Chapter 4
Stochastic Feature Compensation for Robust Speaker Verification

Abstract This chapter explores the impact of standard stereo-based stochastic feature compensation (SFC) methods for robust speaker verification in uniform noisy environments. In this work, SFC using independent as well as joint probability models are explored for compensating the effect of noise. Integration of a SFC stage in the GMM-UBM framework is proposed for speaker verification evaluation under mismatched conditions.

The choice of features used for speaker recognition (SR) tasks is usually a tradeoff between accuracy, implementation costs and robustness. Short-term spectral or vocal tract features (e.g., MFCC) are the most extensively used for SR tasks due to their high speaker discriminative properties [1]. However, they are highly susceptible to noise-degradation and are therefore aided by compensation procedures in most SR applications [2,3]. The role of feature compensation was briefly introduced in Chap. 1. Despite the existence of inherent robust features, SR applications often prefer simple spectral features due to their ease of extraction. Such applications essentially require feature compensation methods for noise-robustness.

The discussion about the filtering-based feature compensation methods (e.g., CMS [4], RASTA [5]) in Chap. 2 revealed that they are specifically designed for cepstral features and are commonly applied for suppressing channel effects. However, filtering is often inadequate for additive background environments where the log-spectral effect is ineffective. The application of model-based compensation schemes (e.g., SS [6], CDCN [7]) are likewise compromised due to the unavailability of a noise-model and high amount of training data.

The data-driven feature compensation methods offer a number of significant advantages compared to the other two categories. Firstly, they are independent of any analytical representation about the nature of the noise-corruption process. Secondly, they can better model the noise-effects due to their stochastic nature. Lastly, their performance is consistent across different environments. The only apparent drawback of applying these methods is the requirement of stereo data

K.S. Rao and S. Sarkar, *Robust Speaker Recognition in Noisy Environments*,
SpringerBriefs in Electrical and Computer Engineering,
DOI 10.1007/978-3-319-07130-5_4, © The Author(s) 2014

which can be interpreted as having a priori knowledge about the test environment. Despite such drawbacks, these techniques have been successfully used for far-field speech recognition tasks. To the best of the author's knowledge, the effect of these feature compensation methods have not been studied for robust speaker verification (SV) tasks. The application of standard stochastic feature compensation methods in a SV framework is proposed in this chapter. The significance of the proposed approach is demonstrated through a set of conducted experiments in simulated noisy environment.

The rest of the chapter is organized as follows. Section 4.1 gives a brief introduction to stochastic feature compensation, Sects. 4.2.1–4.3.2 provide detailed description of the feature compensation methods considered in the work [8], the proposed SV framework is discussed in Sect. 4.4 followed by a brief summary of the present work in Sect. 4.5.

4.1 Stochastic Feature Compensation (SFC)

Since accurate enumeration of the environmental effects on speech is a non-trivial task, a simplified form of speech signal degradation based on additive and convolutional channel noise is used in practice. Due to the random nature of noise, a given clean feature vector can generate different noisy feature vectors, and vice-versa, which causes an uncertainty. Conventionally, Gaussian Mixture Models (GMMs) are used to represent the cepstral distribution. The additive noise in general alters the distribution of mel frequency cepstral coefficients (MFCCs) by reducing the variance of each Gaussian component while the convolutional noise shifts the mean vectors.

Stochastic feature compensation (SFC) methods are independent of any mathematical structure of noise degradation. They model stereo training data using GMMs. Given a noisy test feature vector y_t, a minimum mean squared error (MMSE) criterion is used to estimate a clean vector $\hat{x}_t$ as follows

$$\hat{x}_t = E[x|y_t] = \int_X x p(x|y_t) \mathrm{d}x \tag{4.1}$$

where x is a random variable representing clean feature vectors and $p(x|y_t)$ is the conditional probability distribution function (pdf) of x given y_t. Depending on the nature of the feature compensation algorithm, the two broad approaches of deriving $p(x|y_t)$ can be categorized as (i) Independent probability modeling and (ii) Joint probability modeling. The independent probability modeling methods construct individual GMMs for clean and noisy data. The effect of noise is represented as additive terms to the mean vectors and covariance matrices of the GMMs. The conditional pdf is derived based on numerical approximations using the additive terms. Alternatively, joint probability models construct a single GMM using stacked

noisy and clean feature vectors of the stereo data. This is followed by deriving an exact conditional pdf and estimation of clean speech vectors. Each of these methods are discussed in details in the following two sections

4.2 SFC Using Independent Probability Models

Figure 4.1 illustrates the independent probability model based SFC process. The main steps of the process can be outlined as follows

1. Firstly individual GMMs are built for clean vectors X_t and noisy vectors Y_t as follows

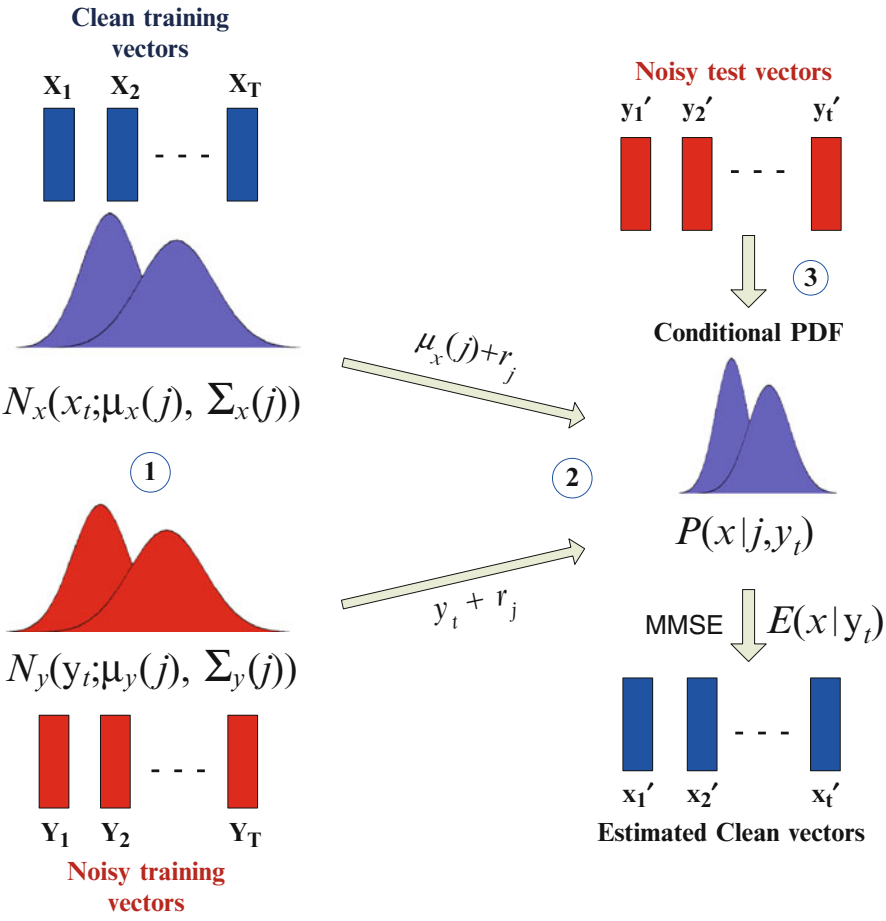

Fig. 4.1 Stochastic feature compensation using independent probability models

$$p(x_t) = \sum_{j=1}^{M} w_x(j)\mathcal{N}_x(x_t; \mu_x(j), \Sigma_x(j)) \tag{4.2}$$

$$p(y_t) = \sum_{j=1}^{M} w_y(j)\mathcal{N}_y(y_t; \mu_y(j), \Sigma_y(j)) \tag{4.3}$$

where $w(j)$, $\mu(j)$ and $\Sigma(j)$ denotes the weight, mean vector and covariance matrix of the jth Gaussian component and M is the total number of components.

2. The conditional pdf $p(x|j, y_t)$ is then approximated by means of additive factors r_j to the clean or noisy training vectors. The values of the additive terms are determined my maximizing the likelihood of the training data.
3. Given a set of noisy test vectors, the equivalent set of clean vectors are estimated by MMSE

The GMM representations given by Eqs. (4.2) and (4.3), are used in the remaining chapter. In the following sections three standard independent probability model based SFC techniques used for robust speech recognition tasks are discussed briefly. Each of the methods differ in the way by which they derive $p(x|y_t)$ and thereby estimate $\widehat{x_t}$. Detailed derivations of the additive terms and the MMSE estimator for each of these algorithms can be found in Appendix A.

4.2.1 *Multivariate Gaussian-Based Cepstral Normalization (RATZ)*

The RATZ algorithm [9], derives the required MMSE clean feature estimate in three stages. In the first stage, the clean feature vectors are used to train a GMM as in Eq. (4.2) using the standard Expectation Maximization (EM) algorithm. The second stage consists of estimating the statistics of the noise-degraded speech by applying appropriate correction vectors to the mean and covariance matrices of the clean speech pdf. The additive correction vectors, which model environmental effect are in turn estimated by maximizing the likelihood of the noisy feature vectors. Finally, given a noisy test feature vector, a MMSE estimate of clean speech is made using the correction vectors learned during the training phase. Given a sequence of $\mathbf{T}$ noisy MFCC vectors $Y = [y_1, y_2, \ldots y_{\mathbf{T}}]$, the log-likelihood is given by

$$\begin{aligned} L(Y) = \log \prod_{t=1}^{\mathbf{T}} p(y_t) &= \sum_{t=1}^{\mathbf{T}} \log \sum_{j=1}^{M} w_y(j)\mathcal{N}_y(y_t; \mu_y(j), \Sigma_y(j)) \\ &= \sum_{t=1}^{\mathbf{T}} \log \sum_{j=1}^{M} w_y(j)\mathcal{N}_y(y_t; \mu_x(j) + r_j, \Sigma_x(j) + R_j) \end{aligned} \tag{4.4}$$

where r_j and R_j are the correction vectors for the jth Gaussian component of the clean speech pdf. The complete set of unknown bias vectors is iteratively estimated by maximizing L using an EM algorithm. Details of the EM algorithm have been outlined in Appendix A. The solutions obtained are given by the following equations

$$\hat{r}_j = \frac{\sum_{t=1}^{\mathbf{T}} p(s_y(j)|y_t, \phi)(y_t - \mu_x(j))}{\sum_{t=1}^{\mathbf{T}} p(s_y(j)|y_t, \phi)} \tag{4.5}$$

$$\hat{R}_j = \frac{\sum_{t=1}^{\mathbf{T}} p(s_y(j)|y_t, \phi)\{(y_t - \mu_x(j) - \hat{r}_j)(y_t - \mu_x(j) - \hat{r}_j)^T - \Sigma_x(j)\}}{\sum_{t=1}^{\mathbf{T}} p(s_y(j)|y_t, \phi)} \tag{4.6}$$

where $p(s_y(j)|y_t, \phi)$ is the posterior probability of the latent noisy GMM component $s_y(j)$ given y_t, $\phi = \{r_j, R_j\}$ is the set of model parameters and T denotes matrix transpose. It was studied by Moreno et al. [9] that in case of stereo recordings, a one-one correspondence of the each Gaussian component of the noisy speech GMM and clean speech GMM can be established. This is done by assuming *posterior invariance* which states that the posterior probabilities of each GMM component with respect to a clean vector and its noisy equivalent vector are equal. This assumption, although less reliable in low SNR conditions suggest that each Gaussian undergoes the same shift and negligible compression. It gives a convenient approximation of $p(s_y(j)|y_t, \phi)$ as follows

$$\begin{aligned} p(s_y(j)|y_t, \phi) &= \frac{p(s_y(j))p(y_t|s_y(j), \phi)}{\sum_{k=1}^{M} p(s_y(k))p(y_t|s_y(k), \phi)} \\ &= \frac{p(s_x(j))p(x_t|s_x(j))}{\sum_{k=1}^{M} p(s_x(k))p(x_t|s_x(k))} \\ &= \frac{w_x(j)\mathcal{N}_x(x_t; \mu_x(j), \Sigma_x(j))}{\sum_{j=1}^{M} w_x(j)\mathcal{N}_x(x_t; \mu_x(j), \Sigma_x(j))} \end{aligned} \tag{4.7}$$

Given the above relation, Eqs. (4.5) and (4.6) can now be approximated as

$$\hat{r}_j = \frac{\sum_{t=1}^{\mathbf{T}} p(s_x(j)|x_t)(y_t - x_t)}{\sum_{t=1}^{\mathbf{T}} p(s_x(j)|x_t)} \tag{4.8}$$

$$\hat{R}_j = \frac{\sum_{t=1}^{\mathbf{T}} p(s_x(j)|x_t)\{(y_t - x_t - \hat{r}_j)(y_t - x_t - \hat{r}_j)^T - \Sigma_x(j)\}}{\sum_{t=1}^{\mathbf{T}} p(s_x(j)|x_t)} \tag{4.9}$$

Since the above equations do not have ϕ in the right hand side, the solutions are non-iterative. The environmental effects on clean speech x in MFCC domain are modeled as additive linear correction vectors $r(x)$. The MMSE estimate for clean speech $\hat{x}_t$ given a noisy test vector y_t is calculated by Eq. (4.1). The conditional mean is solved using a numerical approximation as follows

$$\hat{x}_t = E[x|y_t] = y_t - \sum_{j=1}^{M} p(j|y_t) r_j \tag{4.10}$$

4.2.2 *Stereo Piece-Wise Linear Compensation for Environment (SPLICE)*

The effectiveness of the RATZ algorithm depends on the posterior invariance assumption made in Eq. (4.7). However in low SNR conditions this assumption becomes unrealistic since the Gaussian pdfs of noisy speech are compressed in different amounts due to changes in its variance. As an alternative, the SPLICE algorithm proposed in [10] models the noisy feature space as given by the following equation

$$p(y_t) = \sum_{j=1}^{M} p(j) p(y_t|j) \tag{4.11}$$

where $p(j)$ is the prior probability of the Gaussian component j mathematically equivalent to the component weight $w_y(j)$ and $p(y|j)$ is the multivariate Gaussian $\mathcal{N}_y(y_t; \mu_y(j), \Sigma_y(j))$ as given in Eq. (4.3). A distinct advantage of SPLICE compared to other model-based feature enhancement techniques like Spectral Subtraction, is its consistent performance in non-stationary environments. Feature compensation using SPLICE is based on a two simple assumptions. Firstly, a clean MFCC vector x_t generated by each discrete Gaussian component j can be approximated in terms of its noisy counterpart y_t. This is often termed as piece-wise linear approximation. Secondly the conditional pdf of clean speech vectors given the noisy speech vectors and Gaussian component j is also a multivariate Gaussian distribution. The mean of the resultant distribution is assumed to shifted by the corrective vector r_j as follows

$$p(x|j, y_t) = \mathcal{N}_y(x; y_t + r_j, \Gamma_j) \tag{4.12}$$

Estimation of the parameters r_j and Γ_j are based on maximum likelihood training similar to that of RATZ (Eq. 4.4) using an EM algorithm (outlined in Appendix A). The solutions are given by

$$\hat{r}_j = \frac{\sum_{t=1}^{\mathbf{T}} p(j|y_t)(x_t - y_t)}{\sum_{t=1}^{\mathbf{T}} p(j|y_t)} \tag{4.13}$$

$$\Gamma_j = \frac{\sum_{t=1}^{\mathbf{T}} p(j|y_t)\{(x_t - y_t)(x_t - y_t)^T - \hat{r}_j \hat{r}_j^T\}}{\sum_{t=1}^{\mathbf{T}} p(j|y_t)} \tag{4.14}$$

where $p(j|y_t)$ is the posterior probability of component j given y_t

$$p(j|y_t) = \frac{p(j)p(y_t|j)}{\sum_{j=1}^{M} p(j)p(y_t|j)} \tag{4.15}$$

For stereo training data, the solution of Eqs. (4.13) and (4.14) are non-iterative. The MMSE estimate for clean speech from the noisy speech pdf is then given by

$$\hat{x}_t = E[x|y_t] = y_t + \sum_{j=1}^{M} p(j|y_t) r_j \tag{4.16}$$

The approximation of the mean of the conditional pdf in Eq. (4.12) using additive terms r_j is often considered to be a limitation of the SPLICE framework. An accurate estimation of the conditional mean would require joint probability modeling of the clean and noisy vectors followed by estimating MLLR-type transforms [11]. Despite these drawbacks, SPLICE is commonly applied for pre-processing feature vectors in robust speech recognition tasks.

4.2.3 *Multivariate Model Based Cepstral Normalization (MMCN)*

The previous techniques discussed so far either models the clean feature space (e.g., RATZ) or the noisy feature space (e.g., SPLICE) using GMMs. A corrective bias vector for each GMM component is trained by weighing the difference between clean and noisy feature vector pairs with normalized posterior probabilities. However, in realistic situations when there are multiple types of environment in the

noisy space, estimates based on single GMM posteriors might be erroneous. The Multi-Environment Model based LInear Normalization (MEMLIN) algorithm [12] aims to enhance performance accuracy by modeling both noisy and clean spaces in parallel. The noisy feature space is divided into several basic environments and modeled with individual GMM.

$$p_e(y_t) = \sum_{s_y^e=1}^{M} p(y_t|s_y^e)p(s_y^e) \tag{4.17}$$

where s_y^e denotes the latent Gaussian component for the noisy GMM trained in environment indexed by e, $p_e(y_t|s_y^e)$ and $p_e(s_y^e)$ denote the Gaussian pdf for the s_y^eth component and its prior probability, respectively as shown below

$$p(y_t|s_y) = \mathcal{N}(y_t; \mu(s_y^e), \Sigma(s_y^e)) \tag{4.18}$$

$$p(s_y^e) = w_y^e \tag{4.19}$$

The clean feature space is modeled by a single GMM and has a similar structure as that of Eq. (4.2).

$$p(x_t) = \sum_{s_x=1}^{M} p(x_t|s_x)p(s_x) \tag{4.20}$$

The objective is to learn the difference between clean and noisy feature vectors associated with a pair of Gaussians (one for a clean model, and the other one for a noisy model), for each basic environment. The bias vector transformations are computed independently for each basic environment. Alike SPLICE, MEMLIN assumes that each clean feature vector x_t is approximated by a linear function of the noisy feature vector y_t and an additive bias vector $r(s_x, s_y^e)$. However unlike SPLICE, the additive vectors are now a function of both clean and noisy GMM components for a particular environment. The second assumption approximates the conditional pdf of x given y_t as a multivariate Gaussian with covariance matrix $\Sigma(s_x, s_y^e)$ and mean given by a linear transformation of the environment-dependent noisy vector, as follows

$$p(x|y_t, s_y^e, s_x) = \mathcal{N}(x|y_t - \sum_e p(e|y_t)r(s_x, s_y^e), \Sigma(s_x, s_y^e)) \tag{4.21}$$

where $p(e|y_t)$ and $r(s_x, s_y^e)$ are the posterior probability of environment e given y_t and the additive bias vector, respectively. The estimation of these factors are discussed briefly. The factor $p(e|y_t)$ is trained recursively as follows

$$p(e|y_t) = \beta p(e|y_{t-1}) + (1-\beta)\frac{p_e(y_{t-1})}{\sum_e p_e(y_{t-1})} \tag{4.22}$$

where $(0 \leq \beta \leq 1)$ is a constant and $p(e|y_0)$ is uniform across all environments. The $r(s_x, s_y^e)$ factor is obtained by maximizing the likelihood of noisy feature vector with respect to $r(s_x, s_y^e)$, using the standard EM algorithm. Given the stereo training data for environment e which comprises the noisy vectors $Y_e = \{y_{t_e}\}_{t_e=1}^{\mathbf{T_e}}$ and clean vectors $X_e = \{x_{t_e}\}_{t_e=1}^{\mathbf{T_e}}$, the complete data log-likelihood of Y_e is given by the following equation

$$L(Y_e) = \sum_{t_e=1}^{\mathbf{T_e}} \log \sum_{s_y^e=1}^{M} p(s_y^e) \mathcal{N}_y(y_{t_e}; \mu(s_y^e) + r(s_x, s_y^e), \Sigma(s_x, s_y^e)) \tag{4.23}$$

Maximizing the above equation with respect to $r(s_x, s_y^e)$ gives

$$r(s_x, s_y^e) = \frac{\sum_{t_e=1}^{\mathbf{T_e}} p(s_x|x_{t_e}) p(s_y^e|y_{t_e})(y_{t_e} - x_{t_e})}{\sum_{t_e=1}^{\mathbf{T_e}} p(s_x|x_{t_e}) p(s_y^e|y_{t_e})} \tag{4.24}$$

where $p(s_x|x_{t_e})$ the posterior probability of Gaussian s_x with respect to clean vector x_t. Similarly $p(s_y^e|y_{t_e})$ is the posterior probability of Gaussian s_y^e with respect to noisy vector y_t. These can be easily calculated using Eqs. (4.20) and (4.17), respectively as follows

$$p(s_x|x_{t_e}) = \frac{p(x_{t_e}|s_x) p(s_x)}{\sum_{s_x=1}^{M} p(x_{t_e}|s_x) p(s_x)} \tag{4.25}$$

$$p(s_y^e|y_{t_e}) = \frac{p(y_t|s_y^e) p(s_y^e)}{\sum_{s_y^e=1}^{M} p(y_t|s_y^e) p(s_y^e)} \tag{4.26}$$

The resultant MMSE estimate $\hat{x}_t$ is computed as a weighted sum of all of the basic environment bias vector transformations.

$$\hat{x}_t = E[x|y_t] = y_t - \sum_{e} \sum_{s_y^e=1}^{M} \sum_{s_x=1}^{M} r(s_x, s_y^e) p(e|y_t) p(s_y^e|y_t) p(s_x|s_y^e, y_t, e) \tag{4.27}$$

The above equation introduces a new factor $p(s_x|s_y^e, y_t, e)$ known as cross probability model. It compensates for the mismatch that occurs when the Gaussian component s_x associated with clean vector x_t is different from the Gaussian component s_y^e associated with corresponding noisy vector y_t. For simplicity the time dependency with y_t is omitted, and the resultant factor $p(s_x|s_y^e, e)$ is estimated using relative frequency of occurrence. It is calculated as the ratio of the number of

times the most probable pair of decoded Gaussians are $\{s_x, s_y^e\}$ and the number of times s_y^e is decoded singly. The resultant form is as follows

$$p(s_x|s_y^e, e) = \frac{\sum_{t_e=1}^{\mathbf{T_e}} p(s_x|x_{t_e})\, p(s_y^e|y_{t_e})\, p(s_x)\, p(s_y^e)}{\sum_{t_e=1}^{\mathbf{T_e}} \sum_{s_x=1}^{M} p(s_x|x_{t_e})\, p(s_y^e|y_{t_e})\, p(s_x)\, p(s_y^e)} \tag{4.28}$$

The single environment version of MEMLIN is often termed as Multivariate Model based Cepstral Normalization (MMCN). It can be easily deduced that in case of single environment, the variable e can be omitted which simplifies most of the above equations. In such case the factor $p(e|y_t)$ can be entirely ignored. The scope of the present work is restricted to the single-environment version of MEMLIN i.e., MMCN.

4.3 SFC Using Joint Probability Models

The only apparent drawback of the independent probability model based SFC methods is the determination of the additive terms which may turn out be inaccurate in degraded environmental conditions. Alternatively, joint probability modeling can be used for feature compensation provided sufficient training data is available.

Figure 4.2 illustrates the independent probability model based SFC process. The main steps of the process can be outlined as follows

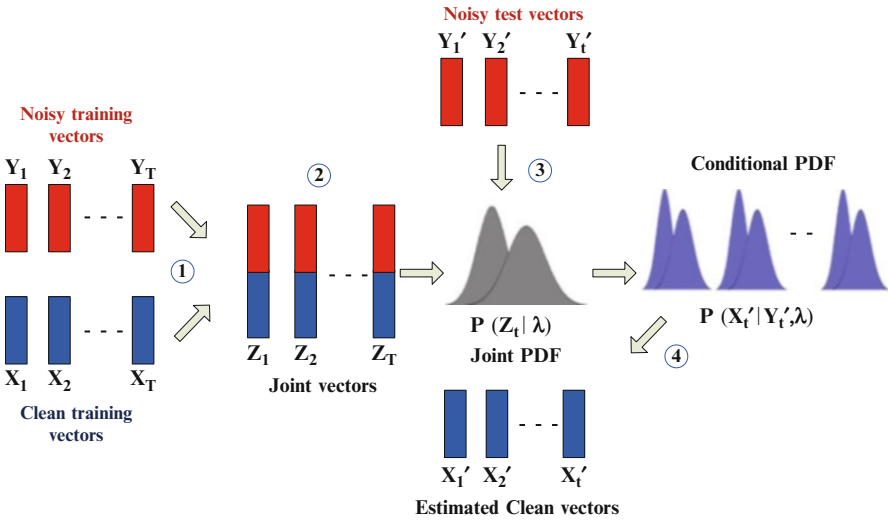

Fig. 4.2 Stochastic feature compensation using joint probability models

1. The noisy and clean training vectors are concatenated to produce joint vectors (Z)
2. The joint vectors are modeled using a single GMM which represents the joint pdf
3. The conditional pdf is derived using parameters of the joint pdf.
4. Given a noisy test vector Y_t', a clean vector X_t' is obtained based on MMSE or maximum likelihood estimate (MLE).

Two standard joint probability model based SFC methods are discussed in the following sections

4.3.1 Stereo-Based Stochastic Mapping (SSM)

The main idea of the SSM algorithm [13] is to estimate the joint probability distribution of noisy and clean feature spaces instead of modeling them independently. This eliminates the need for training the hypothesized additive bias term 'r' for each GMM component as employed by previous methods like SPLICE or MEMLIN. Unlike previous methods, concatenated pair of noisy and clean feature vectors are used as training data for GMM building. The desired transformation parameters are derived from the joint probability model (GMM) during the training phase. The improvement in performance accuracy is associated with a demand of larger amount of training data for estimating the model parameters in a higher dimensional space. The clean speech estimated during evaluation phase ($\hat{x}$), can be derived iteratively using MAP estimation or non-iteratively using the MMSE criterion. The scope of the present discussion is restricted to the MMSE version of the SSM for the ease of comparison with earlier methods. The details of the algorithm is described in the remaining part of this subsection.

As usual let's consider a pair of d dimensional clean and noisy feature vector x_t and y_t, respectively. A joint vector z_t of dimension $2d$ is constructed as $z_t = [y_t^T, x_t^T]^T$. The joint vectors are modeled using a GMM $\lambda^{(z)}$ as follows

$$p(z_t) = \sum_{j=1}^{M} w_z(j) \mathcal{N}(z_t; \mu_z(j), \Sigma_z(j)) \tag{4.29}$$

where

$$\mu_z(j) = \begin{bmatrix} \mu_y(j) \\ \mu_x(j) \end{bmatrix}, \Sigma_z(j) = \begin{bmatrix} \Sigma_{yy}(j) & \Sigma_{yx}(j) \\ \Sigma_{xy}(j) & \Sigma_{xx}(j) \end{bmatrix} \tag{4.30}$$

This model is similar to those defined in Eqs. (4.2) and (4.3). The mean vector $\mu_z(j)$ for component j is now a concatenation of individual mean vectors $\mu_y(j)$ and $\mu_x(j)$. The composition of the covariance matrix $\Sigma_z(j)$ can be similarly related. $\Sigma_{yy}(j)$ and $\Sigma_{xx}(j)$ are the covariance matrices for the jth component of the noisy and clean GMMs, respectively. Apart from these, $\Sigma_{yx}(j)$ and $\Sigma_{xy}(j)$ denote the cross-covariance matrices of y and x for the jth GMM component. The GMM is

trained with the standard EM algorithm using the joint vectors z. The training stage essentially comprises deriving the model parameters by partitioning the matrices $\mu_z(j)$ and $\Sigma_z(j)$ as shown above. During the evaluation stage, the partitioned parameters are used to formulate the conditional pdf $p(x_t|y_t)$ required for the MMSE-based prediction of $\hat{x}_t$ as defined in Eq. (4.1). Unlike previous methods, mathematical derivations show that without any approximations the conditional pdf is another GMM where the mixture weights are posterior probabilities of each Gaussian component with respect to y [14].

$$p(x_t|y_t, \lambda^{(z)}) = \sum_{j=1}^{M} p(j|y_t, \lambda^{(Z)}) p(x_t|y_t, j, \lambda^{(z)}) \tag{4.31}$$

where

$$p(j|y_t, \lambda^{(z)}) = \frac{w_y(j)\mathcal{N}(y_t; \mu_y(j), \Sigma_{yy}(j))}{\sum_{j=1}^{M} w_y(j)\mathcal{N}(y_t; \mu_y(j), \Sigma_{yy}(j))} \tag{4.32}$$

$$p(x_t|y_t, j, \lambda^{(Z)}) = \mathcal{N}(x_t; E_x(j,t), D_x(j)) \tag{4.33}$$

The mean vector $E_x(j,t)$ and covariance matrix $D_x(j)$ of the jth Gaussian in the conditional pdf are defined as

$$E_x(j,t) = \mu_x(j) + \Sigma_{xy}(j)\Sigma_{yy}(j)^{-1}(y_t - \mu_y(j)) \tag{4.34}$$

$$D_x(j) = \Sigma_{xx}(j) - \Sigma_{xy}(j)\Sigma_{yy}(j)^{-1}\Sigma_{yx}(j) \tag{4.35}$$

Given a noisy test vector y_t, its equivalent clean estimate $\widehat{x_t}$ can be then derived by the MMSE predictor as follows

$$\begin{aligned}
\widehat{x_t} &= E[x_t|y_t] \\
&= \int_X x_t\, p(x_t|y_t, \lambda^{(z)}) \mathrm{d}x_t \\
&= \int_X \sum_{j=1}^{M} x_t\, p(j|y_t, \lambda^{(Z)}) p(x_t|y_t, j, \lambda^{(z)}) \mathrm{d}x_t \\
&= \sum_{j=1}^{M} p(j|y_t, \lambda^{(Z)}) E_x(j,t)
\end{aligned} \tag{4.36}$$

The principle of SSM is similar to SPLICE except for the joint probability distribution of noisy and clean feature spaces. In fact SPLICE with MMSE predictor reduces to its SSM counterpart if the cross-correlation of clean and noisy data is taken into account. SSM bears close resemblance to other model-based non-linear

transformation methods like Constrained MLLR [15]. However the difference lies in the fact that the transformations in SSM are learned offline during the training phase while those in case of CMLLR, are done online during evaluation. A comparative study of SSM and other contemporary feature compensation methods can be found in [13].

4.3.2 Trajectory-Based Stochastic Mapping (TRAJMAP)

The MMSE estimator of SSM as discussed in Sect. 4.3.1 is a mixture of linear transforms weighted by the posterior probability of each GMM component. The parameters for the linear transform are derived from the joint distribution of both spaces. The approach is similar to any conventional GMM-based mapping techniques which has diverse applications [16]. However, a distinct drawback of such frame-wise mapping frameworks is that they fail to capture the correlation of features in the entire sequence. This results in inappropriate dynamic characteristics and an excessively smoothed spectra. The cepstral trajectory based GMM mapping (TRAJMAP) algorithm [17, 18] addressed this drawback by applying a Hidden Markov Model (HMM)-based parameter generation algorithm [19] with dynamic features, to the GMM-based mapping framework. Instead of individual frame-wise mapping, an entire sequence of frames (cepstral trajectory) is transformed in parallel. This approach had shown promising results for both noise-compensation [18] and voice conversion applications [17], in past. A fundamental assumption of the TRAJMAP algorithm is that despite noise corruption underlying spectral properties of a speaker remain preserved. The algorithm is used to learn a mapping function from a sequence of vectors in a speaker's noisy utterance to the corresponding sequence of clean vectors in the stereo training data. Mathematical details of the TRAJMAP transformation framework [17] is discussed in the remaining part of the section.

The cepstral vector trajectory is represented by a sequence of clean MFCC vectors $\mathbf{X}$ and noisy MFCC vectors $\mathbf{Y}$ where $\mathbf{X}$ and $\mathbf{Y}$ together constitute the stereo training data.

$$\mathbf{X} = [X_1^T, X_2^T, \ldots X_{\mathbf{T}}^T]^T \tag{4.37}$$

$$\mathbf{Y} = [Y_1^T, Y_2^T, \ldots Y_{\mathbf{T}}^T]^T \tag{4.38}$$

where $\mathbf{T}$ denotes the total number of vectors in the sequence. Individual vectors of each sequence are a concatenation of the static MFCC, its delta and acceleration coefficients. Each vector in the above sequence are $3d$ dimensional considering static MFCC vectors of d dimension,

$$X_t = [x_t^T, \Delta x_t^T, \Delta^2 x_t^T]^T \tag{4.39}$$

$$Y_t = [y_1^T, \Delta y_t^T, \Delta^2 y_t^T]^T \tag{4.40}$$

The GMM $\lambda^{(Z)}$ of the joint pdf $p(Z_t|\lambda^{(Z)})$ is trained by a concatenated pair of clean and noisy vector (Z_t) from the stereo training data where $Z_t = [Y_t^T, X_t^T]^T$. The aim to map the noisy MFCC trajectory $\mathbf{Y}$ to its clean counterpart $\mathbf{X}$. This is achieved by maximizing the following likelihood function

$$\begin{aligned} p(\mathbf{X}|\mathbf{Y}, \lambda^{(Z)}) &= \sum_{\mathbf{j}} p(\mathbf{j}|\mathbf{Y}, \lambda^{(Z)}) p(\mathbf{X}|\mathbf{Y}, \mathbf{j}, \lambda^{(Z)}) \\ &= \prod_{t=1}^{\mathbf{T}} \sum_{j=1}^{M} p(j|Y_t, \lambda^{(Z)}) p(X_t|Y_t, j, \lambda^{(Z)}) \end{aligned} \tag{4.41}$$

where $\mathbf{j} = \{j_1, j_2 \ldots j_{\mathbf{T}}\}$ is a mixture component sequence. The conditional pdf at each frame is modeled as a GMM. At frame t, the jth mixture component weight $p(j|Y_t, \lambda^{(Z)})$ and the jth conditional probability distribution $p(X_t|Y_t, j, \lambda^{(Z)})$ are given by the following expressions

$$p(j|Y_t, \lambda^{(Z)}) = \frac{w_j^Y \mathcal{N}(Y_t; \mu_j^Y, \Sigma_j^{YY})}{\sum_{j=1}^{M} w_j^Y \mathcal{N}(Y_t; \mu_j^Y, \Sigma_j^{YY})} \tag{4.42}$$

$$p(X_t|Y_t, j, \lambda^{(Z)}) = \mathcal{N}(X_t; E_{j,t}^X, D_j^X) \tag{4.43}$$

where

$$E_{j,t}^X = \mu_j^X + \Sigma_j^{XY}(\Sigma_j^{YY})^{-1}(Y_t - \mu_j^Y) \tag{4.44}$$

$$D_j^X = \Sigma_j^{XX} - \Sigma_j^{XY}(\Sigma_j^{YY})^{-1}\Sigma_j^{YX} \tag{4.45}$$

The notations for conditional mean and conditional covariance used in Eqs. (4.44) and (4.45) are similar to the ones discussed earlier in Sect. 4.3.1.

The task is to estimate a sequence of clean vectors $\hat{X}$ from the entire sequence of noisy feature vectors $\mathbf{Y}$. This is achieved in two stages. In the first stage, a HMM-based parameter generation algorithm [19] is used to convert $\mathbf{Y}$ to the static MFCC parameters $\hat{x}$. In the next stage, the delta and acceleration coefficients are derived from each static MFCC vector of $\hat{x}$ and concatenated with itself to obtain the resultant sequence $\hat{X}$. In contrast to the MMSE-based methods, the derivation of $\hat{x}$ is based on a maximum likelihood estimate (MLE) as follows

$$\hat{x} = \arg\max_{x} p(\mathbf{X}|\mathbf{Y}, \lambda^{(Z)}) \tag{4.46}$$

where $\hat{x} = [\hat{x}_1^T, \hat{x}_2^T, \ldots, \hat{x}_{\mathbf{T}}^T]$ is the sequence of estimated static feature vectors. A matrix $\mathbf{W}$ of dimension $3d\mathbf{T} \times d\mathbf{T}$ is defined such that it converts the static sequence $\hat{x}$ to the expanded sequence $\hat{X}$ as follows

$$\hat{X} = \mathbf{W}\hat{x} \tag{4.47}$$

where $\hat{X}$ is the sequence of denoised MFCC vectors with dynamic (delta and acceleration) co-efficients as defined in Eqs. (4.38) and (4.40). The composition of the matrix $\mathbf{W}$ is discussed as follows

$$\mathbf{W} = [W_1, W_2, \ldots W_t, \ldots W_{\mathbf{T}}]^T \otimes I_{DXD} \tag{4.48}$$

$$W_t = [\mathbf{w_t^{(0)}}, \mathbf{w_t^{(1)}}, \mathbf{w_t^{(2)}}] \quad t = 1, 2, \ldots \mathbf{T} \tag{4.49}$$

$$\mathbf{w_t^{(n)}} = [\overbrace{0, \ldots, 0}^{1\text{st}}, \overbrace{w^{(n)}(-L_-^{(n)}), \ldots,}^{(t-L_-^{(n)})th} \overbrace{w^{(n)}(L_+^{(n)})}^{(t+L_+^{(n)})th} \overbrace{, \ldots, w^{(n)}(0), \ldots,}^{(t)th} \overbrace{0}^{\mathbf{T}\text{-th}}]^T \quad n = 0, 1, 2 \tag{4.50}$$

In Eq. (4.48), each submatrix W_t is of size $\mathbf{T} \times 3$ and '$\otimes$' denotes the Kronecker product. In Eq. (4.50), $w^{(n)}(\tau)$ denotes the weights required for calculating the Δ^n MFCC coefficient for the $(t + \tau)$th time frame. τ varies in a frame span of $[-L_-^{(n)}, L_+^{(n)}]$ as shown in the following equations ($L_+^{(0)} = L_-^{(0)} = 0$ and $w^{(0)}(0) = 1$)

$$\Delta x_t = \sum_{\tau=-L_-^{(1)}}^{L_+^{(1)}} w^{(1)}(\tau) x_{t+\tau} \tag{4.51}$$

$$\Delta^2 x_t = \sum_{\tau=-L_-^{(2)}}^{L_+^{(2)}} w^{(2)}(\tau) x_{t+\tau} \tag{4.52}$$

The maximum likelihood estimate in Eq. (4.46) is solved by an EM algorithm which iteratively maximizes an auxillary function with respect to $\hat{x}$ as follows

$$Q(\mathbf{X}, \hat{\mathbf{X}}) = \sum_{\mathbf{j}} p(\mathbf{j}|\mathbf{Y}, \mathbf{X}, \lambda^{(Z)}) \log(p(\hat{\mathbf{X}}, \mathbf{j}|\mathbf{Y}, \lambda^{(Z)})) \tag{4.53}$$

The sequence of vector $\hat{x}$ obtained as a solution of Eq. (4.53) is given by

$$\hat{x} = (\mathbf{W}^T \overline{(\mathbf{D^X})^{-1}} \mathbf{W})^{-1} \mathbf{W}^T \overline{(\mathbf{D^X})^{-1} \mathbf{E^X}} \tag{4.54}$$

where

$$\overline{(\mathbf{D^X})^{-1}} = diag[\overline{(D_1^X)^{-1}}, \overline{(D_2^X)^{-1}}, \ldots, \overline{(D_t^X)^{-1}}, \ldots, \overline{(D_{\mathbf{T}}^X)^{-1}}] \tag{4.55}$$

$$\overline{(\mathbf{D^X})^{-1} \mathbf{E^X}} = [\overline{(D_1^X)^{-1} E_1^{X^T}}, \overline{(D_2^X)^{-1} E_2^{X^T}}, \ldots, \overline{(D_t^X)^{-1} E_t^{X^T}}, \ldots, \overline{(D_{\mathbf{T}}^X)^{-1} E_{\mathbf{T}}^{X^T}}]^T \tag{4.56}$$

$\overline{(\mathbf{D^X})^{-1}}$ in Eq. (4.55) is a block diagonal matrix of size $3d\mathbf{T} \times 3d\mathbf{T}$ while $\overline{(\mathbf{D^X})^{-1}\mathbf{E^X}}$ in Eq. (4.56) is a vector of size $3d\mathbf{T} \times 1$. The individual constituents of the matrices i.e., $\overline{(D_t^X)^{-1}}$ and $\overline{(D_t^X)^{-1}E_t^X}$ are given by

$$\overline{(D_t^X)^{-1}} = \sum_{j=1}^{M} \lambda_{j,t}(D_j^X)^{-1} \tag{4.57}$$

$$\overline{(D_t^X)^{-1}E_t^X} = \sum_{j=1}^{M} \lambda_{j,t}(D_j^X)^{-1}E_{j,t}^X \tag{4.58}$$

$$\lambda_{j,t} = p(j|Y_t, X_t, \lambda^{(Z)}) \tag{4.59}$$

Detailed derivation of Eq. (4.54) is provided in Appendix A. The solution $\hat{x}$ is only a sequence of static MFCC vectors i.e., a vector of size $d\mathbf{T} \times 1$. The full sequence with delta and acceleration coefficients appended with the resultant vector can be obtained by a simple linear operation $\mathbf{W}\hat{x}$.

4.4 Development of Proposed SV Systems

All experiments are carried out in the NIST-2003-SRE database [20] introduced in Chap. 3. The data consists of single training utterances of approximately 2 min length from each of 356 enrolled speakers and 3,500 test utterances (approximately 10–15 s each) for evaluation. The purpose of present work is to address the issue of speaker verification in mismatched condition where a speaker enrolls in a clean environment whereas during verification his/her speech is corrupted by background noise. However stereo-data based techniques as described in Sect. 4.1 require simultaneous recording of a speaker's training data over two channels i.e., one in clean condition and the other in a noisy environment. Due to unavailability of such data, the noisy utterances used in the present work were simulated by corrupting the clean speech utterances of the NIST-SRE-2003 by different types of additive noises. The approach is motivated by synthetic generation of stereo-data as described in [21]. The standard GMM-UBM framework was used for speaker verification [22]. Figure 4.3 shows the block diagram of the feature compensation process in a GMM-UBM based speaker verification system. The various stages of the SV system development are discussed in the following sections.

4.4.1 Simulation of Background Environment

Four additive noises (i.e., car, factory, pink and white) collected from the NOISEX-92 database [23] were used for representing unique background environments. The speech segment from each of the 356 enrolled speakers was degraded by adding a

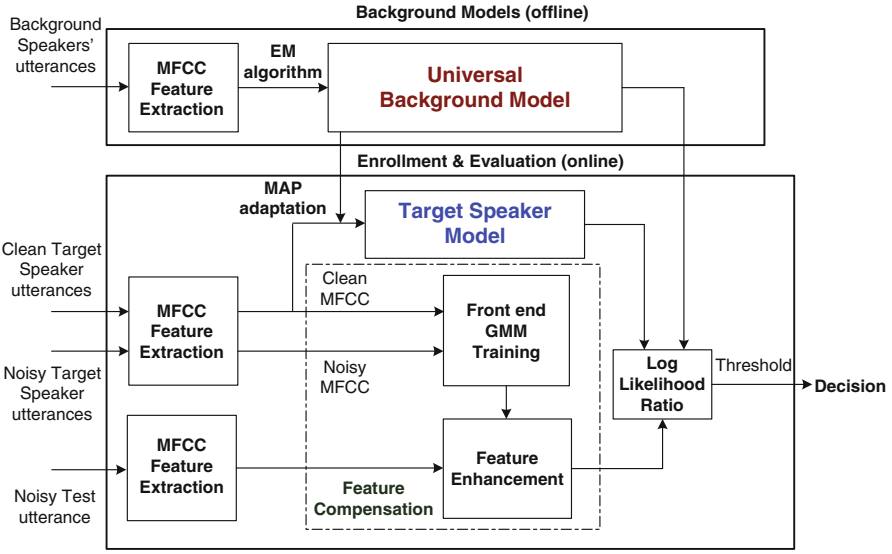

Fig. 4.3 Block diagram of the feature compensation process in the speaker verification system

specific type of noise at 0 and 5 dB SNRs, respectively. The noise level was scaled to maintain the desired SNRs of the reconstructed speech segments. Eight different sets of noisy training utterances were obtained (one for each noise at a particular SNR). The default training set of the NIST-SRE-2003 was used as the clean recordings.

All test utterances were similarly reconstructed by noise addition at the two SNRs. Each set of noisy utterances were used for the following sets of experiments.

1. Mismatched Condition: The noisy test utterances were evaluated against speaker models built from clean training data.
2. Matched Condition: The noisy test utterances were evaluated against speaker models built from noisy training data.
3. Feature compensated: The noisy test utterances were subjected to feature enhancement prior to evaluation against clean speaker models. Each of the four feature compensation techniques discussed in Sect. 4.1 were compared with the above two conditions and the proposed method, on the basis of their performance.

The simulated stereo training data was used for front-end GMM training as discussed later in Sect. 4.4.3. For comparing relative improvements in performance accuracy produced by the various feature compensation schemes, the SV systems under mismatched conditions have been considered as a baseline.

4.4.2 Feature Extraction and Speaker Modeling

The feature extraction and speaker modeling process are identical to that used in the GMM-UBM framework described in Chap. 3. Standard MFCC coefficients were used as features. After pre-emphasis and an energy-based voiced activity detection, 39-dimensional feature vectors (consisting of 13 MFCCs $+ \Delta + \Delta\Delta$ excluding C_0) derived from a 26 channel mel-scaled filterbank, were extracted from speech frames of 20 ms with a frame-overlap of 10 ms. All feature vectors were subjected to cepstral mean subtraction followed by cepstral variance normalization. The resultant distribution was scaled to zero mean and unit variance. In the remaining part of the chapter, the MFCC feature vectors extracted from the noisy training data and its clean counterpart are referred as 'noisy vectors' and 'clean vectors', respectively.

Acoustic modeling using the standard GMM-UBM framework was performed in two stages i.e., construction of a Universal Background Model (UBM) and the target speaker models. Twenty hours of speech collected from the SwitchBoard II corpus was used to construct a 1,024-component GMM offline using 200 iterations of the EM algorithm. The target speaker models (GMMs) were derived by MAP adaptation of the UBM using each enrolled speaker's training data. The process was repeated twice, once each for the clean and noise-degraded speech of the stereo training data. The clean speaker models were used for evaluation in the mismatched condition as well as the feature compensated conditions.

4.4.3 Feature Compensation

The two basic stages of the feature compensation process are discussed below.

- **Front-end GMM Training**: The stereo training data corresponding to each speaker was used for building speaker-specific front-end GMMs prior to feature enhancement. For RATZ, SPLICE and MMCN, a pair of 8-component GMMs (clean and noisy) with diagonal covariance matrices were constructed for each speaker using the standard EM algorithm.

  For SSM and TRAJMAP, individual pairs of noisy and clean MFCC vectors in the aligned sequence were first concatenated to create a single sequence of 78-dimensional MFCC vectors. The joint vectors were used to build speaker specific 8-component GMMs with full covariance matrices. The number of components for the GMMs were empirically determined according to the available training data. However in practical applications without training data constraints, higher number of components can be explored. The conditional GMM parameters required for SSM and TRAJMAP were derived using Eqs. (4.34), (4.35) and (4.44), (4.45), respectively.
- **Feature Enhancement**: Each noisy test feature vector was transformed using the front-end GMM parameters of each of the 11 target speaker models specified

for the evaluation phase of the NIST-2003 primary task. In contrast to the actual evaluation process, each of the 11 transformed vectors were scored against the corresponding speaker model and the UBM.

The corrective bias vectors of the mean and covariance terms for RATZ were estimated using Eqs. (4.8) and (4.9), non-iteratively. This was followed by the MMSE predicted value given by Eq. (4.10). Only the noisy front-end GMMs trained as in Eq. (4.3) were used to estimate the bias vectors for SPLICE as given by Eqs. (4.13) and (4.14). This was followed by the MMSE estimate given by Eq. (4.16). The simplified single environment version of MEMLIN i.e., MMCN was used for feature enhancement. The posterior probability factor for each environment given by Eq. (4.22) could thus be entirely omitted. The cross probability model (Eq. 4.28) and the MMSE predictor (Eq. 4.27) were likewise simplified. MMSE estimates for SSM and the MLE estimate for TRAJMAP were calculated using Eqs. (4.36) and (4.54), respectively. The static MFCCs obtained from TRAJMAP were concatenated with the delta and acceleration coefficients to yield the resultant 39-dimensional vector.

4.4.4 Effect of Feature Compensation in Cepstral Domain

Effectiveness of the stochastic feature compensation methods is demonstrated by a set of plots which highlight some characteristics of the transformed and distorted MFCC features. Since the lower order MFCC coefficients represent the broad spectral shape, the first MFCC coefficient has been considered without loss of generality for demonstrating the impact of feature compensation. Figure 4.4a shows the histogram of the first MFCC coefficients of an arbitrary test speech utterance from the NIST-2003-SRE and its equivalent noisy signal obtained by white noise addition at 0 dB SNR. Figure 4.4b–f shows the effect of enhancing the noisy utterance by applying various feature compensation algorithms. Since the feature vectors were mean and variance normalized, both the distributions are centered around zero. However the area under the overlapping region of the curves is a measure of accuracy in the conversion process. A fully overlapped curve would suggest the ideal situation of perfect conversion. The distortion caused by noise statistics can be observed in Fig. 4.4a in which the peak of noisy distribution is significantly skewed towards the left. The skewness shows a gradual reduction after the application of feature compensation algorithms. The shape of the transformed (compensated) distributions is similarly affected by noise addition. The simple noisy distribution shows arbitrary changes in the spectral shape as seen in several regions of the curve. The change in spectral shape is negligible in case of RATZ with minor differences at the peak region. The change in the noisy distribution shows more prominence in case of SPLICE and MMCN. A spectral smoothening effect can be observed at the peak regions for the SPLICE and MMCN-compensated distributions, respectively with slightly more overlap in case of the former. The SSM and TRAJMAP compensated

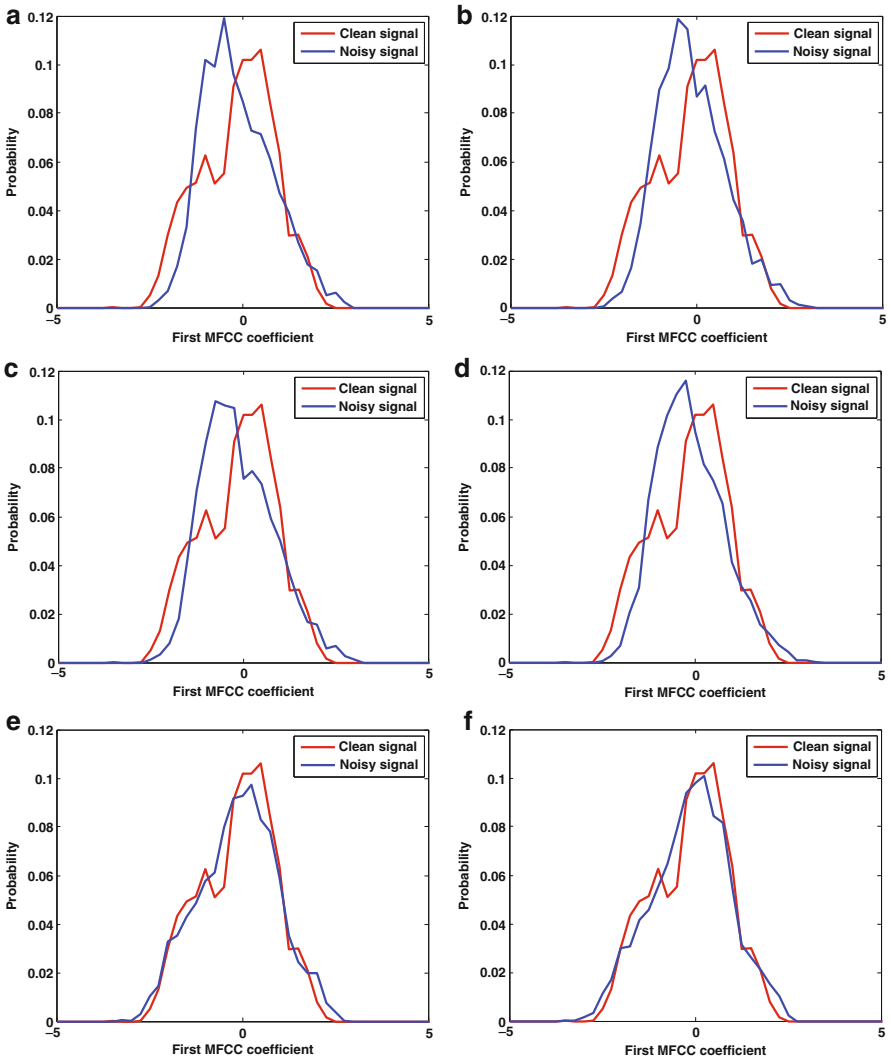

Fig. 4.4 Histograms of the first MFCC coefficient of a clean test speech signal (*red*) and the same signal contaminated with white noise at 0 dB (*blue*) (**a**) without feature compensation and with feature compensation using (**b**) RATZ (**c**) SPLICE (**d**) MMCN (**e**) SSM and (**f**) TRAJMAP

distributions shows comparatively higher resemblances with the clean distribution. The significant increase in the overlapping area of the histograms is apparent. The changes are also reflected on the spectral shape which shows that the transformed distribution captures minute similarities at the peak region.

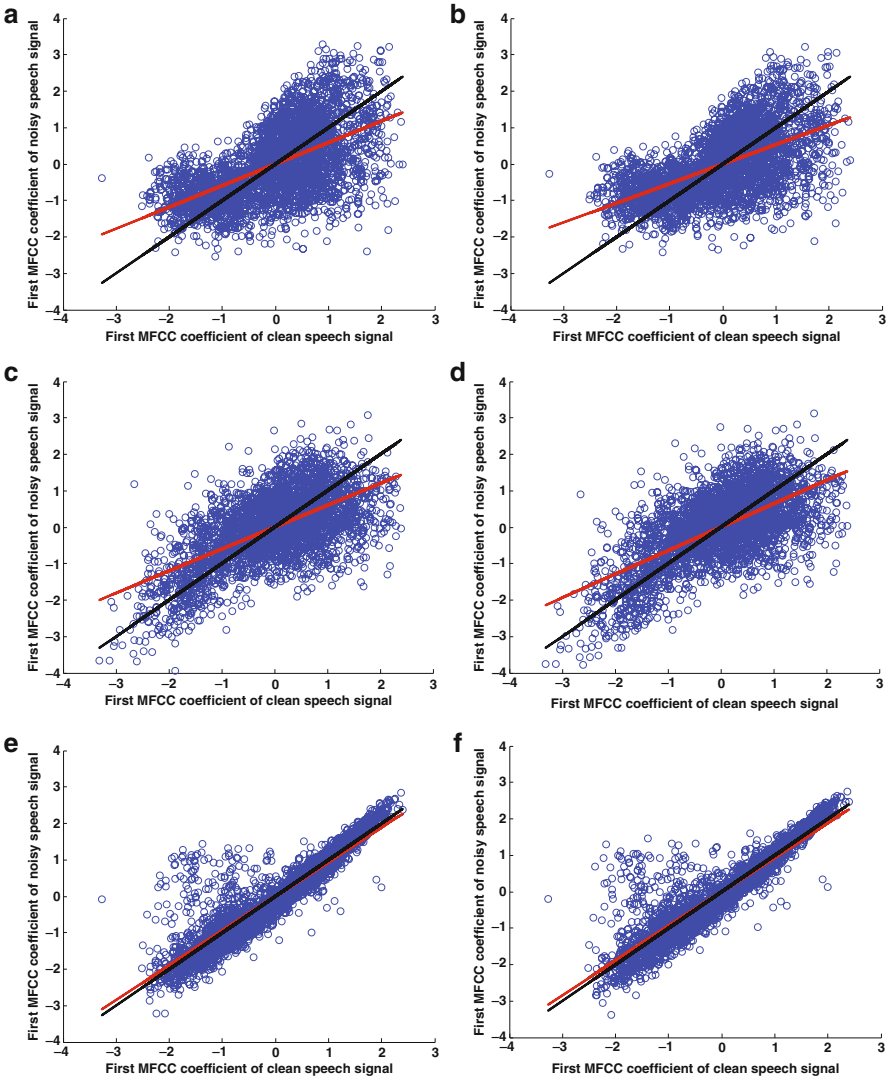

Fig. 4.5 Scatterplots between the first MFCC coefficient of non-silence frames of a clean test speech signal (x-axis) and the same signal contaminated with white noise at 0 dB (y-axis) (**a**) without feature compensation and with feature compensation using (**b**) RATZ (**c**) SPLICE (**d**) MMCN (**e**) SSM and (**f**) TRAJMAP. The (x = y) line is colored *black* while the line of best fit is colored *red*

Figure 4.5a–f shows the scatter plots between the first MFCC coefficients (C_1) of the given test utterance (x-axis) and its white noise corrupted equivalent (y-axis). The C_1s extracted from non-silence frames of the test utterance are represented by blue circles. The black line represents the ideal condition of perfect feature

transformation (x = y). The red line is a first order polynomial of the clean feature vectors which best fits the noisy feature vectors in a least square sense. The imperfections in the transformation process can be inferred from the deviation between the two lines in a figure. The distortion of the cepstral distribution due to the addition of white noise is apparent from Fig. 4.5a in which the two lines are significantly deviated from each other. The spread of the data (blue dots) across the black line is a measure of the covariance of the clean and noisy data. Significant changes in the scatter plots can be observed after the application of the feature compensation algorithms. The increased covariance of data is noticed in case of SPLICE and MMCN where the deviation between the red and black lines is relatively lower compared to RATZ. SSM and TRAJMAP shows the best fit in terms of covariance of the given data with the latter showing marginal improvements over the former. Despite outliers most of the data points are considerably aligned along the line of best fit with very little noticeable deviation.

4.4.5 Performance Evaluation

All experiments were performed in mismatched, matched and compensated conditions each of which has been discussed in Sect. 4.4.1. The NIST-2003 primary task was carried out in which each noisy test utterance was evaluated against 11 target speaker models (GMMs). The equal error rate (EER) and minimum DCF (MinDCF) values were used as metrics for performance evaluation.

Figures 4.6 and 4.7 show the DET curves of the SV systems in various conditions, with background noise at 0 and 5 dB SNRs, respectively. The summary of the performance of SV systems in different noisy background is shown in Table 4.1. A quick observation reveals a general order of precedence of the SV performance accuracy in terms of EER values i.e., mismatched, RATZ, matched, MMCN, SPLICE, SSM and TRAJMAP. The pattern is also valid for the MinDCF values except for the fact that they often do not show a monotonic decrement across the various methods. The only exception to this order is seen in case of car noise at 0 and 5 dB SNRs. The mismatched condition expectedly shows the worst case scenario in every noisy environments with an average EER of 29.93% across all of them for both SNRs. This is in conformity with the known fact that noise degradation causes arbitrary changes in the clean feature distributions due to which noisy test utterances yield poor scores during the pattern matching stage. The performance of the RATZ compensation scheme shows minor improvement over the baseline (mismatch) with an average decrement of 3.07 % EER. Interestingly, the matched condition in most cases outperform RATZ. A possible explanation to this behavior is the invalidity of the posterior invariance assumption in low SNR conditions as discussed in Sect. 4.2.1. The effect of feature normalization using posterior probability of the noisy MFCC vectors with respect to clean Gaussian components has other interesting implications. As discussed in Sect. 4.2.3, the MEMLIN algorithm (multienvironment version of MMCN), uses both noisy and clean GMMs as inputs,

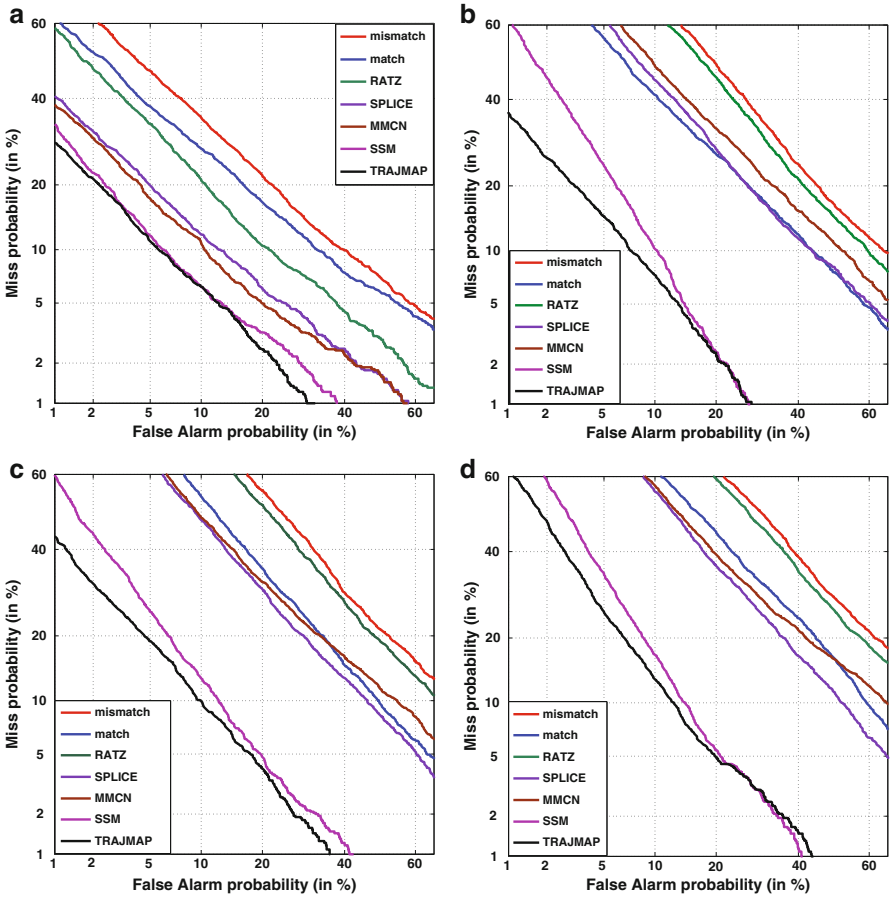

Fig. 4.6 DET plots for the SV systems developed using the features derived from SFC methods in uniform background environment containing (**a**) car noise (**b**) factory noise (**c**) pink noise and (**d**) white noise at 0 dB SNR

thus incorporating both types of posterior probabilities in the final transformation. However, contrary to known facts, the SPLICE algorithm performs moderately better than the MMCN algorithm with an average improvement of 1.62 % EER for factory, pink and white background environments. The improvement is consistent in case of MinDCF values and more pronounced in case of factory, pink and white noises. There are two possible justifications to this phenomenon. Firstly, the inclusion of the inaccurate clean Gaussian posteriors in estimating the corrective vectors and secondly, an oversimplified cross probability model which excludes the environment factor '*e*' from the final transformations, as discussed in Sect. 4.2.3. It is interesting to note that this effect is in conformity with the anomalous behavior of the SV performances observed in case of car noise. Unlike the other background

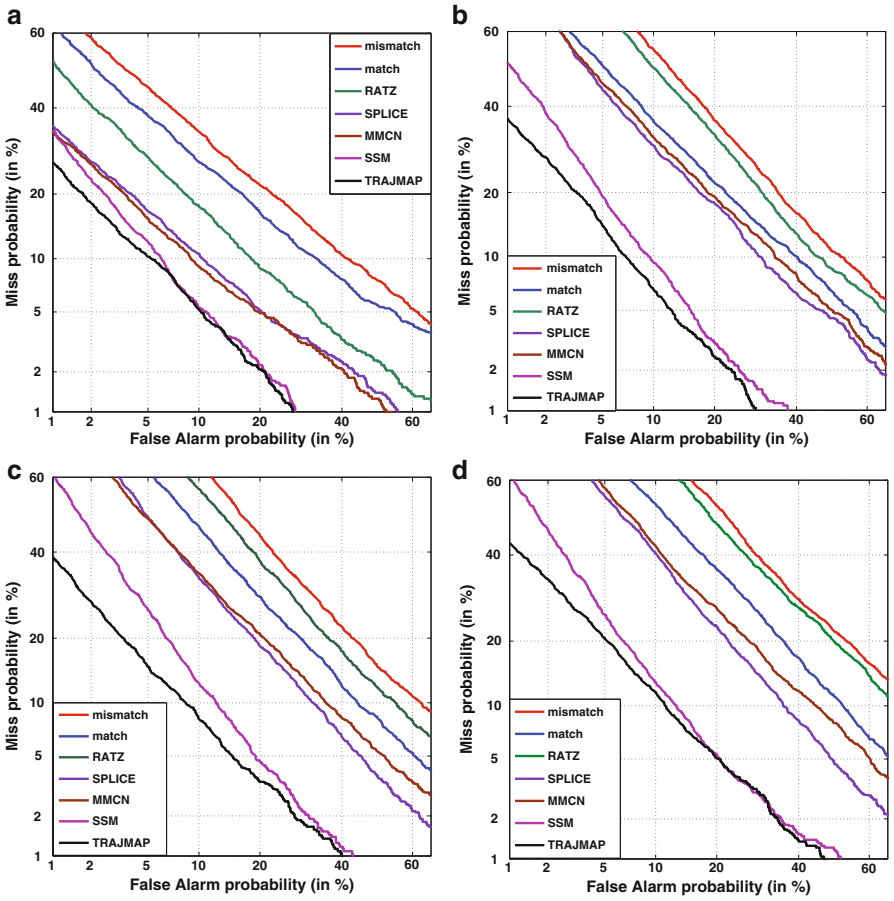

Fig. 4.7 DET plots for the SV systems developed using the features derived from SFC methods in uniform background environment containing (**a**) car noise (**b**) factory noise (**c**) pink noise and (**d**) white noise at 5 dB SNR

noises, in case of car environment, it is observed that the performance in mismatched condition is only slightly worse than that of the matched one with an average drop of 2.67 % EER across both SNRs. In this case, the positive effect of clean Gaussian components in normalization, is also reflected by the considerable improved SV performances of RATZ and MMCN in comparison to the matched condition and SPLICE, respectively.

The SSM and TRAJMAP shows a significant improvement in performance compared to the other algorithms with a large margin of difference in terms of EER and MinDCF. In comparison to SPLICE, an average EER reduction as high as 9.37 and 10.32 % is obtained for SSM and TRAJMAP, respectively. The improvement is consistent even in the case of the anomalous car noise in which TRAJMAP is seen

Table 4.1 Summary of performance of the SV systems developed using the features derived from SFC methods in uniform background environments

		Car		Factory		Pink		White	
SNR (dB)	Methods	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF
0	Mismatch	20.55	0.079	32.16	0.099	35.05	0.099	39.02	0.099
	Matched	18.04	0.071	23.17	0.097	26.65	0.092	30.98	0.097
	RATZ	14.32	0.066	30.44	0.099	33.42	0.098	37.62	0.099
	SPLICE	10.75	0.049	23.26	0.085	23.87	0.096	27.32	0.098
	MMCN	9.94	0.048	26.02	0.089	25.47	0.095	28.91	0.096
	SSM	7.59	0.041	9.49	0.058	11.02	0.063	12.19	0.078
	TRAJMAP	7.45	0.039	8.45	0.045	9.71	0.051	11.07	0.066
5	Mismatch	20.95	0.077	27.15	0.098	30.39	0.097	34.16	0.099
	Matched	18.11	0.071	20.96	0.085	23.89	0.092	27.41	0.094
	RATZ	13.41	0.059	25.20	0.096	27.64	0.097	32.77	0.099
	SPLICE	10.02	0.044	18.47	0.054	19.11	0.087	21.14	0.092
	MMCN	9.26	0.043	19.24	0.063	20.14	0.086	23.08	0.091
	SSM	7.10	0.037	9.47	0.057	10.98	0.065	11.11	0.066
	TRAJMAP	7.09	0.036	8.04	0.045	9.08	0.046	10.48	0.052

Table 4.2 Relative equal error rates of the proposed SV systems developed using the features derived from SFC methods

Feature compensation methods	Relative equal error rate (EER_R) %							
	0 dB SNR				5 dB SNR			
	Car	Factory	Pink	White	Car	Factory	Pink	White
RATZ	30.32	5.35	4.66	3.58	35.99	7.18	9.05	4.07
SPLICE	47.69	26.67	31.90	29.98	52.17	31.97	37.11	38.11
MMCN	51.63	19.09	27.33	25.91	55.80	29.13	33.73	32.44
SSM	63.07	70.05	68.55	68.76	66.11	65.12	63.87	67.48
TRAJMAP	63.74	77.73	72.30	71.63	66.16	70.39	70.12	69.32

to perform moderately better than the SSM algorithm. The MinDCF values which varied in the range of 0.099–0.085 are reduced to the range 0.045–0.078. Compared to SSM, an EER drop as high as 1.9% is noticed in case of pink noise at 5 dB SNR, while the other cases closely follow by with reductions of 1.43% for factory noise at 5 dB, 1.31% for pink noise at 0 dB and 1.12% for white noise at 0 dB, respectively. The EER variance for TRAJMAP from 0 to 5 dB SNRs is much lower than the rest of the compared cases. This is an indication of the suitability of its application for SV tasks which are robust to SNR changes.

In order to demonstrate the performance improvement of the feature compensated SV framework over the baseline SV system in terms of EER, the 'Relative Equal Error Rate' (EER_R) given by $EER_R = \frac{(EER_B - EER_V)}{EER_B} \times 100\,\%$ is calculated where EER_B and EER_V are the equal error rates for the baseline and proposed SV systems, respectively. Table 4.2 shows the relative EER values of the proposed SV systems for different background environments.

The overall performance improvement gained by the use of feature compensation algorithms is apparent. An average relative EER of 12.52 % for RATZ, 37.07 % for SPLICE, 34.39 % for MMCN, 66.68 % for SSM and 69.67 % for TRAJMAP across all noisy environments is obtained.

4.5 Summary

In this chapter we demonstrated the significance of stochastic feature compensation methods for robust speaker verification in noisy environment. The effectiveness of the these data-driven methods was demonstrated for speaker verification in different simulated noisy environments. Recent state-of-the-art algorithms based on joint GMM modeling of clean and noisy data (i.e., SSM, TRAJMAP) were found to outperform well known algorithms like SPLICE and MMCN in terms of EER and minDCF metrics of speaker verification. The overall best performance was observed in case of the TRAJMAP method, which thereby suggests significance of dynamic feature correlation and robustness of long-term utterances towards background noise. Synthetic noisy data and clean utterances were used instead of actual stereo

data in all experiments. For a better evaluation of the proposed method, actual stereo data may be used in future work. Data from real life environments at various other SNRs may be used instead of artificially constructed noisy data for a better insight into the efficiency of the proposed method.

References

1. T. Kinnunen, Spectral features for automatic text-independent speaker recognition. PhD thesis, Department of Computer Science, University of Joensuu, 2004
2. D.A. Reynolds, Experimental evaluation of features for robust speaker identification. IEEE Trans. Speech Audio Process. **2**(4), 639–643 (1994)
3. R. Mammone, X. Zhang, R. Ramachandran, Robust speaker recognition: a feature-based approach. IEEE Signal Process. Mag. **13**(5), 58–71 (1996)
4. S. Furui, Cepstral analysis technique for automatic speaker verification. IEEE Trans. Acoust. Speech Signal Process. **29**(2), 254–272 (1981)
5. H. Hermansky, N. Morgan, RASTA processing of speech. IEEE Trans. Speech Audio Process. **2**(4), 578–589 (1994)
6. S. Boll, Suppression of acoustic noise in speech using spectral subtraction. IEEE Trans. Acoust. Speech Signal Process. **27**(2), 113–120 (1979)
7. A. Acero, R.M. Stern, Environmental robustness in automatic speech recognition, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '90)*, Albuquerque, 1990, vol. 2, pp. 849–852
8. S. Sarkar, K.S. Rao, Stochastic feature compensation methods for speaker verification in noisy environments. Appl. Soft Comput. **19**, 198–214 (2014). Elsevier
9. P.J. Moreno, B. Raj, R.M. Stern, Data-driven environmental compensation for speech recognition: a unified approach. Speech Commun. **24**(4), 267–285 (1998)
10. L. Deng, A. Acero, L. Jiang, J. Droppo, X. Huang, High-performance robust speech recognition using stereo training data, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Salt Lake City, 2001, vol. 1, pp. 301–304
11. M.J.F. Gales, P.C. Woodland, Mean and variance adaptation within the MLLR framework. Comput. Speech Lang. **10**, 249–264 (1996)
12. L. Buera, E. Lleida, A. Miguel, A. Ortega, Multi-environment models based linear normalization for speech recognition in car conditions, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '04)*, Montreal, 2004
13. M. Afify, X. Cui, Y. Gao, Stereo-based stochastic mapping for robust speech recognition. IEEE Trans. Audio Speech Lang. Process. **17**(7), 1325–1334 (2009)
14. C.M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006)
15. V. Digalakis, D. Rtischev, L. Neumeyer, E. Sa, Speaker adaptation using constrained estimation of Gaussian mixtures. IEEE Trans. Speech Audio Process. **3**(5), 357–366 (1995)
16. Y. Stylianou, O. Cappe, E. Moulines, Continuous probabilistic transform for voice conversion. IEEE Trans. Speech Audio Process. **6**(2), 131–142 (1998)
17. T. Toda, A.W. Black, K. Tokuda, Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. IEEE Trans. Audio Speech Lang. Process. **15**(8), 2222–2235 (2007)
18. H. Zen, Y. Nankaku, K. Tokuda, Stereo-based stochastic noise compensation based on trajectory GMMs, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '09)*, Taipei, 2009

19. K. Tokuda, T. Masuko, T. Yamada, T. Kobayashi, S. Imai, An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features, in *Proceedings of the European Conference of Speech Communication Technology (EUROSPEECH '95)*, Madrid, Sept 1995, pp. 757–760
20. NIST-speaker recognition evaluations (1995) http://www.itl.nist.gov/iad/mig/tests/spk/
21. H. Hirsch, D. Pearce, The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions, in *Proceedings of the International Conference of Spoken Language Processing (ICSLP '00)*, Beijing, 2000
22. D. Reynolds, T. Quatieri, R. Dunn, Speaker verification using adapted Gaussian mixture models. Digit. Signal Process. **10**(1), 19–41 (2000)
23. A. Varga, H.J. Steeneken, Assessment for automatic speech recognition: II. NOISEX-92: a database and an experiment to study the effect of additive noise on speech recognition systems. Speech Commun. **12**, 247–251 (1993)