (5)$$

where $k = \arg \max_{j=1}^M \pi_j^{\text{CD}_i} p(\mathbf{y} | \boldsymbol{\mu}_j^{\text{CD}_i}, \Sigma_j^{\text{CD}_i})$.

To account for the effect of other Gaussian components on the transformed features, the transformation should be based on a weighted average of all Gaussian components, which leads to the probabilistic feature mapping (PFM). More specifically, we have

$$\mathbf{x} = \sum_{j=1}^M g_j(\mathbf{y}) \left[\left(\mathbf{y} - \boldsymbol{\mu}_j^{\text{CD}_i} \right) \frac{\sigma_j^{\text{CI}}}{\sigma_j^{\text{CD}_i}} + \boldsymbol{\mu}_j^{\text{CI}} \right], \quad (6)$$

where

$$g_j(\mathbf{y}) = P(j | \mathbf{y}, \Lambda^{\text{CD}_i}) = \frac{\pi_j^{\text{CD}_i} p(\mathbf{y} | \boldsymbol{\mu}_j^{\text{CD}_i}, \Sigma_j^{\text{CD}_i})}{\sum_{l=1}^M \pi_l^{\text{CD}_i} p(\mathbf{y} | \boldsymbol{\mu}_l^{\text{CD}_i}, \Sigma_l^{\text{CD}_i})}$$

is the posterior probability of the j component mixture. Note that the original feature mapping (Eq. 5) is a special case of the probabilistic feature mapping (Eq. 6).

The fast technique mentioned in Section 3 can also be applied to PFM. Specifically, Eq. 6 is changed to

$$\mathbf{x} = \sum_{j \in \mathcal{C}} g_j(\mathbf{y}) \left[\left(\mathbf{y} - \boldsymbol{\mu}_j^{\text{CD}_i} \right) \frac{\sigma_j^{\text{CI}}}{\sigma_j^{\text{CD}_i}} + \boldsymbol{\mu}_j^{\text{CI}} \right], \quad (7)$$

where $\mathcal{C}$ contains the indexes of the top- C Gaussians.

Let us use a 2-D hypothetical example as shown in Fig. 1 to illustrate the idea of PFM. In Fig. 1, the black dots represent patterns from a specific channel and the red circles and green triangles represent the centers of the root model and channel-dependent model, respectively. Fig. 1(a) shows that the patterns transformed by FM form a number of clusters (blue dots). Because only the top-1 Gaussian is used in the mapping function, distorted patterns near the boundary of two Gaussians can be transformed by two mapping functions with different characteristics (different means and variances). As a result, these patterns can be transformed to different regions of the feature spaces, causing the clustering effect in the transformed patterns. Fig. 1(b) shows that this clustering effect can be largely eliminated by using PFM. According to Eq. 6, the transformation of a pattern depends on all gaussian components; therefore no hard decision is made to decide which Gaussian the pattern should belong to. This has the effect of averaging out the effect caused by different mapping functions. The capability of PFM is also demonstrated in a speaker verification evaluation to be described next.

Fig. 1. A 2-D hypothetical example illustrating the clustering effect in feature mapping. In the figure, black dots (upper right region) represent features from a channel-dependent source and blue dots (central region) represent features transformed by (a) feature mapping and (b) probabilistic feature mapping. The red circles $\bullet$ and green triangles $\blacktriangle$ represent the centers of the root model and channel-dependent model, respectively.

5. EXPERIMENTS

5.1. Speech Corpus and Features

The feature transformation methods described in Sections 3 and 4 were applied to the one-speaker detection task specified in the 2000 NIST speaker recognition evaluation set [6]. The evaluation set contains landline telephone speech extracted from the SwitchBoard-II, Phase 1 and Phase 4 Corpus. The evaluation set includes 457 male and 546 female target speakers. For each speaker, approximately 2 minutes of speech is available for enrollment. There are 3026 female and 3026 male verification utterances. Each verification utterance has length not exceeding 60 seconds and is evaluated against 11 hypothesized speakers of the same sex as the speaker of the verification utterance.

Nineteen Mel-frequency cepstral coefficients (MFCCs) [16] and their first-order derivatives were computed every 10ms using a Hamming window of 25ms. Cepstral mean subtraction (CMS) was applied to the MFCCs to remove linear channel effects. The MFCCs and delta MFCCs were concatenated to form 38-dimensional feature vectors.

5.2. Creating Speaker and Background Models

For each gender, the corresponding gender-dependent evaluation utterances in NIST99 were used to train a 1024-component gender-dependent universal background models (UBMs) Λ_g , where $g \in \{\text{male, female}\}$.

The target-speaker models in BSFT and fBSFT are different from those in feature mapping. In the former, a target-speaker model $\Lambda_{g,k}$ was created for the k -th speaker

in NIST00 by adapting the corresponding gender-dependent UBM Λ_g using maximum a posteriori (MAP) adaptation [17]. Note that the adaptation process captures the speaker characteristics together with the channel characteristics of the enrollment session in the speaker models. For feature mapping, the speaker model $\Lambda_{g,k}$ is created by adapting the root model Λ_g using the transformed data $\mathbf{x}$ obtained from feature mapping. Fig. 2 shows the process of creating speaker models from the UBMs.

5.3. Creating Channel-Dependent Models

For feature mapping and probabilistic feature mapping, the gender-dependent UBMs Λ_g were used as the root GMMs, and gender- and channel-dependent evaluation utterances in NIST99 were used to adapt the corresponding gender-dependent UBMs to create the gender- and channel-dependent models $\Lambda_g^{\text{CD}_i}$, where $g \in \{\text{male, female}\}$ and $\text{CD}_i \in \{\text{cb, el}\}$. During verification, these models were then used for calculating the mapping function (Eqs. 5 or 6) that transforms the distorted features derived from the evaluation utterances in NIST00 to fit the gender-dependent root models Λ_g . Fig. 2 illustrates the feature mapping process.

For BSFT and fBSFT, a 128-mixture gender-dependent composite models was created for each speaker by combining his/her 64-mixture compact speaker models with a 64-mixture gender-dependent UBMs, i.e., $M_c = 128$ in Eq. 2. For SFT, the feature transformation parameters (A and b in Eq. 1) for transforming features from carbon-button (cb) handsets to electret (el) handsets or vice versa were determined off-line using the SFT estimation algorithm (Eq. 2 with $M_c = 64$) and channel-dependent data

from NIST99. During verification, if the test utterance was recorded from a handset with type identical to that of enrollment, no transformation was applied. Otherwise, the test patterns were transformed using the appropriate transformation ($f_{\nu_{cb \rightarrow el}}(\mathbf{y})$ or $f_{\nu_{el \rightarrow cb}}(\mathbf{y})$) to reduce the acoustic mismatch between the training and testing features.

Fig. 2. Procedures of creating target speaker models for FM, PFM, and BSFT.

5.4. Fusion of PFM and BSFT

Because BSFT and FM (or their fast variants) transform features differently, the scores obtained by speaker verification systems based on these transformation techniques may contain complementary information. To confirm this hypothesis, we trained a 2-input linear SVM using the training set of NIST00 to classify scores vectors $\mathbf{s} = [s_{\text{PFM}} \ s_{\text{BSFT}}]^T$ into speaker class and impostor class, where s_{PFM} and s_{BSFT} are scores obtained from a PFM-based and a BSFT-based system, respectively. The distances of $\mathbf{s}$ from the decision hyperplane of the SVM are then used for plotting DET curves.

6. RESULTS AND DISCUSSIONS

Fig. 3 and Table 1 show the performance of various transformation techniques and the fusion of PFM and BSFT. Evidently, all methods show significant reduction in error rates when compared to CMS. The DET curves also show that they outperform CMS at all operating points.

Table 1 also shows the computational complexity and transformation time of different transformation approaches. The measurements were performed on a Pentium IV 3.2 GHz processor using a verification utterance of 53 seconds. The results show that CMS is the fastest among all investigated methods because subtracting the mean from feature vectors is a very simple procedure. PFM is considerably

Fig. 3. DET curves comparing speaker verification performance of different feature transformation methods. See the caption of Table 1 for the full name of the acronyms.

slower than FM because for large M (1024 here), the time spent on computing Eq. 6 is considerably longer than that on computing Eq. 5. This is also reflected in the computational complexities of these two methods. The results also show that using only the top- C Gaussians can reduce the verification time of both BSFT and PFM.

The p-values [18] in Table 3 suggest that there is no significant difference between the EERs achieved by FM, PFM, fpFM, BSFT, fBSFT, and SFT. On the other hand, fusion of PFM and BSFT can reduce the EER significantly. Surprisingly, FM and BSFT (or their fast variants) achieve almost the same EER and minimum decision cost although the former uses the information about the channel types during training whereas the latter is an unsupervised technique that does not rely on any a priori channel information. Because BSFT and fBSFT do not require handset detection, they are more flexible than FM and PFM. Bear in mind that in real-world systems, users are likely to use handsets with widely different characteristics. In this situation, it is imperative to use a method that neither requires handset detection nor a priori information about the handset characteristics. Therefore, given its high performance in terms of EER, DET, decision cost, and moderate complexity, fBSFT appears to be the best channel compensation method (among those that have been investigated in this work) for practical implementation of speaker verification system.

The software for BSFT can be downloaded from <http://www.eie.polyu.edu.hk/~mwmak/mypubl.htm>.

Transformation Method		EER in %	p-value	Minimum Decision Cost	Computational Complexity of Feature Transformation	Transformation Time (sec.)
Blind	CMS	17.43	0.000	0.0611	$\mathcal{O}(PT)$	0.02
	BSFT	16.27	0.189	0.0564	$\mathcal{O}(PT + TDM_c + TDM_c + TD)$	78.01
	fBSFT	16.04	0.402	0.0557	$\mathcal{O}(PT + TDM_c + TCM_c + TDC + TD)$	18.84
Non- Blind	FM	16.05	NA	0.0577	$\mathcal{O}(PT + TDM + TDC + TM + TD)$	7.89
	PFM	15.91	0.567	0.0574	$\mathcal{O}(PT + TDM + TDC + TDM)$	66.19
	fPFM	16.47	0.009	0.0594	$\mathcal{O}(PT + TDM + TDC + TCM + TDC)$	7.90
	SFT	15.64	0.032	0.0589	$\mathcal{O}(PT + TDM + TDC + TD)$	7.67
	PFM+BSFT	15.29	0.001	0.0545	$\mathcal{O}(PT + 2TDM + TDC + 2TDM_c + TD + DS)$	144.18

P : No. of cepstral coefficients, excluding delta coefficients (= 19)

T : No. of feature vectors in the test utterance (= 5300)

D : Feature dimension (= 38)

M : No. of mixtures in speaker and background models (= 1024)

M_c : No. of mixtures in the composite models in BSFT and fBSFT (= 128)

C : No. of top mixtures used in handset detection, fBSFT, and fPFM (= 5)

S : No. of support vectors in the fusion SVM

Table 1. Equal error rates, p-value of EER with respect to FM, and minimum decision cost achieved by cepstral mean subtraction (CMS), blind stochastic feature transformation (BSFT), fast BSFT (fBSFT), feature mapping (FM), probabilistic feature mapping (PFM), fast PFM (fPFM), stochastic feature transformation (SFT), and fusion of PFM and BSFT. Note that the transformation time for non-blind methods (FM, PFM, fPFM, and SFT) includes the time for handset detection, transformation parameter estimation, and actual transformation. The feature extraction time (computing MFCCs) and scoring time for all approach is 0.72 and 7.38 seconds, respectively. All CPU times are based on the average of 20 verification attempts using a 53-second utterance (without silence).

7. CONCLUSIONS

This paper has compared several state-of-the-art feature transformation methods for robust speaker verification. The feature mapping method is also extended to probabilistic feature mapping. Fast algorithm for these methods are proposed and results show that computation saving can be achieved by considering the top few Gaussians only in the parameter estimation process. It was also found that although BSFT is more computationally demanding than FM, its fast version can reduce the computation time to a manageable level without jeopardizing verification accuracy.

8. REFERENCES

- [1] A. C. Surendran, C. H. Lee, and M. Rahim, "Nonlinear compensation for stochastic matching," *IEEE Trans. on Speech and Audio Processing*, vol. 7, no. 6, pp. 643–655, 1999.
- [2] K.K. Yiu, M.W. Mak, M.C. Cheung, and S.Y. Kung, "Blind stochastic feature transformation for channel robust speaker verification," *J. of VLSI Signal Processing*, vol. 42, no. 2, pp. 117–126, 2006.
- [3] F. Beaufays and M. Weintraub, "Model transformation for robust speaker recognition from telephone data," in *Proc. ICASSP'97*, April 1997, pp. 21–24.
- [4] K. K. Yiu, M. W. Mak, and S. Y. Kung, "Environment adaptation for robust speaker verification by cascading maximum likelihood linear regression and reinforced learning," *Computer Speech and Language*, in press.
- [5] D. A. Reynolds, "Comparison of background normalization methods for text-independent speaker verification," in *Proc. Eurospeech'97*, 1997, pp. 963–966.
- [6] "The NIST year 2000 speaker recognition evaluation plan," in <http://www.nist.gov/speech/tests/spk/2000/doc>.
- [7] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Am.*, vol. 55, no. 6, pp. 1304–1312, 1974.
- [8] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, Oct 1994.
- [9] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. Speaker Odyssey*, 2001, pp. 213–218.
- [10] B. Xiang, U. Chaudhari, J. Navratil, G. Ramaswamy, and R. Gopinath, "Short-time Gaussianization for robust speaker verification," in *Proc. ICASSP'02*, 2002, vol. 1, pp. 681–684.
- [11] L. P. Heck, Y. Konig, M. K. Sonmez, and M. Weintraub, "Robustness to telephone handset distortion in speaker recognition by discriminative feature design," in *Speech Communication*, 2000, vol. 31, pp. 181–192.
- [12] D. A. Reynolds, "Channel robust speaker verification via feature mapping," in *IEEE ICASSP*, 2003, vol. 2, pp. 6–10.
- [13] T. F. Quatieri, D. A. Reynolds, and G. C. O'Leary, "Estimation of handset nonlinearity with application to speaker recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 5, pp. 567–584, 2000.
- [14] S. Y. Kung, M. W. Mak, and S. H. Lin, *Biometric Authentication: A Machine Learning Approach*, Prentice Hall, Upper Saddle River, New Jersey, 2005.
- [15] A. Sankar and C. H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 3, pp. 190–202, 1996.
- [16] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on ASSP*, vol. 28, no. 4, pp. 357–366, August 1980.
- [17] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.
- [18] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. ICASSP'89*, 1989, pp. 532–535.