Probabilistic Feature-Based Transformation for Speaker Verification over Telephone Networks

Man-Wai Mak [a], Kwok-Kwong Yiu [a] and Sun-Yuan Kung [b]

[a] *Center for Multimedia Signal Processing*
Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong SAR

[b] *Dept. of Electrical Engineering*
Princeton University, USA

Abstract

Feature transformation aims to reduce the effects of channel- and handset-distortion in telephone-based speaker verification. 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), blind stochastic feature transformation (BSFT), feature warping (FW), and short-time Gaussianization (STG). 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 (fBSFT) strikes a good balance between computational complexity and error rates, and FW and STG are the best in terms of error rates but with higher computational complexity. It was also found that fusion of the scores derived from systems using fBSFT and STG can reduce the error rate further. This study advocates that fBSFT, FW, and STG have the highest potential for robust speaker verification over telephone networks because they achieve good performance without any *a priori* knowledge of the communication channel.

Key words: Speaker verification, feature transformation, channel compensation, biometrics, EM algorithm
PACS:

1 Introduction

Speaker verification is a biometric technology that aims to authenticate users via their voice patterns. Among the biometric traits that are currently under intensive investigation, speaker verification is apparently the best candidate for identifying or authenticating users over the telephone networks. Although commercial speaker verification systems that aim at securing financial transactions and remote information access 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) [7], stochastic feature transformation (SFT) [8], blind stochastic feature transformation (BSFT) [2], feature warping (FW) [9], and short-time Gaussianization (STG) [10]. The first three techniques attempt to transform distorted spectral features to fit the clean acoustic models. The last two, FW and STG, attempt to make the features less channel-dependent by normalizing the feature distribution. 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 excessively long for large model size. 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.

The paper is organized as follows. Section 2 discusses two main types of channel compensation: blind and non-blind. Section 3 explains the BSFT and its fast version, which is followed by Section 4 where FM and fast FM are outlined. These methods are then compared in Sections 5 and 6 under the NIST00 evaluation protocol.

Email address: enmwmak@polyu.edu.hk (Man-Wai Mak).
URL: http://www.eie.polyu.edu.hk/∼mwmak/mypage.htm (Man-Wai Mak).

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 during recognition 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. Specifically, during recognition, the channel type is identified and is used to select the pre-computed channel compensation to reduce the acoustic mismatch 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) [11] subtracts the cepstral mean of an utterance from each of the cepstral vectors. RASTA [12] 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 [11,12]. However, these techniques may cause performance degradation when both training and recognition are derived from the same acoustic environment [4].

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 channel variations and background noise. 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 Gaussian mixture models (GMMs). Short-time Gaussianization has shown advantages over 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 [13], 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 [7], spectral-magnitude matching [14], and stochastic feature transformation [8].

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 [14], 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) [8] 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 [8].

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

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

where $A = \text{diag}\ \{a_1, \ldots, a_D\}$ is a transformation matrix, $\mathbf{b} = [b_1, \ldots, b_D]^T$ 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

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

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); $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$ is the distorted features extracted 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 $h_j(f_\nu(\mathbf{y}_t))$ is the posterior probability given by

$$h_j(f_\nu(\mathbf{y}_t)) = P(j|f_\nu(\mathbf{y}_t), \Lambda, \nu) = \frac{\pi_j p(f_\nu(\mathbf{y}_t)|\boldsymbol{\mu}_j, \Sigma_j)}{\sum_{l=1}^{M_c} \pi_l p(f_\nu(\mathbf{y}_t)|\boldsymbol{\mu}_l, \Sigma_l)},$$

where

$$p(f_\nu(\mathbf{y}_t)|\boldsymbol{\mu}_j, \Sigma_j) = (2\pi)^{-\frac{D}{2}}|\Sigma_j|^{-\frac{1}{2}} \cdot \exp\left\{-\frac{1}{2}(f_\nu(\mathbf{y}_t) - \boldsymbol{\mu}_j)^T \Sigma_j^{-1}(f_\nu(\mathbf{y}_t) - \boldsymbol{\mu}_j)\right\}. \tag{3}$$

Maximizing $Q(\nu'|\nu)$ with respect to ν' leads to the following close-form solution for $\{a_i'\}$ and $\{b_i'\}$ for $i = 1, \ldots, D$ in the M-step of the EM algorithm [16]:

$$b_i' = \frac{p_i - q_i a_i'}{r_i} \tag{4}$$

and

$$\left(s_i - \frac{q_i^2}{r_i}\right) a_i'^2 + \left(\frac{q_i p_i}{r_i} - u_i\right) a_i' - T = 0 \tag{5}$$

where

$$
\begin{aligned}
p_i &= \sum_{t=1}^{T} \sum_{j=1}^{M_c} h_j(f_\nu(\mathbf{y}_t)) \mu_{ji} \sigma_{ji}^{-2} \\
q_i &= \sum_{t=1}^{T} \sum_{j=1}^{M_c} h_j(f_\nu(\mathbf{y}_t)) y_{ti} \sigma_{ji}^{-2} \\
r_i &= \sum_{t=1}^{T} \sum_{j=1}^{M_c} h_j(f_\nu(\mathbf{y}_t)) \sigma_{ji}^{-2} \\
s_i &= \sum_{t=1}^{T} \sum_{j=1}^{M_c} h_j(f_\nu(\mathbf{y}_t)) y_{ti}^2 \sigma_{ji}^{-2} \\
u_i &= \sum_{t=1}^{T} \sum_{j=1}^{M_c} h_j(f_\nu(\mathbf{y}_t)) \mu_{ji} y_{ti} \sigma_{ji}^{-2}.
\end{aligned} \tag{6}
$$

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 $\{h_j(f_\nu(\mathbf{y}_t))\}_{j=1}^{M_c}$ involve 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

$$
Q(\nu'|\nu) = \sum_{t=1}^{T} \sum_{j \in \mathcal{C}} h_j(f_\nu(\mathbf{y}_t)) \log \left\{ \frac{p(f_{\nu'}(\mathbf{y}_t)|\boldsymbol{\mu}_j, \Sigma_j)}{|J_{\nu'}(\mathbf{y}_t)|} \right\} \tag{7}
$$

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

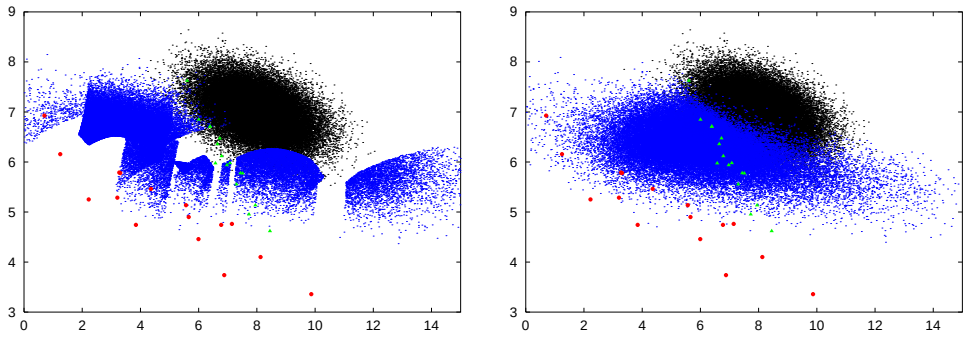
4 Probabilistic Feature Mapping

Feature mapping [7] is a non-blind technique because it requires a handset detector to identify the channel type during verification. In feature mapping, the transformation is based on the top-1 Gaussian only. Specifically, let GMM $\Lambda^{\mathrm{CD}_i} = \{\pi_j^{\mathrm{CD}_i}, \boldsymbol{\mu}_j^{\mathrm{CD}_i}, \Sigma_j^{\mathrm{CD}_i}\}_{j=1}^{M}$, be an M-mixture channel-dependent GMM for channel i and GMM $\Lambda = \{\pi_j^{\mathrm{CI}}, \boldsymbol{\mu}_j^{\mathrm{CI}}, \Sigma_j^{\mathrm{CI}}\}_{j=1}^{M}$ 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

$$
\mathbf{x} = \left(\mathbf{y} - \boldsymbol{\mu}_k^{\mathrm{CD}_i}\right) \frac{\sigma_k^{\mathrm{CI}}}{\sigma_k^{\mathrm{CD}_i}} + \boldsymbol{\mu}_k^{\mathrm{CI}}, \tag{8}
$$

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

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).



(a) Feature Mapping (b) Probabilistic Feature Mapping

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 • and green triangles ▲ represent the centers of the root model and channel-dependent model, respectively.

More specifically, we have

$$\mathbf{x} = \sum_{j=1}^{M} g_j(\mathbf{y}) \left[\left(\mathbf{y} - \boldsymbol{\mu}_j^{\mathrm{CD}_i} \right) \frac{\sigma_j^{\mathrm{CI}}}{\sigma_j^{\mathrm{CD}_i}} + \boldsymbol{\mu}_j^{\mathrm{CI}} \right], \tag{9}$$

where

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

is the posterior probability of the j-th mixture. Note that the original feature mapping (Eq. 8) is a special case of the probabilistic feature mapping (Eq. 9).

The fast technique mentioned in Section 3 can also be applied to PFM. Specifically, Eq. 9 is rewritten as

$$\mathbf{x} = \sum_{j \in \mathcal{C}} g_j(\mathbf{y}) \left[\left(\mathbf{y} - \boldsymbol{\mu}_j^{\mathrm{CD}_i} \right) \frac{\sigma_j^{\mathrm{CI}}}{\sigma_j^{\mathrm{CD}_i}} + \boldsymbol{\mu}_j^{\mathrm{CI}} \right], \tag{10}$$

where $\mathcal{C}$ contains the indexes of the top-C Gaussians. Fig. 2 shows the mean posterior probabilities $g_j(\mathbf{y})$ of the top-10 Gaussians based on an utterance with 932 frames. Evidently, the posterior probability is large for the first few Gaussians only. In particular, the posterior probability of the 5-th Gaussian is only 5% of that the first one. This suggests that only the top few Gaussians have significant influence on the transformation. Based on this observation, C was set to 5 in this work, i.e., only the top-5 Gaussians will be considered in the fast PFM.

The idea of PFM can be illustrated by a 2-D hypothetical example as shown in Fig. 1. In the figure, the black dots represent patterns from a specific channel and the red circles and green triangles represent the centers of the root

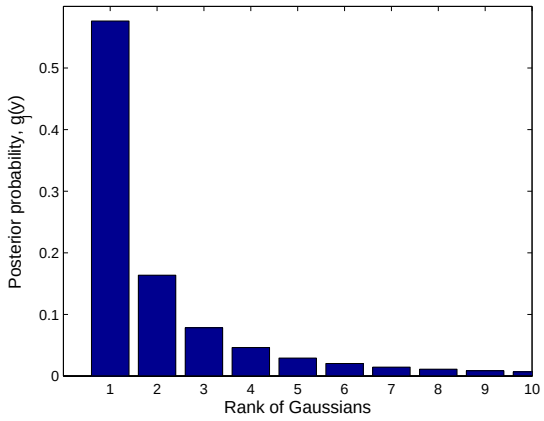


Fig. 2. Average posterior probabilities $g_j(\mathbf{y})$ of the top-10 Gaussians. The probabilities are based on a 932-frame utterance obtained from a carbon button handset.

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 a clustering effect in the transformed patterns. Fig. 1(b) shows that this clustering effect can be largely reduced by using PFM. According to Eq. 9, 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.

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) [17] 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 [18]. 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. 3 shows the process of creating speaker models from the UBMs.

For feature warping and short-time Gaussianization, features from both enrollment and verification utterances were either warped or Gaussianized. These warped or Gaussianized features were used to create the background models using the EM algorithm. Then, MAP adaptation was applied to the background models to create the target-speaker models using warped or Gaussianized features.

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 $\text{CD}_i \in \{\text{cb, el}\}$.[1] During verification, these models were then used for calculating the mapping function (Eq. 8, 9, or 10) that transforms the distorted features derived from the evaluation utterances

[1] cb: carbon button handsets; el: electret handsets. The relevant factors for adapting the means, variances, and mixture weights were set to 16.

in NIST00 to fit the gender-dependent root models Λ_g. Fig. 3 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 transformation parameters (A and $\mathbf{b}$ 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 $M_c = 64$, Eq. 4, and Eq. 5) 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 ($f_{\nu_{cb\to el}}(\mathbf{y})$ or $f_{\nu_{el\to cb}}(\mathbf{y})$) to reduce the acoustic mismatch between the training and testing features.

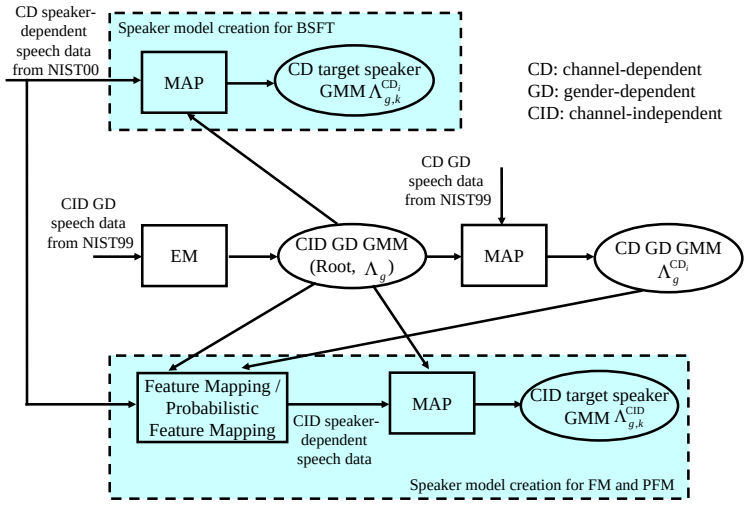


Fig. 3. Procedures of creating target speaker models for FM, PFM, and BSFT.

5.4 Fusion of PFM, BSFT and STG

Because BSFT, STG, 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 verify this hypothesis, we trained a 2-input linear SVM using the training set of NIST00 to classify scores vectors $\mathbf{s} = [s_{\text{PFM}} \quad s_{\text{fBSFT}}]^T$ into speaker class and impostor class, where s_{PFM} and s_{fBSFT} are scores obtained from a PFM-based and a fBSFT-based system, respectively. The distances of $\mathbf{s}$ from the decision hyperplane of the SVM are then used for plotting DET curves. Similarly, we have also fused the scores of fBSFT and STG.

6 Results and Discussions

Table 1, Fig. 5, and Fig. 6 show the computation time, computational complexity, EER performance, and DET performance of various transformation methods.

6.1 Computational Complexity and Computation Time

Table 1 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 complexity of CMS, which amounts to $\mathcal{O}(PT)$, is common to all methods. For BSFT and fBSFT, the computation of the posterior probabilities $h_j(f_\nu(\mathbf{y}_t))$ and the actual transformation requires $\mathcal{O}(TDM_c)$ and $\mathcal{O}(TD)$ operations, respectively. The computation saving of fBSFT comes from the estimation of the transformation parameters A and $\mathbf{b}$. More precisely, BSFT requires $\mathcal{O}(TDM_c)$ operations, whereas fBSFT only requires $\mathcal{O}(TCM_c)$ operations for finding the C maximum posteriors and $\mathcal{O}(TDC)$ operations to find A and $\mathbf{b}$.

Assuming Reynolds' fast scoring approach is used, the handset detection in FM, PFM, fPFM, and SFT requires $\mathcal{O}(TDM + TDC)$ operation. The transformation in FM requires $\mathcal{O}(TD)$ operations because only the top-1 Gaussian is involved. On the other hand, PFM and fPFM require $\mathcal{O}(TDM)$ and $\mathcal{O}(TDC)$ operations, respectively, because in PFM all mixtures are involved and in fPFM only the top-C mixtures are involved. Note that fPFM also requires $\mathcal{O}(TCM)$ operations to find the top-C Gaussians. fPFM is considerably faster than PFM because $DM \gg CM + CD$. The transformation of SFT requires $\mathcal{O}(TD)$ operations. Because SFT does not require to find the top-1 Gaussian and the transformation is computationally light, it is the fastest among all non-blind approaches.

Both FW and STG require $\mathcal{O}(TPW \log W)$ operations for sorting the feature values in the warping window of length W (assuming quicksort [19] is used), $\mathcal{O}(TPZ)$ operations for looking up a z-table of size Z (assuming linear search is used), and $\mathcal{O}(TPK)$ operations for computing delta cepstra using a window of length K. STG requires additional $\mathcal{O}(TP^2)$ operations for performing the linear transformation before warping the features.

The last column of Table 1 shows that CMS is the fastest among all investigated methods because subtracting the mean from feature vectors is a very simple procedure. PFM is considerably slower than FM because for large M

(1024 here), the time spent on computing Eq. 9 is considerably longer than that on computing Eq. 8. 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.

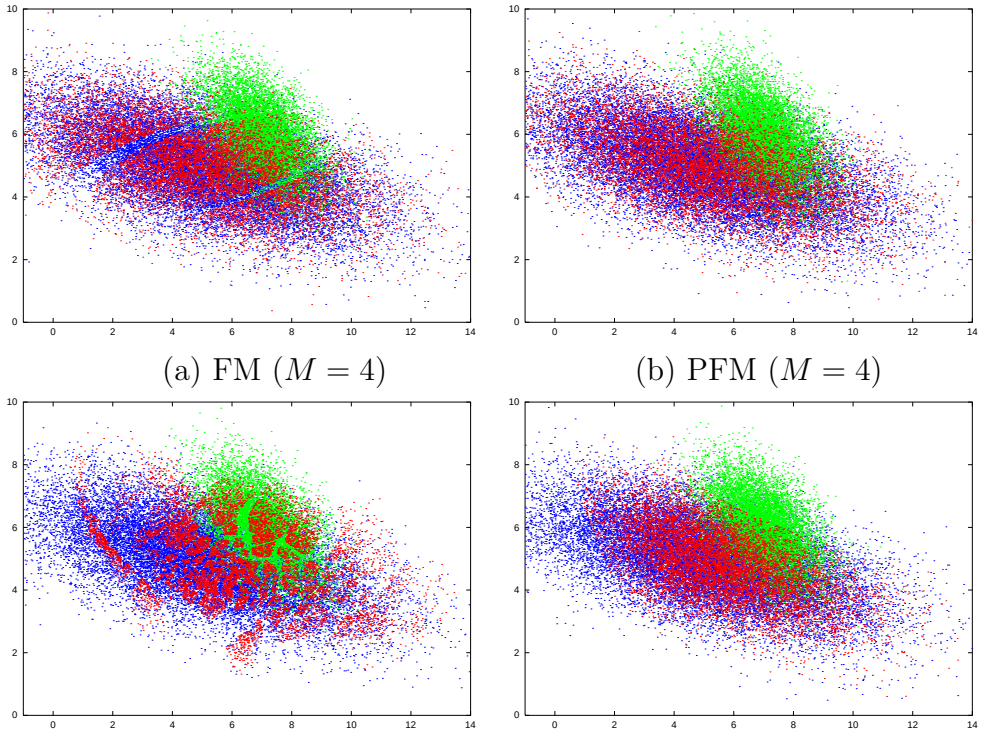
6.2 EER and DET Performance

Fig. 6 and Table 1 show the performance of various transformation techniques and the fusion of PFM and fBSFT and the fusion of STG and fBSFT. Evidently, all methods show significant reduction in error rates when compared to CMS. The DET curves also show that these methods, in particular feature warping and Gaussianization, outperform CMS at all operating points.

The p-values [20] [2] in Table 2 suggest that there is no significant difference between the EERs of the following pairs: FM-PFM, FM-BSFT, FM-fBSFT, FM-fPFM, FM-SFT, BSFT-fBSFT, BSFT-fPFM, fBSFT-fPFM, and PFM-SFT. On the other hand, fusion of PFM and fBSFT and fusion of STG and fBSFT can reduce EERs 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.

The PFM is theoretically better than FM because the former takes all Gaussians in the GMMs into consideration and avoids sharp changes in transformation. However, Table 1 shows that PFM is only slightly better than FM in terms of EER and minimum DCF. To investigate why this is the case and the condition under which PFM becomes superior to FM, we compared FM and PFM using small and large number of mixtures in a hypothetical 2-D problem, and the results are shown in Fig. 4. Evidently, the clustering effect in FM becomes more prominent when the number of mixtures is large. Some of the transformed patterns (red dots) in Fig. 4(c) even overlap with the channel-dependent patterns (green dots). The PFM, on the other hand, is able to transform the channel-dependent data to the region occupied by the channel-independent space for both small and large number of mixtures, suggesting that PFM has merit provided that the number of mixtures is sufficiently large. Therefore, whether increasing the number of mixtures to say 2048 will bring more performance improvement for real speech data worth further investigation.

[2] A p-value less than 0.005 means that the difference in EERs between the two corresponding transformation methods is statistically significant with a confidence level of 99.5%.



(a) FM ($M = 4$) (b) PFM ($M = 4$)

(c) FM ($M = 256$) (d) PFM ($M = 256$)

Fig. 4. A 2-D hypothetical example comparing feature mapping (FM) and probabilistic feature mapping (PFM) under small and large numbers of mixtures in the GMMs. (a) and (b) 4 mixtures. (c) and (d) 256 mixtures. *Green dots*: channel-dependent patterns. *Blue dots*: channel-independent patterns. *Red dots*: transformed patterns. Both FM and PFM attempt to transform the channel-dependent patterns to the space occupied by the channel-independent patterns.

Feature Warping and short-time Gaussianization are among the best method in terms of DET performance and EER, but their computation time is longer than many others. Because BSFT, fBSFT, FW, and STG do not require handset detection, they are more flexible than FM, PFM, and SFT. 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 their high performance in terms of EER and DET performance and their moderate complexity, FW and STG appear to be the best channel compensation method (among those that have been investigated in this work) for practical implementation of speaker verification system. On the other hand, if computation time is a concern, fBSFT is an appropriate choice because it achieves reasonable performance at a low computational complexity. If it is necessary to reduce the error rate further, we may fuse the scores of fBSFT and STG.

The software for BSFT, FM, FW, and STG can be downloaded from http://www.eie.polyu.edu.hk/∼mwmak/programs/feaTx.tgz.

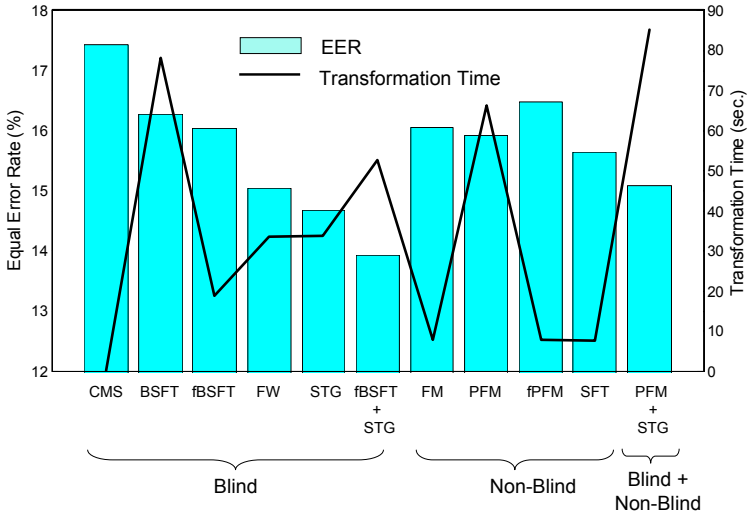


Fig. 5. Transformation time and EER performance of various transformation methods.

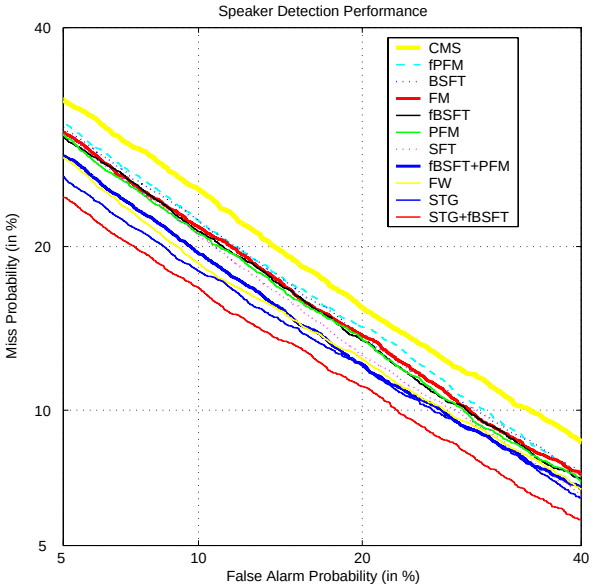


Fig. 6. DET curves comparing speaker verification performance of different feature transformation methods. See the caption of Table 1 for the full name of the acronyms. Legends are arranged in descending order of EER.

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.

Acknowledgement

This work 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).

References

[1] A. C. Surendran, C. H. Lee, M. Rahim, Nonlinear compensation for stochastic matching, IEEE Trans. on Speech and Audio Processing 7 (6) (1999) 643–655.

[2] K. K. Yiu, M. W. Mak, M. C. Cheung, S. Y. Kung, Blind stochastic feature transformation for channel robust speaker verification, J. of VLSI Signal Processing 42 (2) (2006) 117–126.

[3] F. Beaufays, M. Weintraub, Model transformation for robust speaker recognition from telephone data, in: Proc. ICASSP'97, 1997, pp. 21–24.

[4] K. K. Yiu, M. W. Mak, 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] D. A. Reynolds, Channel robust speaker verification via feature mapping, in: IEEE ICASSP, Vol. 2, 2003, pp. 6–10.

[8] S. Y. Kung, M. W. Mak, S. H. Lin, Biometric Authentication: A Machine Learning Approach, Prentice Hall, Upper Saddle River, New Jersey, 2005.

[9] J. Pelecanos, 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, R. Gopinath, Short-time Gaussianization for robust speaker verification, in: Proc. ICASSP'02, Vol. 1, 2002, pp. 681–684.

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

[12] H. Hermansky, N. Morgan, RASTA processing of speech, IEEE Trans. on Speech and Audio Processing 2 (4) (1994) 578–589.

[13] L. P. Heck, Y. Konig, M. K. Sonmez, M. Weintraub, Robustness to telephone handset distortion in speaker recognition by discriminative feature design, in: Speech Communication, Vol. 31, 2000, pp. 181–192.

[14] T. F. Quatieri, D. A. Reynolds, G. C. O'Leary, Estimation of handset nonlinearity with application to speaker recognition, IEEE Trans. on Speech and Audio Processing 8 (5) (2000) 567–584.

[15] A. Sankar, C. H. Lee, A maximum-likelihood approach to stochastic matching for robust speech recognition, IEEE Trans. on Speech and Audio Processing 4 (3) (1996) 190–202.

[16] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. of Royal Statistical Soc., Ser. B. 39 (1) (1977) 1–38.

[17] S. B. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE Trans. on ASSP 28 (4) (1980) 357–366.

[18] D. A. Reynolds, T. F. Quatieri, R. B. Dunn, Speaker verification using adapted Gaussian mixture models, Digital Signal Processing 10 (2000) 19–41.

[19] M. A. Weiss, Data Structures and Algorithm Analysis in C, Benjamin/Cummings Pub. Company, Inc., 1993.

[20] L. Gillick, S. Cox, Some statistical issues in the comparison of speech recognition algorithms, in: Proc. ICASSP'89, 1989, pp. 532–535.

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), feature warping (FW), short-time Gaussianization (STG), fusion of PFM and fBSFT, and fusion of STG and fBSFT. 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).

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
	FW	15.04	0.000	0.0573	$\mathcal{O}(PT + TPW\log W + TPZ+TPK)$	33.50
	STG	14.67	0.000	0.0553	$\mathcal{O}(PT + TP^2 + TPW\log W + TPZ + TPK)$	33.80
	STG+fBSFT	13.93	0.000	0.0514	$\mathcal{O}(PT + TP^2 + TPW\log W + TPZ + TPK+TDM_c+TCM_c+TDC+TD+DS)$	52.64
Non-Blind	FM	16.05	–	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+fBSFT	15.09	0.000	0.0540	$\mathcal{O}(PT+2TDM+2TDC+TDM_c+TCM_c+TD+DS)$	85.03

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

W : Length of warping window in feature warping and Gaussianization (= 301)

Z : Size of the z-table in feature warping and Gaussianization (= 40001)

K : Window size for calculating delta cepstra (= 7)

	FM	PFM	BSFT	fBSFT	fPFM	SFT	PFM + fBSFT	FW	STG	STG + fBSFT
CMS	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
FM	–	0.5666	0.1888	0.4016	0.0094	0.1029	0.0000	0.0000	0.0000	0.0000
PFM	–	–	0.0058	0.0403	0.0000	0.0535	0.0003	0.0000	0.0000	0.0000
BSFT	–	–	–	0.1002	0.1011	0.0000	0.0000	0.0000	0.0000	0.0000
fBSFT	–	–	–	–	0.0197	0.0001	0.0000	0.0000	0.0000	0.0000
fPFM	–	–	–	–	–	0.0000	0.0000	0.0000	0.0000	0.0000
SFT	–	–	–	–	–	–	0.0090	0.0026	0.0000	0.0000
PFM+fBSFT	–	–	–	–	–	–	–	0.5906	0.0001	0.0000
FW	–	–	–	–	–	–	–	–	0.0000	0.0000
STG	–	–	–	–	–	–	–	–	–	0.0000

Table 2
p-values of McNemar's tests on the differences between the equal error rates of various transformation methods. For each entry, $p < 0.005$ means that the difference between the EERs of the two corresponding transformation methods is statistically significant at a confidence level of 99.5%.