The following publication Y. Jiang and F. H. F. Leung, "Vector-Based Feature Representations for Speech Signals: From Supervector to Latent Vector," in IEEE Transactions on Multimedia, vol. 23, pp. 2641-2655, 2021 is available at https://doi.org/10.1109/TMM.2020.3014559.

Vector-based Feature Representations for Speech Signals: From Supervector to Latent Vector

Yuechi Jiang, *Student Member*, *IEEE*, and Frank H. F. Leung, *Senior Member*, *IEEE*

***Abstract*—There are two basic types of feature representations for speech signals. The first type refers to probabilistic models, such as the Gaussian mixture model (GMM). The second type refers to vector-based feature representations, such as the Gaussian supervector (GSV). Since vector-based feature representations are easier to use and process, they are more popular than probabilistic model-based feature representations. In this paper, we begin by explaining the rationale behind two widely used vector-based feature representations, viz. GSV and the i-vector, and then make extensions. GSV is a supervector (SV) based on maximum a posteriori (MAP) adaptation. Its computation is simple and fast, but its dimensionality is high and fixed. While the i-vector is a latent vector (LV) based on factor analysis (FA). Although the computation can be time-consuming because of additional model parameters, its dimensionality is changeable. To generalize GSV, we propose the MAP SV, which is also based on MAP adaptation but can have an even higher dimensionality and thus carry more information. To boost the computational efficiency of the i-vector, we adopt the concept of the mixture of factor analyzers (MFA) and propose the MFA LV, which exhibits a similar flexibility in dimensionality but is faster in computation. The experimental results for speaker identification and verification tasks demonstrate that, MAP SV can be more robust than GSV, and MFALV is comparable to or even better than the i-vector in effectiveness and meanwhile maintains a higher computational efficiency. With a powerful backend, GSV and MAP SV are comparable to the i-vector and MFALV, but the latter two are more flexible in dimensionality.**

***Index Terms*—Acoustic and speech signal processing, vector-based feature representation, Gaussian supervector, i-vector, supervector and latent vector**

I. Introduction

A piece of acoustic signal may carry different types of information. For example, a speech signal may carry information about the speaker, the acquisition device, or the surrounding environment. A nonspeech acoustic signal may carry information regarding acoustic scenes or environmental sounds. In order to further process the acoustic signal for detection, recognition or visualization purposes, effective and informative feature representations should be constructed. As the length of a piece of acoustic signal may vary, it is common to first divide the signal into equal-length short-time frames to obtain a sequence of frame-level feature vectors, and then obtain a sample-level feature representation based on them.

The sample-level feature representation can be a probabilistic model, such as the adapted Gaussian mixture model (GMM) [1], which works well for speaker recognition [1][2] and acoustic scene classification [3]. Intrinsically, it is difficult to compare the similarity of two probabilistic models in terms of distance. A feasible distance metric for two probabilistic models is the Kullback-Leibler (KL) divergence; however, it is not symmetric, and its calculation involves an integration over the whole feature space. This makes KL divergence inconvenient, as a closed-form solution only exists for simple models. Nonetheless, approximations can be made with certain constraints such that the computation is tractable [4].

Naturally, we prefer to use a vector to represent an acoustic sample, which means that the comparison of acoustic samples becomes the comparison of vectors. In addition, vector-based feature representations are also easy to visualize. Two prevalent vector-based feature representations are the Gaussian supervector (GSV) and the i-vector [5]. The applications of GSV include speaker recognition [4], speech acquisition device identification [6][7], speech clustering [8] and acoustic scene classification [9]. The i-vector has been widely adopted for speaker recognition [10][11][5] and is also applicable to voice search [12], acoustic scene classification [13][14] and acoustic signal clustering [15]. Due to the generality of GSV and the i-vector, their applications also extend to action recognition [16], facial expression recognition [17], and video indexing [18].

Both GSV and the i-vector are obtained based on the posterior probabilities provided by a universal background model (UBM), which is usually a GMM trained using unlabeled data. Nonetheless, the computation of GSV only depends on the parameters of the GMM, while that of the i-vector requires more parameters. From this perspective, GSV can be more efficient in computation than the i-vector, but the latter can have a flexible dimensionality. UBM provides some prior information regarding the distributions of the acoustic features [19], which is highly useful, especially when the number of labeled data is limited. The posterior probabilities can also be provided by a deep belief net (DBN) trained in an unsupervised manner [20] or a deep neural network (DNN) trained in a supervised manner [21]. The activations of the hidden layer in a DNN can also be used to produce vector-based

The work described in this paper was substantially supported by a grant from The Hong Kong Polytechnic University (Project Account Code: RUG7).

Yuechi Jiang and Frank H. F. Leung are with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. (e-mail: yuechi.jiang@connect.polyu.hk, frank-h-f.leung@polyu.edu.hk).

feature representations, such as the d-vector [22] and the x-vector [23][24]. However, performance improvement usually requires abundant training data or data augmentation [24].

GSV is based on maximum a posteriori (MAP) adaptation of the UBM. After adapting the parameters of the GMM-based UBM using the frame-level feature vectors obtained from an acoustic sample, the adapted mean vectors are concatenated to form the GSV. Naturally, one may wonder whether we can concatenate other parameters of the adapted GMM to obtain a more informative feature representation. This results in the proposed MAP supervector (SV), which may involve the concatenation of different MAP adapted parameters. The concatenation mechanism also makes the choice of the UBM flexible. In addition, we also propose a simple means of determining the value of the relevance factor used to construct GSV or MAP SV.

The i-vector is based on a factor analysis (FA) model, where it is presumed that an acoustic sample is generated by a latent vector. It is then obtained as the expected value of that latent vector. In the model assumption, the latent vector can be treated as being shared among all the frame-level feature vectors in an acoustic sample, as all of them together produce it. This sharing mechanism makes the i-vector robust and able to capture the common characteristics among all the frame-level feature vectors in an acoustic sample. However, this mechanism also makes the computation inefficient when the dimensionality of the latent vector is high. Thus, some studies try to simplify its formulation to reduce the computational burden [25][26]. The e-vector is a variant of the i-vector, differing in the estimation of the factor-loading matrix [42]. It can be more discriminative than the i-vector but still suffers from the high computational burden.

In this paper, from a different angle, we employ the mixture of factor analyzers (MFA) as the UBM to construct feature representations, resulting in the MFA supervector (MFASV) and MFA latent vector (MFALV). Different from the i-vector, MFA adopts the assumption that each frame-level feature vector has its own latent vector, whose dimensionality is then considerably lower than that of the i-vector. As a consequence, the estimation of model parameters for MFA is more efficient than the i-vector. MFALV is then obtained as the weighted sum of the expectations of the latent vectors, and MFASV is an affine transformation of MFALV. The rationale of MFA is that it models the frame-level feature vectors better if they carry abundant information and exhibit large variation across frames. In [44], MFA has also been used as a probability estimator for speaker identification and demonstrated some improvements over GMM. In this paper, we employ MFA to construct vector-based feature representations. In [45], the authors report that there may be redundancies within the frame-level feature vector; therefore, they employ the mixture of probabilistic principal component analysis (MPPCA) to model the distribution of the frame-level feature vectors, which is similar to employing MFA, as MPPCA is a special form of MFA with some constraints on the noise covariance. However, the MPPCA-based frame-level feature vectors are then fed into the i-vector framework, which also suffers from a high computational burden.

The performance of GSV, the i-vector and their extensions (viz. MAP SV, MFALV and MFASV) are evaluated by performing two small-scale speaker identification tasks and one large-scale speaker verification task. Linear support vector machine (SVM) and the probabilistic linear discriminant analysis (PLDA) model are employed as the classifiers. PLDA is also used to produce the verification score. SVM is an efficient classifier with a broad range of applications, while PLDA is a powerful model for certain specific research areas such as face recognition [27]-[29], speaker recognition [21]-[26], and audio-visual studies [30][31].

The major contributions of this paper include the following points:

- We analyze and discuss the rationale of GSV and the i-vector and perform comprehensive comparative experiments to evaluate their effectiveness and efficiency.
- Starting from GSV, we propose MAP SV, which generalizes GSV. MAP SV demonstrates the generality of the concept of the Supervector and can even be further extended.
- Starting from the i-vector, we propose MFALV, which can consume less time to compute and less memory space to store. Both theoretical analysis and experimental results show that, compared to the i-vector, MFALV has higher computational efficiency and similar effectiveness.
- We compare different feature representations from both theoretical and experimental perspectives, attempting to provide some inspirations for designing new types of feature representations.

The rest of this paper is organized as follows. In Section II, we describe the formulation of GSV and explain the rationale behind it. We then extend it to MAP SV, which may carry more abundant information. The concept of MAP SV can even be further extended. In Section III, we describe the formulation of the i-vector and explain how to interpret it from the perspective of FA. We then introduce MFA and MFALV and make comparisons with the i-vector in terms of formulation and computational complexity. In Section IV, we analyze the relationship between GSV, the i-vector and their extensions, and explain the importance of UBM. In Section V, we make comparisons and discussions about GSV, the i-vector and their extensions in two speaker identification tasks and one speaker verification task, in terms of effectiveness and efficiency. In Section VI, a conclusion is drawn, together with some inspirations.

II. GSV AND ITS EXTENSIONS

A. Formulation of GSV

GSV is based on adapting the parameters of a universal background model (UBM). The UBM is usually a GMM. Suppose a GMM-based UBM has been constructed, denoted as $\Theta = \{\theta_m \mid m = 1, 2...M\}$, where $\theta_m = \{\omega_m, \mu_m, \sigma_m\}$ represents the parameters of the m-th Gaussian component with weight ω_m, mean μ_m and standard deviation σ_m (σ_m is a vector, assuming the covariance of each Gaussian component is

diagonal). Given an acoustic sample *s*, denoted as a sequence of *T* frame-level feature vectors $\{ x_1 , x_2 ,\ldots x_T \}$, the posterior probability $\eta_m(x_t)$ with respect to component θ_m given the observation of a vector x_t is calculated using (1), where $p(x_t \mid \mu_m, \sigma_m)$ is the Gaussian probability with parameters $\{\mu_m, \sigma_m\}$.

$$\eta_m(x_t) = \frac{\omega_m p(x_t \mid \mu_m, \sigma_m)}{\sum_{j=1}^{M} \omega_j p(x_t \mid \mu_j, \sigma_j)} \tag{1}$$

Then the zero-order, first-order and second-order sufficient statistics are computed as given by (2) ~ (4) respectively, based on maximum likelihood estimation (MLE) [1]. The square operation is an elementwise operation.

$$E_m[x^0] = \frac{1}{T}\sum_{t=1}^{T} \eta_m(x_t) \tag{2}$$

$$E_m[x] = \frac{\sum_{t=1}^{T} \eta_m(x_t) x_t}{\sum_{t=1}^{T} \eta_m(x_t)} \tag{3}$$

$$E_m[x^2] = \frac{\sum_{t=1}^{T} \eta_m(x_t) x_t^2}{\sum_{t=1}^{T} \eta_m(x_t)} \tag{4}$$

Having the sufficient statistics, a new set of parameters for each component can be obtained based on maximum a posteriori (MAP) adaptation, as given by (5) ~ (7), where $\tilde{\omega}_m$, $\tilde{\mu}_m$ and $\tilde{\sigma}_m$ denote the adapted weight, adapted mean and adapted standard deviation, respectively, α_m is the adaptation coefficient given by (8), where *r* is the relevance factor, and λ is an automatically determined factor used to ensure the weights sum to unity [1]. The square operation is an elementwise operation. MAP adaptation adjusts the parameters of the UBM towards the statistics of the sample *s*, and the relevance factor *r* indicates the degree of adjustment.

$$\tilde{\omega}_m = \lambda\left(\alpha_m E_m[x^0] + (1-\alpha_m)\omega_m\right) \tag{5}$$

$$\tilde{\mu}_m = \alpha_m E_m[x] + (1-\alpha_m)\mu_m \tag{6}$$

$$\tilde{\sigma}_m^2 = \alpha_m E_m[x^2] + (1-\alpha_m)(\sigma_m^2 + \mu_m^2) - \tilde{\mu}_m^2 \tag{7}$$

where

$$\alpha_m = \frac{E_m[x^0]}{E_m[x^0] + r/T} \tag{8}$$

GSV is then the concatenation of the adapted mean vectors $\tilde{\mu}_m$ for *m*=1,2…*M*, as given by (9) [4]. If the dimensionality of x_t is *D*×1, then the dimensionality of GSV is *MD*×1.

$$X_{GSV} = \begin{bmatrix} \tilde{\mu}_1^T & \tilde{\mu}_2^T & \cdots & \tilde{\mu}_M^T \end{bmatrix}^T \tag{9}$$

B. Rationale behind GSV

The proposal of GSV originates from the difficulty of utilizing the KL divergence to compare two distributions for speaker recognition [4]. Before the emergence of GSV, an acoustic sample *s* can be represented by an adapted GMM with parameters $\tilde{\Theta} = \{\tilde{\theta}_m \mid m = 1, 2...M\}$, where $\tilde{\theta}_m = \{\tilde{\omega}_m, \tilde{\mu}_m, \tilde{\sigma}_m\}$ is given by (5) ~ (7). For two acoustic samples *a* and *b*, suppose their corresponding adapted GMMs are represented by $\tilde{\Theta}^{(a)} = \{\tilde{\theta}_m^{(a)} \mid m = 1, 2...M\}$ and $\tilde{\Theta}^{(b)} = \{\tilde{\theta}_m^{(b)} \mid m = 1, 2...M\}$, with $\tilde{\theta}_m^{(a)} = \{\tilde{\omega}_m^{(a)}, \tilde{\mu}_m^{(a)}, \tilde{\sigma}_m^{(a)}\}$ and $\tilde{\theta}_m^{(b)} = \{\tilde{\omega}_m^{(b)}, \tilde{\mu}_m^{(b)}, \tilde{\sigma}_m^{(b)}\}$. The KL divergence between $\tilde{\Theta}^{(a)}$ and $\tilde{\Theta}^{(b)}$ is defined in (10), where $p_m^{(a)}(x)$ and $p_m^{(b)}(x)$ represent the Gaussian probability with parameters $\{\tilde{\mu}_m^{(a)}, \tilde{\sigma}_m^{(a)}\}$ and $\{\tilde{\mu}_m^{(b)}, \tilde{\sigma}_m^{(b)}\}$, respectively.

$$D_{KL}(\tilde{\Theta}^{(a)} \,||\, \tilde{\Theta}^{(b)}) = \int_x \left(\sum_{m=1}^{M} \tilde{\omega}_m^{(a)} p_m^{(a)}(x) \right) \ln \left(\frac{\sum_{m=1}^{M} \tilde{\omega}_m^{(a)} p_m^{(a)}(x)}{\sum_{m=1}^{M} \tilde{\omega}_m^{(b)} p_m^{(b)}(x)} \right) dx \tag{10}$$

The KL divergence between two GMMs is upper bounded by the weighted sum of the KL divergence between each pair of Gaussian components, as expressed in (11) [32].

$$D_{KL}(\tilde{\Theta}^{(a)} \,||\, \tilde{\Theta}^{(b)}) \le D_{KL}(\tilde{\omega}_m^{(a)} \,||\, \tilde{\omega}_m^{(b)}) + \sum_{m=1}^{M} \tilde{\omega}_m^{(a)} D_{KL}(p_m^{(a)} \,||\, p_m^{(b)}) \tag{11}$$

If we assume $\tilde{\sigma}_m^{(a)} = \tilde{\sigma}_m^{(b)} = \sigma_m$ and $\tilde{\omega}_m^{(a)} = \tilde{\omega}_m^{(b)} = \omega_m$, and define a diagonal matrix Σ_m whose *ii*-th element is the square of the *i*-th element of σ_m, (11) is further simplified to (12). This assumption on the adapted weights and adapted standard deviations ensures that the upper bound can be written in the form of a Euclidean distance. The result in (12) is also given by [4], but some details are not mentioned, such as the assumption of some adapted parameters being equal. Therefore, we provide a detailed proof in the appendix.

$$D_{KL}(\tilde{\Theta}^{(a)} \,||\, \tilde{\Theta}^{(b)}) \le \sum_{m=1}^{M} \left\| \sqrt{\frac{\omega_m}{2}} \Sigma_m^{-1/2} \tilde{\mu}_m^{(a)} - \sqrt{\frac{\omega_m}{2}} \Sigma_m^{-1/2} \tilde{\mu}_m^{(b)} \right\|^2 \tag{12}$$

The righthand side in (12) is the distance between two normalized GSVs (nGSV) given by (13), where the vector division operation is an elementwise operation. This implies that under the assumption that the weight and the covariance of the two adapted GMMs are the same, the distance between the two adapted GMMs can be approximated by the distance

between the two nGSVs. This finding lays the foundation for GSV.

$$X_{nGSV}=\left[\sqrt{\frac{\omega_1}{2}}\left(\frac{\tilde{\mu}_1}{\sigma_1}\right)^T \quad \sqrt{\frac{\omega_2}{2}}\left(\frac{\tilde{\mu}_2}{\sigma_2}\right)^T \quad \cdots \quad \sqrt{\frac{\omega_M}{2}}\left(\frac{\tilde{\mu}_M}{\sigma_M}\right)^T\right]^T \quad (13)$$

The distance between two GMMs can also be measured using the Bhattacharyya distance, which can then be approximated as the Euclidean distance between two GMM-UBM mean interval (GUMI) supervectors [49]. A GUMI supervector performs normalization using the average of the covariances of the two GMMs; therefore, its computation depends on the statistics of a pair of acoustic samples. However, given an acoustic sample, we may prefer that a feature representation is generated merely from this specific sample, instead of a pair of samples.

C. Generalization of GSV: MAP Supervector

GSV is merely the concatenation of the adapted mean vectors. Naturally, we may wonder whether it is feasible to concatenate other parameters, such as the adapted weights and adapted standard deviations, as some studies show that the weight parameter may provide complementary information to the mean parameter [50]. This idea is heuristic but simple to implement, leading to a more general type of feature representation, which we name the MAP supervector (MAP SV). The MAP adaptation procedure in (5) ~ (7) is highly intuitive and meaningful. It indicates that the adapted parameters are simply the weighted sum of the UBM parameters (i.e., prior parameters) and the sample-based parameters (i.e., posterior parameters). The key parameter is the posterior probability provided in (1). This posterior probability can also be provided by other types of UBM, such as DBN [20].

In general, we have MAP weight SV (WSV), MAP mean SV (MSV), which is GSV, and MAP variance SV (VSV), as given by (14) ~ (16), respectively. If the dimensionality of x_t is D×1, then WSV has a dimensionality of M×1, MSV has a dimensionality of MD×1, and VSV has a dimensionality of MD×1. We may further concatenate WSV, MSV and VSV to form a concatenated SV having a dimensionality of (M+2MD)×1, as given by (17), which may carry more abundant information and be more robust than the SV based on a single type of adapted parameters. Further operations can be applied to MAP SV, such as performing factor analysis on WSV [50].

$$X_{MAP,\omega}=\left[\tilde{\omega}_1 \quad \tilde{\omega}_2 \quad \cdots \quad \tilde{\omega}_M\right]^T \quad (14)$$

$$X_{MAP,\mu}=\left[\tilde{\mu}_1^T \quad \tilde{\mu}_2^T \quad \cdots \quad \tilde{\mu}_M^T\right]^T \quad (15)$$

$$X_{MAP,\sigma}=\left[\tilde{\sigma}_1^T \quad \tilde{\sigma}_2^T \quad \cdots \quad \tilde{\sigma}_M^T\right]^T \quad (16)$$

$$X_{MAP,\omega,\mu,\sigma}=\left[X_{MAP,\omega}^T \quad X_{MAP,\mu}^T \quad X_{MAP,\sigma}^T\right]^T \quad (17)$$

In addition to performing the normalization using the parameters of the UBM, which is the case of nGSV, we may alternatively perform the normalization using the adapted parameters, yielding the normalized MSV (nMSV), as given by (18), where the vector division operation is an elementwise operation.

$$X_{nMSV}=\left[\sqrt{\frac{\tilde{\omega}_1}{2}}\left(\frac{\tilde{\mu}_1}{\tilde{\sigma}_1}\right)^T \quad \sqrt{\frac{\tilde{\omega}_2}{2}}\left(\frac{\tilde{\mu}_2}{\tilde{\sigma}_2}\right)^T \quad \cdots \quad \sqrt{\frac{\tilde{\omega}_M}{2}}\left(\frac{\tilde{\mu}_M}{\tilde{\sigma}_M}\right)^T\right]^T \quad (18)$$

Usually, the relevance factor r in (8) is determined according to some prior experimental results. In this paper, we propose a heuristic but reasonable way to semiautomatically determine the value of r, which is given by (19), where β is a scaling factor, T is the number of frame-level feature vectors for the sample s, and M is the number of mixture components in the UBM. Therefore, r can be different for different samples, depending on T.

$$r=\beta\frac{T}{M} \quad (19)$$

The rationale of (19) is explained as follows. When $\beta=1$, we have $\sum_{m=1}^{M} r/T=1$. We also have $\sum_{m=1}^{M} E_m[x^0]=1$ according to (1) and (2). This means that, if the sample s fits well to some Gaussian component m such that $E_m[x^0]>r/T$, then $\alpha_m>0.5$, causing the adapted parameters to be adjusted more towards the sample s instead of the m-th component in the UBM. Therefore, the larger the posterior probability $\eta_m(x_t)$ is, the greater the adapted parameters will be adjusted towards the sample; the smaller the posterior probability is, the greater the adapted parameters will be adjusted towards the m-th component in the UBM. β controls the dependence of the adapted parameters on the sample or the UBM. When $\beta<1$, the adapted parameters are more dependent on the sample; when $\beta>1$, the adapted parameters are more dependent on the UBM. Therefore, $\beta=1$ is the critical point, and the appropriate value of β can be determined by varying the value in the neighborhood of $\beta=1$. On this ground, determining the value of β is relatively simpler than directly determining the value of r, because r can have any value. In [51], an adaptive relevance factor is proposed, which makes the adaptation less affected by T. However, the adaptive relevance factor is considerably more complicated, and different mixture components have different relevance factors.

D. Beyond MAP Supervector

The Supervector can be a very generic feature representation. GSV or MAP SV is based on the concatenation of the MAP adapted parameters. By using other ways to compute the adapted parameters, such as computing the gradients of the UBM [43], other types of supervectors can be constructed. In

addition, the UBM can be any type of mixture model, such as GMM or MFA, or deep neural networks, such as DBN, or even the combination of GMM and DBN. The feasibility of such a combination lies in the fact that the adapted parameters are simply concatenated instead of interacting with each other. The flexibility in the design of the UBM and the adaptation method leads to various types of feature representations.

III. I-VECTOR AND ITS EXTENSIONS

A. Formulation of the I-vector

The i-vector is based on the parameters of a UBM, which is usually a GMM, similar to GSV. From the perspective of the i-vector, an acoustic sample $s=\{x_1, x_2, \ldots x_T\}$ is supposed to be able to be represented by a supervector X_s, which is assumed to be generated by a latent vector z_s and expressed as a factor analysis (FA) model given by (20), where μ_{UBM} is the concatenation of the mean vectors of the UBM with $\Theta=\{\theta_m \mid m=1,2...M\}$ and $\theta_m=\{\omega_m,\mu_m,\sigma_m\}$, V is the factor-loading matrix, and ε_s is the noise term with a zero mean and a diagonal covariance Ψ [10]. The subscript s indicates that both X_s and z_s are dependent on the sample s. The expected value of z_s is then the i-vector, denoted as $E[z_s]$. If the dimensionality of x_t is $D\times1$, the dimensionality of X_s will then be $MD\times1$, but the dimensionality of z_s can be smaller, which is the advantage of the i-vector over GSV.

$$X_s = \mu_{UBM} + Vz_s + \varepsilon_s \tag{20}$$

An interesting characteristic is that, both X_s and z_s are unobservable, but their expected values can be estimated using the expectation-maximization (EM) algorithm [33]. Before using the EM algorithm, the zero-order, first-order and second-order centralized Baum-Welch statistics need to be computed, as given by (21) ~ (23), respectively.

$$\hat{E}_m[x^0] = \sum_{t=1}^{T} \eta_m(x_t) \tag{21}$$

$$\hat{E}_m[x] = \sum_{t=1}^{T} \eta_m(x_t)(x_t-\mu_m) \tag{22}$$

$$\hat{E}_m[xx^T] = \sum_{t=1}^{T} \eta_m(x_t)(x_t-\mu_m)(x_t-\mu_m)^T \tag{23}$$

The centralized Baum-Welch statistics are used to form the sample-level zero-order, first-order and second-order statistics, as given by (24) ~ (26), respectively, where I is an identity matrix with a dimensionality of $D\times D$. The subscript s indicates that these statistics depend on a specific sample s.

$$E_s[X^0] = \begin{bmatrix} \hat{E}_1[x^0]I & 0 & \cdots & 0 \\ 0 & \hat{E}_2[x^0]I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{E}_M[x^0]I \end{bmatrix} \tag{24}$$

$$E_s[X] = \begin{bmatrix} \hat{E}_1[x]^T & \hat{E}_2[x]^T & \cdots & \hat{E}_M[x]^T \end{bmatrix}^T \tag{25}$$

$$E_s[XX^T] = \begin{bmatrix} \hat{E}_1[xx^T] & 0 & \cdots & 0 \\ 0 & \hat{E}_2[xx^T] & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{E}_M[xx^T] \end{bmatrix} \tag{26}$$

Having the sample-level statistics, the factor-loading matrix V is obtained using the EM algorithm, which involves an E-step and an M-step. In the E-step, the posterior expected mean and the posterior expected covariance are computed using (27) and (28) respectively, where the parameters $\{V,\Psi\}$ are computed in the M-step.

$$E[z_s] = (I + V^T\Psi^{-1}E_s[X^0]V)^{-1}V^T\Psi^{-1}E_s[X] \tag{27}$$

$$E[z_s z_s^T] = (I + V^T\Psi^{-1}E_s[X^0]V)^{-1} + E[z_s]E[z_s]^T \tag{28}$$

In the M-step, the model parameters $\{V,\Psi\}$ are computed using the posterior expectations in the E-step and a set of training acoustic samples S as given by (29) and (30), where $diag\{.\}$ is the operation that sets all the nondiagonal elements to zero.

$$\sum_{s\in S} E_s[X^0]VE[z_s z_s^T] = \sum_{s\in S} E_s[X]E[z_s]^T \tag{29}$$

$$\sum_{s\in S} E_s[X^0]\Psi = diag\left\{\sum_{s\in S}\left(E_s[XX^T] - VE[z_s]E_s[X]^T\right)\right\} \tag{30}$$

After several EM iterations and finding the values of $\{V,\Psi\}$, given a sample s, the i-vector is obtained using (27). Having obtained the i-vector corresponding to a sample s, we can then recover the unobservable supervector X_s using its expected value $E[X_s]$. To be clearer, we call the i-vector the FA latent vector (FALV) and $E[X_s]$ the FA supervector (FASV), as given by (31) and (32), respectively.

$$X_{FALV} = E[z_s] \tag{31}$$

$$X_{FASV} = E[X_s] = \mu_{UBM} + VE[z_s] = \mu_{UBM} + VX_{FALV} \tag{32}$$

B. Rationale behind the I-vector

Strictly speaking, the i-vector is not the consequence of a standard FA model. Although (20) is the same as the assumption of an FA model, the parameters $\{V,\Psi\}$ are computed in a different way [34]. Additionally, in the formulation of the i-vector, the high-dimensional vector X_s is unobservable, which makes the formulation of standard FA inapplicable. Nevertheless, this inconsistency does not mean that the performance of the i-vector is poor. In contrast, the i-vector performs notably well, as the model parameters are estimated by MLE [33]. The log-likelihood $\mathcal{L}(s)$ of a sample *s* is given by (33), where $p(x_t \mid X_{s,m},\Psi_m)$ is the Gaussian probability with mean $X_{s,m}$ and covariance Ψ_m, and Ψ_m is the *m*-th submatrix of Ψ, i.e., Ψ_m consists of the variables between row $(m-1)D+1$ and row mD and between column $(m-1)D+1$ and column mD (as Ψ is a diagonal matrix). $X_{s,m}$ is given by (34), where V_m is the *m*-th submatrix of *V*, which consists of the variables between row $(m-1)D+1$ and row mD. The log-likelihood given by (33) is thus equivalent to that in [33] when adopting the i-vector assumption in [10].

$$\begin{aligned}\mathcal{L}(s) &= \sum_{m=1}^{M}\sum_{t=1}^{T}\ln p(x_t \mid X_{s,m},\Psi_m)^{\eta_m(x_t)} \\ &= \sum_{m=1}^{M}\sum_{t=1}^{T}\eta_m(x_t)\ln p(x_t \mid X_{s,m},\Psi_m) \\ &= \sum_{m=1}^{M}\sum_{t=1}^{T}\eta_m(x_t)\ln\frac{1}{(2\pi)^{D/2}\left|\Psi_m\right|^{1/2}} \\ &\quad -\sum_{m=1}^{M}\sum_{t=1}^{T}\frac{1}{2}\eta_m(x_t)(x_t-X_{s,m})^T\Psi_m^{-1}(x_t-X_{s,m})\end{aligned} \tag{33}$$

where

$$X_{s,m} = \mu_m + V_m z_s \tag{34}$$

The log-likelihood of a set of training samples is then given by (35), which is the objective function needing to be maximized with respect to $\{V,\Psi\}$.

$$\mathcal{L} = \sum_{s\in S}\mathcal{L}(s) \tag{35}$$

The formulation of the i-vector resembles a standard FA model, but the model parameter estimation procedure resembles more of a mixture of factor analyzers (MFA) [35]. Nevertheless, the i-vector is neither the result of FA nor MFA. With respect to a mixture component *m*, a frame-level vector x_t is assumed to follow a Gaussian distribution with mean $\mu_m + V_m z_s$ and covariance Ψ_m. The parameters $\{\mu_m, V_m, \Psi_m\}$ are different for different mixture components, but the latent vector z_s is shared across different mixture components. This makes it suitable to estimate z_s with limited training data, as all the training data are reused *M* times, which means the training data are augmented *M* times with respect to z_s.

C. Origin of the I-vector: Baum-Welch Supervector

In a standard FA model, a high-dimensional vector *Y* is assumed to be generated by a low-dimensional latent vector *y*, as expressed in (36), where μ is the global mean, *W* is the factor-loading matrix, and ε is the noise vector, which is assumed to follow a Gaussian distribution with zero mean and diagonal covariance Σ.

$$Y = \mu + Wy + \varepsilon \tag{36}$$

The model parameters $\{\mu, W, \Sigma\}$ are estimated using the EM algorithm [34]. The E-step is given by (37) and (38).

$$E[y] = (I + W^T\Sigma^{-1}W)^{-1}W^T\Sigma^{-1}(Y-\mu) \tag{37}$$

$$E[yy^T] = (I + W^T\Sigma^{-1}W)^{-1} + E[y]E[y]^T \tag{38}$$

Interestingly, the E-step for the i-vector can be reformulated into the same form as FA, given by (39) and (40), with $y = z_s$, $W = E_s[X^0]V$, $\Sigma = E_s[X^0]\Psi$, and $Y - \mu = E_s[X]$.

$$\begin{aligned}E[z_s] = &\left(I + \left(E_s[X^0]V\right)^T\left(E_s[X^0]\Psi\right)^{-1}\left(E_s[X^0]V\right)\right)^{-1} \\ &\times\left(E_s[X^0]V\right)^T\left(E_s[X^0]\Psi\right)^{-1}E_s[X]\end{aligned} \tag{39}$$

$$\begin{aligned}E[z_s z_s^T] = &\left(I + \left(E_s[X^0]V\right)^T\left(E_s[X^0]\Psi\right)^{-1}\left(E_s[X^0]V\right)\right)^{-1} \\ &+ E[z_s]E[z_s]^T\end{aligned} \tag{40}$$

In this reformulation, the latent vector z_s corresponds to a high-dimensional vector given by (41), which we name the Baum-Welch supervector (BWSV), as it is based on the centralized Baum-Welch statistics.

$$X_{BWSV} = E_s[X] \tag{41}$$

Besides, the E-step for the i-vector can also be reformulated in another way, which still has the same form as FA, given by (42) and (43), where $y = z_s$, $W = V$, $\Sigma = E_s[X^0]^{-1}\Psi$, and $Y - \mu = E_s[X^0]^{-1}E_s[X]$.

$$\begin{aligned}E[z_s] = &\left(I + V^T\left(E_s[X^0]^{-1}\Psi\right)^{-1}V\right)^{-1} \\ &\times V^T\left(E_s[X^0]^{-1}\Psi\right)^{-1}\left(E_s[X^0]^{-1}E_s[X]\right)\end{aligned} \tag{42}$$

$$E[z_s z_s^T] = \left(I + V^T \left(E_s[X^0]^{-1} \Psi \right)^{-1} V \right)^{-1} + E[z_s] E[z_s]^T \quad (43)$$

In this way, the latent vector z_s corresponds to another high-dimensional vector given by (44), which we name the normalized BWSV (nBWSV), as it has an extra term $E_s[X^0]^{-1}$ serving as a sort of normalization.

$$X_{nBWSV} = E_s[X^0]^{-1} E_s[X] \quad (44)$$

We may also add back the global mean, which will be μ_{UBM} from the perspective of the i-vector. Then, we obtain the mean-shifted BWSV $E_s[X] + \mu_{UBM}$ and the mean-shifted nBWSV $E_s[X^0]^{-1} E_s[X] + \mu_{UBM}$. Interestingly, the *m*-th subvector of the mean-shifted nBWSV (i.e., the variables between row $(m-1)D+1$ and row mD) can be expressed as (45), which is exactly the *m*-th subvector of GSV with $\alpha_m = 1$. This implies that the i-vector is an affine transformation of GSV.

$$\begin{aligned} \hat{E}_m[x^0]^{-1} \hat{E}_m[x] + \mu_m &= \frac{\sum_{t=1}^{T} \eta_m(x_t)(x_t - \mu_m)}{\sum_{t=1}^{T} \eta_m(x_t)} + \mu_m \\ &= \frac{\sum_{t=1}^{T} \eta_m(x_t) x_t}{\sum_{t=1}^{T} \eta_m(x_t)} \end{aligned} \quad (45)$$

D. Extension of the I-vector: MFA Supervector and Latent Vector

An MFA can be parameterized by $\Theta = \{\theta_m \mid m = 1, 2 ... M\}$, where $\theta_m = \{w_m, u_m, W_m, \Sigma\}$ represents the parameters of the *m*-th factor analyzer with weight w_m, mean u_m, factor-loading matrix W_m, and noise covariance Σ (Σ is shared across different factor analyzers) [35]. Based on this MFA model, a frame-level feature vector x_t can be expressed as (46), where z_t is the corresponding latent vector presumed to follow a Gaussian distribution with zero mean and identity covariance, and ε_t is the noise term presumed to follow a Gaussian distribution with zero mean and diagonal covariance Σ .

$$x_t = \sum_{m=1}^{M} w_m (u_m + W_m z_t) + \varepsilon_t \quad (46)$$

The conditional probability of x_t generated by the *m*-th factor analyzer given the latent vector z_t, is given by (47), where $p(x_t \mid u_m + W_m z_t, \Sigma)$ is the Gaussian probability with mean $u_m + W_m z_t$ and covariance Σ .

$$p(x_t \mid z_t, m) = p(x_t \mid u_m + W_m z_t, \Sigma) \quad (47)$$

Then, the probability of x_t generated by the *m*-th factor analyzer and that generated by the MFA is given by (48) and (49), respectively.

$$p(x_t \mid m) = p(x_t \mid u_m, W_m W_m^T + \Sigma) \quad (48)$$

$$p(x_t \mid \Theta) = \sum_{m=1}^{M} w_m p(x_t \mid u_m, W_m W_m^T + \Sigma) \quad (49)$$

Given a set of training vectors $\{ x_1, x_2, \ldots x_N \}$, the model parameters $\{w_m, u_m, W_m, \Sigma\}$ can be estimated using the EM algorithm [35], similar to the estimation process of FA and GMM. In the E-step, the posterior expected mean and the posterior expected covariance of the latent vectors $\{ z_1, z_2, \ldots z_N \}$ with respect to the *m*-th factor analyzer are computed using (50) and (51), respectively.

$$E_m[z_n] = (I + W_m^T \Sigma^{-1} W_m)^{-1} W_m^T \Sigma^{-1} (x_n - u_m) \quad (50)$$

$$E_m[z_n z_n^T] = (I + W_m^T \Sigma^{-1} W_m)^{-1} + E_m[z_n] E_m[z_n]^T \quad (51)$$

The posterior probability is computed using (52).

$$\hat{\eta}_m(x_n) = \frac{w_m p(x_n \mid u_m, W_m W_m^T + \Sigma)}{\sum_{j=1}^{M} w_j p(x_n \mid u_j, W_j W_j^T + \Sigma)} \quad (52)$$

In the M-step, the model parameters $\{w_m, u_m, W_m, \Sigma\}$ are re-estimated using the posterior expectations $E_m[z_n]$ and $E_m[z_n z_n^T]$ and the posterior probability $\hat{\eta}_m(x_n)$. The weight w_m is computed using (53).

$$w_m = \frac{1}{N} \sum_{n=1}^{N} \hat{\eta}_m(x_n) \quad (53)$$

To ease the computation, the augmented posterior expectations $E_m[\tilde{z}_n]$ and $E_m[\tilde{z}_n \tilde{z}_n^T]$ and the augmented factor-loading matrix $\tilde{W}_m$ are formed, as given by (54).

$$\begin{aligned} E_m[\tilde{z}_n] &= \begin{bmatrix} E_m[z_n] \\ 1 \end{bmatrix}, \quad E_m[\tilde{z}_n \tilde{z}_n^T] = \begin{bmatrix} E_m[z_n z_n^T] & E_m[z_n] \\ E_m[z_n]^T & 1 \end{bmatrix} \\ \tilde{W}_m &= \begin{bmatrix} W_m & u_m \end{bmatrix} \end{aligned} \quad (54)$$

Using the augmented posterior expectations, $\tilde{W}_m$ and Σ can be computed using (55) and (56), respectively.

$$\tilde{W}_m = \left(\sum_{n=1}^{N} \hat{\eta}_m(x_n) x_n E_m[\tilde{z}_n]^T \right) \left(\sum_{n=1}^{N} \hat{\eta}_m(x_n) E_m[\tilde{z}_n \tilde{z}_n^T] \right)^{-1} \quad (55)$$

$$\Sigma = \frac{1}{N} diag\left\{ \sum_{n=1}^{N} \sum_{m=1}^{M} \hat{\eta}_m(x_n)\left(x_n - \tilde{W}_m E_m[\tilde{z}_n]\right) x_n^T \right\} \quad (56)$$

After several EM iterations and obtaining the parameters $\{w_m, u_m, W_m, \Sigma\}$, given a sample s={ $x_1, x_2, \ldots x_T$ }, a sequence of latent vectors { $E_m[z_1]$, $E_m[z_2]$,… $E_m[z_T]$ } can be obtained using (50). The MFA latent vector (MFALV) is then obtained as the weighted sum of { $E_m[z_1]$, $E_m[z_2]$,… $E_m[z_T]$ }, weighted by $\hat{\eta}_m(x_t)$. The *m*-th subvector of MFALV is given by (57), and MFALV is then the concatenation of its subvectors as given by (58).

$$X_{MFALV,m} = \frac{\sum_{t=1}^{T} \hat{\eta}_m(x_t) E_m[z_t]}{\sum_{t=1}^{T} \hat{\eta}_m(x_t)} \quad (57)$$

$$X_{MFALV} = \begin{bmatrix} X_{MFALV,1}^T & X_{MFALV,2}^T & \cdots & X_{MFALV,M}^T \end{bmatrix}^T \quad (58)$$

Similar to the i-vector, we can form an MFA supervector (MFASV) based on MFALV, as given by (59) and (60).

$$X_{MFASV,m} = u_m + W_m X_{MFALV,m} \quad (59)$$

$$X_{MFASV} = \begin{bmatrix} X_{MFASV,1}^T & X_{MFASV,2}^T & \cdots & X_{MFASV,M}^T \end{bmatrix}^T \quad (60)$$

E. Comparison between the Formulation and Computational Complexity of the I-vector and MFA

Given a set of training samples { $a_1, a_2, \ldots a_S$ } where *S* is the total number of training samples. Suppose the *s*-th sample is represented by a sequence of T_s frame-level feature vectors, denoted as { $x_1, x_2, \ldots x_{T_s}$ }, then according to (33) ~ (35), from the perspective of the i-vector, the parameters $\{V, \Psi\}$ are obtained by maximizing the log-likelihood $\mathcal{L}_{i-vector}$, as given by (61), where μ_m is the mean parameter of GMM, and the exponent $\eta_m(x_t)$ is based on the GMM, as given by (1). This log-likelihood is based on the individual sample's log-likelihood $\mathcal{L}(s)$.

$$\mathcal{L}_{i-vector} = \sum_{s=1}^{S} \mathcal{L}(s) = \sum_{s=1}^{S} \sum_{m=1}^{M} \sum_{t=1}^{T_s} \ln p(x_t \mid \mu_m + V_m z_s, \Psi_m)^{\eta_m(x_t)} \quad (61)$$

From the perspective of MFA, the log-likelihood is based on the individual frame-level feature vector's log-likelihood $\mathcal{L}(n)$. The training vectors are collected from all the training samples, denoted as { x_1 , x_2 ,… x_N }, where x_n denotes the *n*-th frame-level feature vector, and $N = \sum_{s=1}^{S} T_s$. Then, the parameters $\{W_m, \Sigma\}$ of MFA can be treated as the solution of maximizing the log-likelihood $\mathcal{L}_{MFA}$ given by (62), where u_m and the exponent w_m are the parameters of MFA.

$$\mathcal{L}_{MFA} = \sum_{n=1}^{N} \mathcal{L}(n) = \sum_{m=1}^{M} \sum_{n=1}^{N} \ln p(x_n \mid u_m + W_m z_n, \Sigma)^{w_m} \quad (62)$$

The differences between the formulation of the i-vector and that of MFA can be seen from (61) and (62). For instance, in $\mathcal{L}_{i-vector}$, each sample is considered independent, and the frame-level feature vectors in the same sample share the same latent vector, whereas in $\mathcal{L}_{MFA}$, each frame-level feature vector is considered independent, and even those from the same sample have different latent vectors. In addition, in $\mathcal{L}_{i-vector}$, μ_m is the *m*-th mean vector of the GMM, and the exponent $\eta_m(x_t)$ is the posterior probability based on the GMM. While in $\mathcal{L}_{MFA}$, both u_m and the exponent w_m are re-estimated based on the training data, which are dependent on the MFA instead of the GMM. This means that MFA has more parameters to be estimated than the i-vector, endowing it with more flexibility. However, for the i-vector, since the same latent vector is shared among all the frame-level feature vectors that belong to the same sample, the freedom of the parameter space is reduced, which makes the i-vector more robust.

Suppose the dimensionality of the i-vector or MFALV is *H*×1. According to (27) and (28), the E-step for the i-vector requires computing the inverse of a matrix $(I + V^T \Psi^{-1} E_s[X^0] V)$, whose size is *H*×*H*. Therefore, if *H* is notably large, the computation of the i-vector is inefficient, and sometimes may even be infeasible if the memory space is not enough. In contrast, as seen from (50) and (51), the E-step for MFA requires computing the inverse of a smaller matrix $(I + W_m^T \Sigma^{-1} W_m)$, whose size is (*H*/*M*)×(*H*/*M*). Because we assume that the dimensionality of X_{MFALV} is *H*×1, the dimensionality of $X_{MFALV,m}$ has to be *H*/*M*; thus, the dimensionality of the latent vector in (50) and (51) is *H*/*M* (assuming *H* is an integer multiple of *M*). Therefore, for MFA, even if the dimensionality of MFALV is large, the computation can still be efficient.

Specifically, let the number of training samples be *S*, the dimensionality of the frame-level feature vector be *D*×1, the dimensionality of the i-vector or MFALV be *H*×1, and the number of mixture components in the GMM or the MFA be *M*. Then, for the i-vector, the size of *V* is *MD*×*H*, the size of Ψ is *MD*×*MD*, the size of $E_s[X^0]$ is *MD*×*MD*, and the size of $E_s[X]$ is *MD*×1. For MFA, the size of u_m is *D*×1, the size of W_m is *D*×(*H*/*M*), and the size of Σ is *D*×*D*. Suppose each sample produces *T* frame-level feature vectors, and assume the time complexity of inverting a matrix with size *A*×*A* is $O(A^3)$, and the time complexity of the multiplication of a matrix with size *A*×*B* and a matrix with size *B*×*C* is $O(ABC)$. Considering the E-step of the i-vector given by (27) and (28) and the E-step

of MFA given by (50) and (51), if we neglect the time consumption of addition operations, the time consumption of inverting Ψ and Σ (because they are diagonal matrices whose inversion is easy to compute), and the time consumption of $E[z_s z_s^T]$ and $E_m[z_n z_n^T]$ (because most time will be consumed by computing $E[z_s]$ and $E_m[z_n]$), then for the i-vector, the time complexity of the E-step is approximately given by (63).

$$\begin{aligned} C_{E-step}^{(i-vector)} &\approx S \times \{ O(H \times MD \times MD \times MD \times H) \\ &\quad + O(H^3) + O(H \times H \times MD \times MD \times 1) \} \\ &= SH^2 \times \left(O\left(M^3 D^3\right) + O(H) + O\left(M^2 D^2\right) \right) \end{aligned} \quad (63)$$

Similarly, the time complexity of the E-step for MFA is approximately given by (64).

$$\begin{aligned} C_{E-step}^{(MFA)} &\approx MST \times \left\{ O\left(\frac{H}{M} \times D \times D \times \frac{H}{M} \right) \right. \\ &\quad \left. + O\left(\frac{H^3}{M^3} \right) + O\left(\frac{H}{M} \times \frac{H}{M} \times D \times D \times 1 \right) \right\} \\ &= SH^2 \frac{T}{M^2} \times \left(O\left(2MD^2\right) + O(H) \right) \end{aligned} \quad (64)$$

The factor S in (63) is the number of training samples, as the expectation term $E[z_s]$ for the i-vector corresponds to the whole sample. The factor ST in (64) is the number of training frame-level feature vectors, as the expectation term $E_m[z_n]$ for MFA corresponds to a frame-level feature vector. Another factor M is also needed, as $E_m[z_n]$ corresponds to only one mixture component in the MFA. The inclusion of these multiplication factors can be clearly seen from the algorithms given in the appendix. As the E-step dominates the time consumption of the EM iteration, the time complexity of training the model parameters for the i-vector and that for MFALV can be approximated by (63) and (64), respectively.

Given an acoustic sample, its corresponding i-vector is $E[z_s]$, while its corresponding MFALV is the weighted sum of $E_m[z_n]$, as seen from (31) and (57), respectively. Therefore, given the model parameters, the time complexity of computing an i-vector and that of computing an MFALV can be approximated by (65) and (66), respectively.

$$C_{feature}^{(i-vector)} \approx \frac{C_{E-step}^{(i-vector)}}{S} = H^2 \times \left(O\left(M^3 D^3\right) + O(H) + O\left(M^2 D^2\right) \right) \quad (65)$$

$$C_{feature}^{(MFALV)} \approx \frac{C_{E-step}^{(MFA)}}{S} = H^2 \frac{T}{M^2} \times \left(O\left(2MD^2\right) + O(H) \right) \quad (66)$$

From (63) ~ (66), if T is small such that $T<<M$, the computation of MFALV will be considerably more efficient than that of the i-vector. This implies that MFALV is more suitable for short-duration data.

Regarding the space complexity, as shown by (62), MFALV only needs to store the parameters of the MFA, namely, the weight, the mean, the factor-loading matrix, and the noise covariance (which is a diagonal matrix) parameters of the MFA. This leads to the space complexity as given by (67).

$$\begin{aligned} C_{space}^{(MFALV)} &= O\left(M + MD + MD\frac{H}{M} + D \right) \\ &= O(M + MD + DH + D) \end{aligned} \quad (67)$$

As shown by (61), the i-vector needs to store the parameters of the GMM and the FA model, namely, the weight, the mean and the standard deviation parameters of the GMM, as well as the factor-loading matrix and the noise covariance (which is a diagonal matrix) parameters of the FA model. This leads to the space complexity as given by (68).

$$\begin{aligned} C_{space}^{(i-vector)} &= O\left((M + 2MD) + (MDH + MD) \right) \\ &= O(M + 3MD + MDH) \end{aligned} \quad (68)$$

By comparing (67) and (68), MFALV requires less storage space than the i-vector. Nevertheless, the dimensionality of MFALV has to be an integer multiple of M, which is less flexible than the i-vector.

IV. Relationship between Different Vector-based Feature Representations and the Necessity of UBM

GSV and its extensions are obtained by adapting the parameters of a GMM-based UBM using MAP adaptation.

TABLE I
VECTOR-BASED FEATURE REPRESENTATION

UBM type	Abbreviation	Description	Dimensionality
GMM	WSV (Eq. (14))	Concatenation of adapted weight	D×1
	MSV (GSV) (Eq. (15))	Concatenation of adapted mean	MD×1
	VSV (Eq. (16))	Concatenation of adapted standard deviation	MD×1
	nGSV (Eq. (13))	Concatenation of adapted mean, scaled by weight and covariance	MD×1
	nMSV (Eq. (18))	Concatenation of adapted mean, scaled by adapted weight and covariance	MD×1
	Concatenated SV (Eq. (17))	Concatenation of WSV, MSV and VSV	(M+2MD)×1
MFA	FASV (Eq. (32))	Transformation of FALV, plus GMM mean	MD×1
	FALV (i-vector) (Eq. (31))	Posterior expectation of the latent vector	1×1 ~ MD×1
	MFASV (Eq. (60))	Transformation of MFALV, plus MFA mean	MD×1
	MFALV (Eq. (58))	Concatenation of the posterior expectations of the latent vectors	M×1 ~ MD×1

FALV (i-vector) and MFALV are obtained based on the parameters of an MFA-based UBM using the MLE algorithm. FASV and MFASV are based on FALV and MFALV, respectively. In addition, FALV (i-vector) makes use of both the parameters of the GMM-based UBM and the MFA-based UBM to form the feature representation, whereas MFALV merely uses the parameters of the MFA-based UBM to form the feature representation. However, the parameters of the MFA-based UBM are initialized from those of the GMM-based UBM. Interestingly, FALV can also be treated as an affine transformation of GSV.

As a matter of fact, UBM plays an important role in constructing the aforementioned feature representations. It has two main uses. First, UBM provides prior knowledge about the underlying distributions of the acoustic features. During the computation of the sample-level feature representation, the feature representation embeds both the information from that specific sample and the information from the UBM (which contains information from many samples). Therefore, UBM provides additional information for the feature representation. Second, UBM provides feature alignment. Intuitively, we may treat one mixture component in the UBM as one attribute. Each sample is fed into the UBM to adapt different mixture components, reflecting how this sample possesses different attributes. This provides feature alignment in the resulting feature representation, namely, the variables in the same position of different feature vectors represent the same attribute.

Regarding the space complexity, GSV or its extensions only need to store the parameters of the GMM, namely, the weight, the mean and the standard deviation parameters of the GMM. This leads to the space complexity of $O(M+MD+MD) = O(M+2MD)$, where D is the dimensionality of the frame-level feature vector, and M is the number of mixture components in the GMM. As shown in the previous section, if the dimensionality of MFALV and FALV (i-vector) is H, then the space complexity of MFALV is $O(M+MD+DH+D)$, while that of FALV (i-vector) is $O(M+3MD+MDH)$.

The characteristics of different vector-based feature representations are also listed in Table I.

V. Experiments and Discussions

In this section, we make experimental comparisons between different vector-based feature representations in terms of their effectiveness and efficiency. Two speaker identification tasks are performed using the Kingline081 dataset [36] and the Ahumada dataset [37]. One speaker verification task is performed using the Voxceleb1 dataset [46].

Kingline081 is an American English speech corpus consisting of continuous speech with normal speed. A part of this corpus is used in the experiments, which consists of the utterances of 20 speakers. Each speaker contributes approximately the same number of utterances. The utterances are recorded in three sessions, with each having approximately 2000 samples. The first two sessions are used for training and constructing the UBM, and the third session is used for testing. This yields a training set of 3997 utterances and a testing set of 1998 utterances. The length of each utterance varies from 2s to 10s, and the average length is approximately 4s.

Ahumada is a Spanish speech corpus consisting of text-dependent and text-independent speech at varying speeds. A part of this corpus is used in the experiments, which consists of the telephone speech utterances of 25 speakers. Each speaker contributes approximately the same number of utterances. The utterances are recorded in four sessions, with each having approximately 600 samples. Two sessions are used for training and constructing the UBM, while the remaining two sessions are used for testing. This yields a training set of 1199 utterances and a testing set of 1200 utterances. The length of each utterance varies from 2s to 2min, but most utterances have a length of approximately 3s. The average length is about 13s.

Voxceleb1 consists of 1251 celebrities' utterances with various acoustic environments and noises extracted from YouTube videos. The length of each utterance varies from 4s to 2min, and the average length is approximately 8s. The development set consists of 1211 celebrities' utterances, while the remaining 40 celebrities' utterances are used for testing. We select the first 100 celebrities' utterances in the development set for constructing the UBM, which contains 11090 utterances. The utterances of the celebrities with id10001 to id10050 in the development set are used for training, containing 5730 utterances. The testing set consists of the utterances of 40

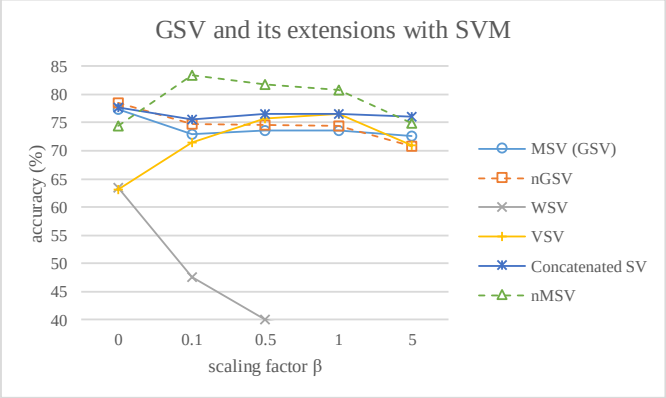


Fig. 1. Effectiveness of GSV and its extensions on Kingline081 speech corpus, employing SVM as the classifier.

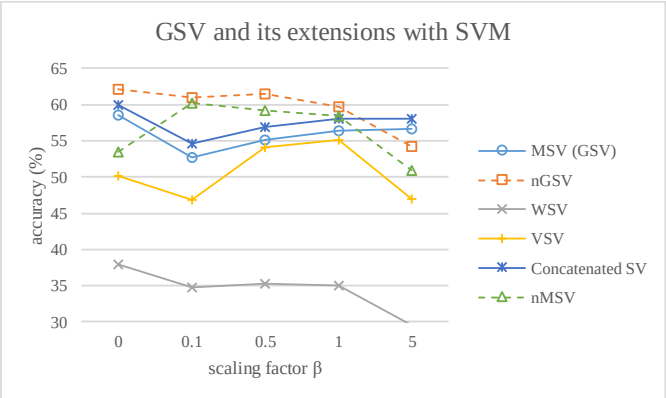


Fig. 2. Effectiveness of GSV and its extensions on Ahumada speech corpus, employing SVM as the classifier.

celebrities, containing 37720 pairs of utterances.

The GMM-based UBM consists of 128 mixture components for the speaker identification tasks and 64 mixture components for the speaker verification task, constructed using the mixture splitting technique [38]. The model parameters $\{V, \Psi\}$ for the i-vector are initialized to have ones on the principal diagonal and zeros otherwhere. For MFA with model parameters $\{w_m, u_m, W_m, \Sigma\}$, the mean parameter u_m is initialized to be the mean parameter μ_m of the GMM-based UBM, and the weight parameter w_m is initialized to be $1/M$. The factor-loading matrix W_m and the noise covariance Σ are initialized to have ones on the principal diagonal and zeros elsewhere. The frame-level feature vector is the MFCC vector [39] with a dimensionality of 20, extracted using the Hamming window with a frame length of 40ms and a frame shift of 20ms.

A. *GSV and Its Extensions for Speaker Identification*

In this part, we compare the speaker identification accuracy achieved by GSV and its extensions on the Kingline081 and Ahumada speech corpora. Linear SVM is employed as the classifier, which is implemented using LIBSVM [40]. The experimental results are shown in Fig. 1 and Fig. 2 (some accuracy values are too low to be shown). The dimensionality of GSV (MSV), nGSV and VSV is 2560×1, the dimensionality of WSV is 128×1, and the dimensionality of the concatenated MAP SV is 5248×1. Different values of the parameter β are used to construct the supervectors.

As shown in the figures, MSV tends to outperform WSV and VSV, reflecting that the adapted mean parameter plays the most important role. Among all, WSV gives the worst performance because its dimensionality is too low to embed enough information. In addition, the adapted weight may strongly depend on the content of the speech instead of the speaker's characteristics, which is undesirable. Nevertheless, combining the adapted weight and adapted standard deviation parameters into MSV may further improve its quality and robustness to different choices of β, as demonstrated by the concatenated MAP SV. The normalization operation can also be useful, as observed from the performance of nGSV and nMSV.

Another important observation is that the performance of the supervectors varies considerably with different choices of β. According to (8) and (19), the larger β is, the smaller the adaptation coefficient α_m is. As seen from (5) ~ (7), the adaptation coefficient controls the ratio between the information the supervector absorbs from the specific sample *s* and the information the supervector absorbs from the UBM. The smaller α_m is, the more the supervector will depend on the statistics of the UBM instead of the statistics of the specific sample *s*. A high dependence on the UBM leads to the high similarity between two supervectors. Thus, briefly speaking, a larger value of β causes the supervectors to be more similar to each other, which may degrade the performance of a discriminative classifier, such as SVM.

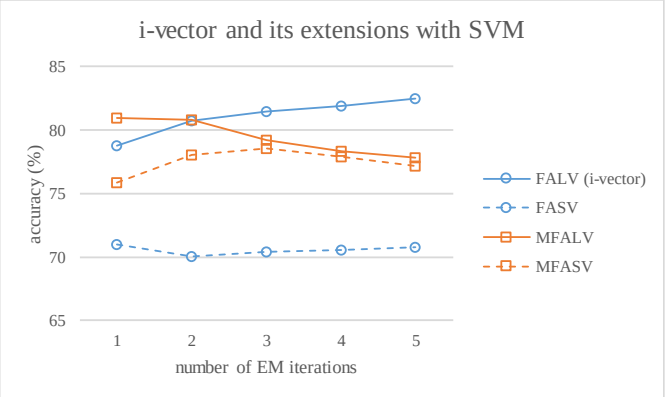


Fig. 3. Effectiveness of i-vector and its extensions on Kingline081 speech corpus, employing SVM as the classifier.

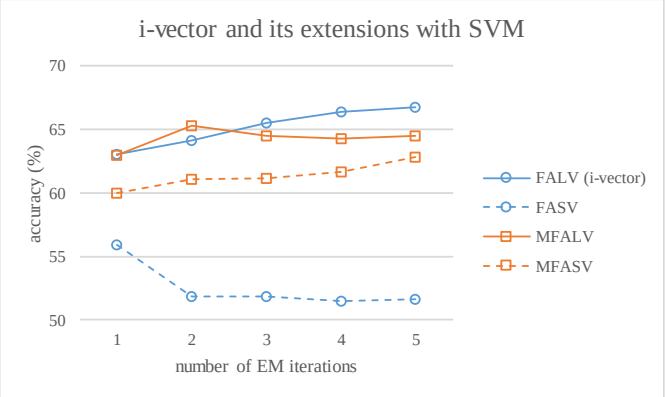


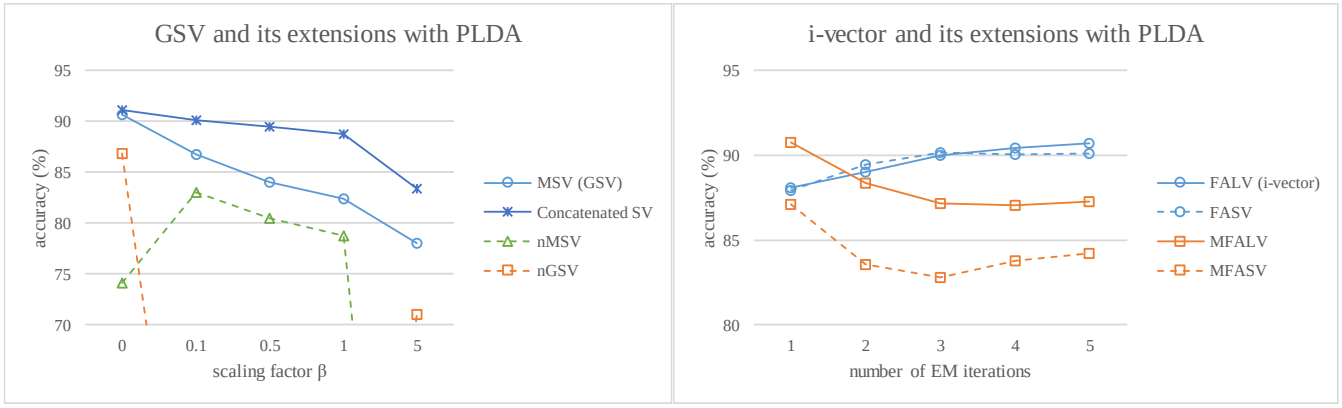
Fig. 4. Effectiveness of i-vector and its extensions on Ahumada speech corpus, employing SVM as the classifier.

B. *I-vector and Its Extensions for Speaker Identification*

In this part, we compare the speaker identification accuracy achieved by the i-vector and its extensions on the Kingline081 and Ahumada speech corpora. Linear SVM is employed as the classifier. The experimental results are shown in Fig. 3 and Fig. 4. The i-vector and its extensions are computed using different EM iterations to investigate the influence of this factor. The dimensionalities of FALV (i-vector), FASV, MFALV and MFASV are all 2560×1.

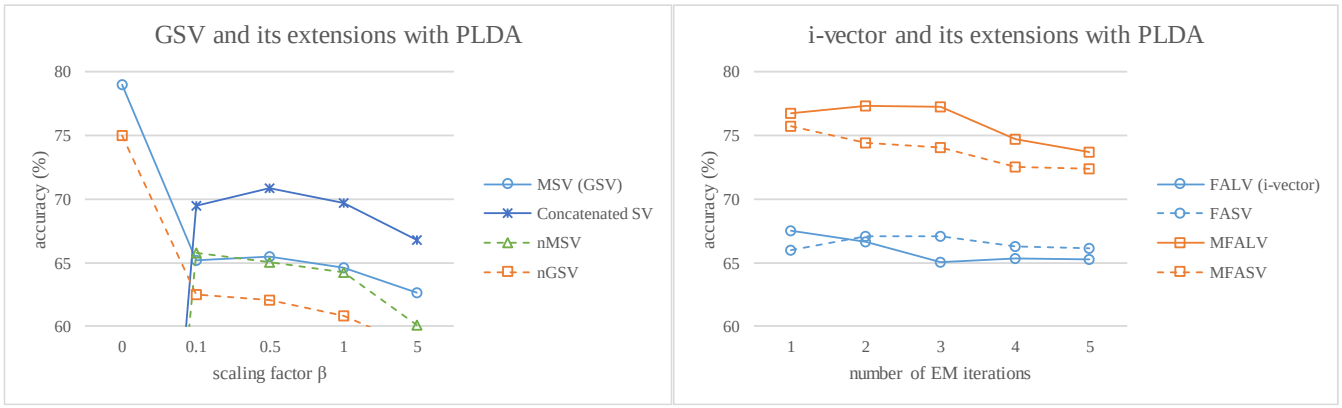
We see that the latent vectors (viz. FALV and MFALV) outperform their corresponding supervectors (viz. FASV and MFASV). This observation can be explained from the perspective of philosophy, that is, the supervector and the latent vector serve as the appearance and the essence of a sample, respectively. The latent vector is the cause of the supervector, and the supervector is the effect of the latent vector. From this angle, we may expect the latent vector to be cleaner.

The highest accuracy achieved by FALV is slightly better than MFALV, but the highest accuracy achieved by FASV is significantly worse than MFASV. In other words, the performance gap between FALV and FASV is larger than that between MFALV and MFASV. In fact, although the computation of FALV involves all the frame-level feature vectors in a sample, FASV is only related to FALV, which means that the relationship between FASV and individual frame-level feature vectors is weak. In contrast, MFALV is calculated as the weighted sum of the latent vectors of the



(a) (b)

Fig. 5. Effectiveness of GSV, i-vector and their extensions on Kingline081 speech corpus, employing PLDA as the classifier. (a) Results of GSV and its extensions. (b) Results of i-vector and its extensions.



(a) (b)

Fig. 6. Effectiveness of GSV, i-vector and their extensions on Ahumada speech corpus, employing PLDA as the classifier. (a) Results of GSV and its extensions. (b) Results of i-vector and its extensions.

frame-level feature vectors, and consequently, MFASV, as an affine transformation of the latent vectors, is strongly related to individual frame-level feature vectors. In addition, the mean parameter used in computing FASV is fixed to be the mean parameter of the GMM-based UBM, whereas that used in computing MFASV is the mean parameter of the MFA-based UBM, which is re-estimated based on the model assumption.

It is also noted that increasing the number of EM iterations can improve the performance of FALV but not MFALV. This effect probably results from the model assumptions of the two methods. For FALV, the latent vector is assumed to be shared among all the frame-level feature vectors, aiming to capture common characteristics among different frames. While for MFALV, the individual frame-level feature vectors are assumed to have their own latent vectors, aiming to capture variation across different frames.

C. Effectiveness and Efficiency of Different Feature Representations for Speaker Identification

In this part, we compare the performance and the computation time when using GSV, the i-vector and their extensions for speaker identification. The scalable PLDA [41] is employed for performing the classification. The latent vectors in the PLDA model have the same dimensionality as the feature representations, and the parameters of the PLDA model are estimated using 2 EM iterations.

The experimental results on speaker identification accuracy are shown in Fig. 5 and Fig. 6 (some accuracy values are too low to be shown), and the dimensionalities of different feature representations are 2560×1, except for the concatenated MAP SV, whose dimensionality is 5248×1.

As shown in Fig. 5 and Fig. 6, GSV and its extensions are notably sensitive to the value of β, especially for nGSV and nMSV. Actually, PLDA is also an FA model that assumes the feature representation to follow Gaussian distributions. Since the scaling factor β controls how much information the GSV absorbs from the UBM, it seems to control how much the GSV will fulfill the model assumptions made by PLDA. Nevertheless, with a suitable value of β, the highest accuracy achieved by GSV is comparable to (Fig. 5) or even better than the i-vector (Fig. 6) when PLDA is employed as the classifier. From another perspective, as the i-vector can be treated as an affine transformation of GSV, this transformation may cause the loss of information carried by the i-vector.

It is also observed that the highest accuracy achieved by MFALV is comparable to (Fig. 5) or even better than the i-vector (Fig. 6) when PLDA is employed as the classifier. MFALV adopts a more complicated model assumption for each frame-level feature vector, which may be beneficial if a

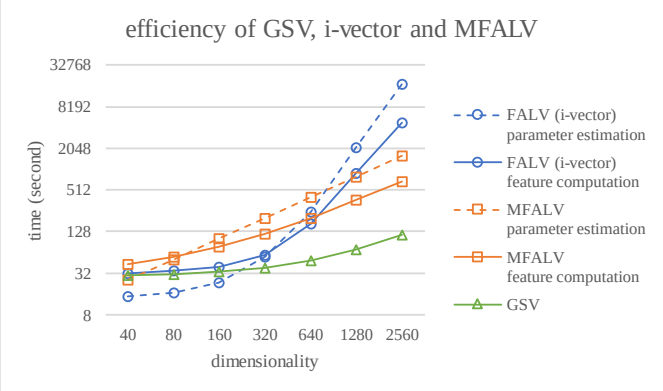


Fig. 7. Efficiency of GSV, i-vector and MFALV on Kingline081.

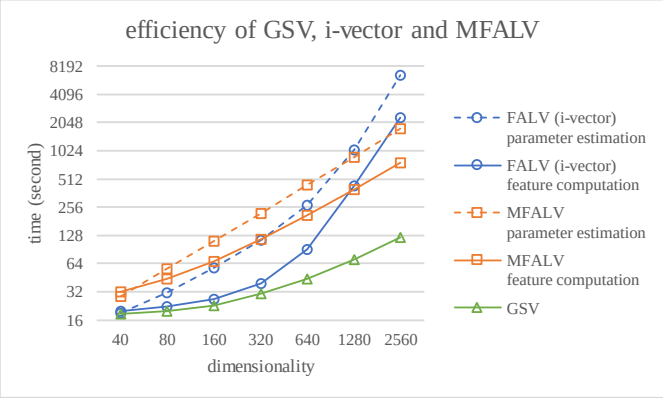


Fig. 8. Efficiency of GSV, i-vector and MFALV on Ahumada.

frame-level feature vector indeed carries abundant information. This abundance may occur when the speech signal is contaminated by session variability. We also notice the different trends of MFALV and the i-vector on different datasets, which is probably due to the different characteristics of the two datasets. Kingline081 (Fig. 5b) contains phonetically rich speech uttered at normal speed, while Ahumada (Fig. 6b) contains some text-independent content and text-dependent content, which has less phonetic information. In addition, the speech is uttered at varying speeds; thus, there may be large variation across frames. MFALV assumes that each frame-level feature vector has its own latent vector, while the i-vector assumes that all the frame-level feature vectors share the same latent vector; therefore, the former may better describe the characteristics of individual frame-level feature vectors in the situation of varying speeds.

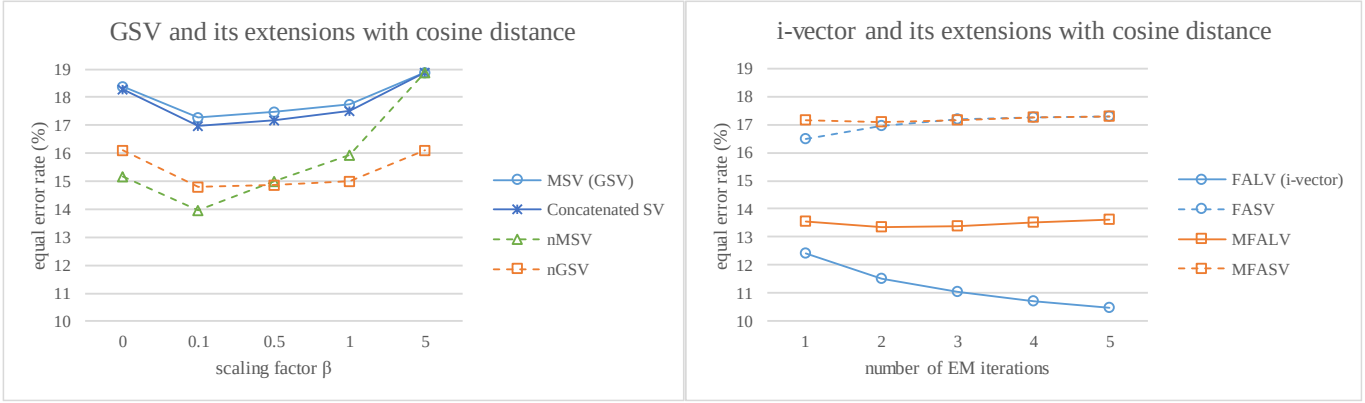
The computation time of GSV, the i-vector and MFALV is shown in Fig. 7 and Fig. 8, where the consumed time is estimated by running the MATLAB codes on an iMac desktop computer with 32G memory. For the i-vector and MFALV, time is consumed in the stage of parameter estimation and the stage of feature computation, whereas for GSV, there is only the stage of feature computation. The model parameters of the i-vector and MFALV are estimated with 1 EM iteration, and GSV is computed with β=0.1. The number of mixture components in the UBM varies from 2 to 128, and thereby the dimensionality of GSV, the i-vector and MFALV varies from 40×1 to 2560×1.

As shown in Fig. 7 and Fig. 8, the computation time of the i-vector is notably short at a low dimensionality but becomes increasingly longer with increasing dimensionality. When the dimensionality is high, MFALV is more efficient than the i-vector. This observation is consistent with our theoretical analysis on the time complexity of MFALV and the i-vector. The computation of GSV is always the fastest because of its simplicity in computation and the avoidance of additional parameter estimation. Nonetheless, the scaling factor β has to be chosen carefully, and the dimensionality of GSV is not as flexible as the i-vector and MFALV.

D. Performance of Different Feature Representations for Speaker Verification

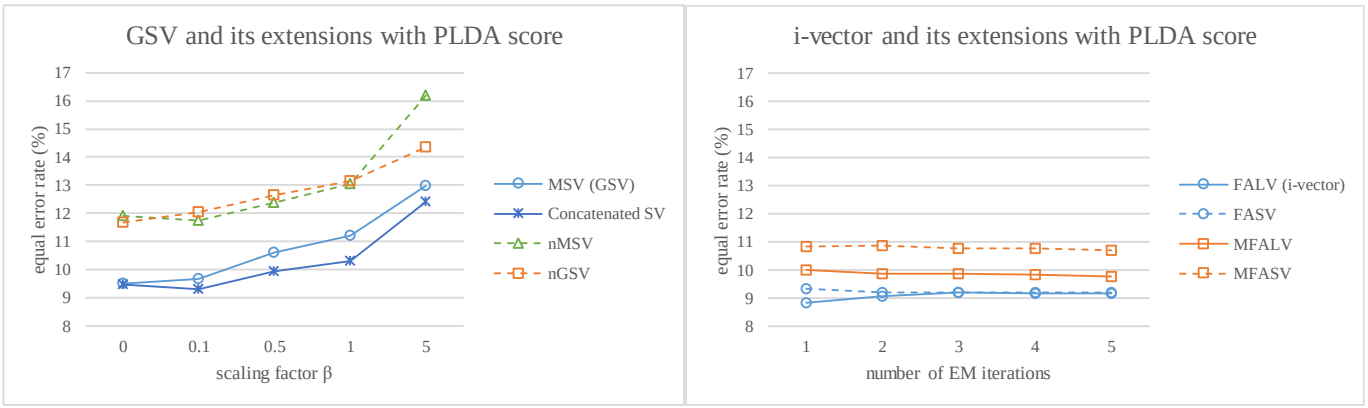
In this part, we compare the performance of GSV, the i-vector and their extensions on speaker verification. The performance is evaluated in terms of the equal error rate (EER), which is generally the lower the better. The cosine distance [10] and the PLDA model [47] are used to generate the verification score. The latent vectors in the PLDA model have the same dimensionality as the feature representations, and the parameters of the PLDA model are estimated using 1 EM iteration.

The experimental results on speaker verification are shown in Fig. 9 and Fig. 10. The dimensionality of different feature representations is 1280×1, except for the concatenated MAP



(a) (b)

Fig. 9. EER achieved by GSV, i-vector and their extensions on Voxceleb1, using cosine distance score. (a) Results of GSV and its extensions. (b) Results of i-vector and its extensions.



(a) (b)

Fig. 10. EER achieved by GSV, i-vector and their extensions on Voxceleb1, using PLDA score. (a) Results of GSV and its extensions. (b) Results of i-vector and its extensions.

SV, whose dimensionality is 2624×1.

As shown in Fig. 9a and Fig. 10a, nGSV and nMSV perform better than GSV and the concatenated SV when cosine distance is used for scoring, but perform worse when PLDA is used for scoring. This observation, together with the discussions in previous parts, implies that the normalization operation takes effect only when the backend is linear, such as a linear SVM or the cosine distance measure. Nevertheless, PLDA scoring is considerably better than cosine distance scoring. It is also notable that β strongly affects the performance of GSV and its extensions, and a small value of β is preferred, because the smaller β is, the more discriminative the feature representation will be. The concatenated SV outperforms GSV regardless of which scoring method is used, demonstrating that other adapted parameters provide complementary information to the adapted mean parameter.

As shown in Fig. 9b and Fig. 10b, FALV and MFALV significantly outperform their corresponding supervectors (i.e., FASV and MFASV) when cosine distance is used for scoring, but the performance gap is narrowed when PLDA is used for scoring, as PLDA exploits the relationship between the variables in a feature representation. It is noticed that, FALV works well using both scoring methods, while MFALV approaches FALV's effectiveness only when PLDA scoring is used. This observation demonstrates the robustness of FALV.

As seen from Fig. 9 and Fig. 10, FALV gives the best EER among all, but GSV, the concatenated SV and MFALV are competitive when a powerful backend, such as the PLDA model, is used for scoring.

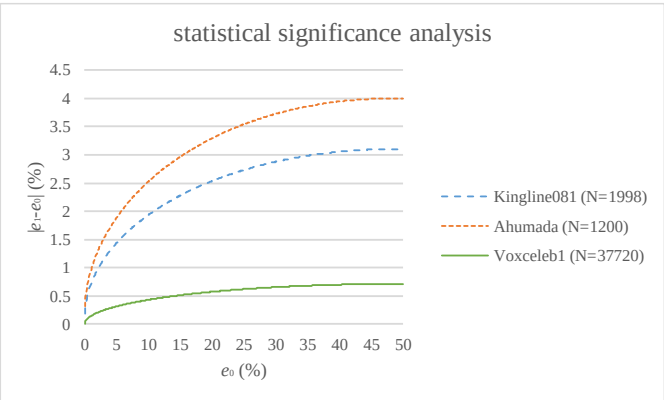


Fig. 11. Statistical significance analysis on different datasets.

E. Statistical Significance of Performance Difference

In this part, we briefly analyze the statistical significance of the performance difference when using different feature representations. The performance is measured using the error rate e, which is $(1-accuracy)$ for speaker identification tasks, or approximately the EER for speaker verification tasks. We adopt the method in [48]. Let $(1-\alpha)$ be a confidence level ($0<\alpha<1$), e_0 and e_1 be the two error rates, and N be the number of training samples. According to [48], if $|e_1-e_0| \geq z_{\alpha/2}\sqrt{\left(e_0(1-e_0)+e_1(1-e_1)\right)/N}$, where $z_{\alpha/2}$ is a value related to α, then the difference between e_0 and e_1 is statistically significant with a confidence level of $(1-\alpha)$. The lower bound for $|e_1-e_0|$ to be statistically significant with a confidence level of 95% is plotted with respect to different values of e_0 (the baseline) for different datasets in Fig. 11. More details about the lower bound are given in the appendix.

On Kingline081 (Fig. 5), the lowest error rates achieved by GSV, the i-vector and MFALV are 9.36%, 9.31% and 9.21%, respectively. According to Fig. 11, the difference between these error rates is not statistically significant. On Ahumada (Fig. 6), the lowest error rates achieved by GSV, the i-vector and MFALV are 21%, 32.5% and 22.67%, respectively. According to Fig. 11, the difference between GSV and MFALV is not statistically significant, while the difference between GSV and the i-vector and the difference between the i-vector and MFALV are statistically significant with a confidence level of 95%. On Voxceleb1 (Fig. 10), the lowest EERs achieved by GSV, the i-vector and MFALV are 9.51%, 8.83% and 9.77%, respectively. According to Fig. 11, the difference between GSV and MFALV is not statistically significant, while the difference between GSV and the i-vector and the difference between the i-vector and MFALV are statistically significant with a confidence level of 95%. In some sense, statistical insignificance implies that the performance of different feature representations is similar.

VI. Conclusion and Inspiration

In this paper, we analyze two popular vector-based feature representations, viz. GSV and the i-vector, from the angle of their construction mechanisms and underlying rationales. Through analyzing the formulation, we make extensions for GSV and the i-vector, introducing new types of feature representations as alternatives to GSV and the i-vector.

GSV is the concatenation of the adapted mean parameters of a GMM-based UBM. This inspires us to extend GSV to MAP SV, which can be the concatenation of the adapted weight parameters, adapted mean parameters or the adapted standard deviation parameters. Concatenating all these adapted parameters produces a concatenated SV, whose dimensionality is even higher than GSV and thus carries richer information.

The i-vector is the posterior expectation of the latent vector in an FA model. It assumes the latent vector to be shared by all the frame-level feature vectors. This renders the i-vector quite robust, but the computational burden can be high if the dimensionality of the latent vector is high. By introducing MFA as the UBM, we propose the MFALV, which assumes that each frame-level feature vector has its own latent vector. This results in a much lower dimensionality of the latent vectors, making the computation more efficient. Experimental results demonstrate that the performance of MFALV is comparable to or even better than the i-vector, and maintains a lower computational complexity. Nevertheless, the dimensionality of MFALV is not as flexible as the i-vector.

As a brief comparison, GSV is based on the parameters of a GMM-based UBM, the i-vector is based on the parameters of a GMM-based UBM and the parameters of an FA, and MFALV is based on the parameters of an MFA-based UBM, whose parameters are initialized from a GMM-based UBM. These feature representations are actually closely related in concept, where UBM plays an important role. Because of the different mechanisms in computation, GSV is the fastest among all, but its dimensionality is not as flexible as the i-vector and MFALV. GSV is a supervector (SV), while the i-vector and MFALV are latent vectors (LV). From a philosophical angle, SV and LV can be regarded as playing the role of appearance and essence of an acoustic sample, respectively. This interpretation may help explain their different behaviors under different backends.

The concept of SV is general and extensible. For example, MAP SV is obtained by concatenating the MAP adapted parameters of the UBM. In fact, the concatenated parameters may not necessarily be based on MAP adaptation. Other types of adaptation or operation on the UBM are feasible, such as calculating the gradients [43]. In addition, the UBM can have different choices, such as GMM, MFA, DBN or DNN. The UBM can even be the combination of different models, such as the combination of GMM and MFA, or the combination of GMM and DBN. This kind of combination is feasible because SV is obtained by simply concatenating the individual adapted parameters, and different adapted parameters are not entangled. Interestingly, FA or MFA or other types of latent factor analysis can be applied to SV to produce various types of LV, which then leads to a large variety of new feature representations.

References

[1] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19-41, 2000.

[2] D. A. Reynolds, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 1, pp. 72-83, 1995.

[3] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Trans. on Multimedia*, vol. 17, no. 10, pp. 1733-1746, 2015.

[4] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308-311, 2006.

[5] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: a tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74-99, 2015.

[6] D. Garcia-Romero and C. Y. Espy-Wilson, "Automatic acquisition device identification from speech recordings," in *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP)*, 2010, pp. 1806-1809.

[7] Y. Jiang and F. H. F. Leung, "Source microphone recognition aided by a kernel-based projection method," *IEEE Trans. on Information Forensics and Security*, vol. 14, no. 11, pp. 2875-2886, 2019.

[8] Y. Li *et al.*, "Mobile phone clustering from speech recordings using deep representation and spectral clustering," *IEEE Trans. on Information Forensics and Security*, vol. 13, no. 4, pp. 965-977, 2018.

[9] L. Jing *et al.*, "DCAR: a discriminative and compact audio representation for audio processing," *IEEE Trans. on Multimedia*, vol. 19, no. 12, pp. 2637-2650, 2017.

[10] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788-798, 2011.

[11] L. Ferrer and M. McLaren, "Joint plda for simultaneous modeling of two factors," *Journal of Machine Learning Research*, vol. 20, no. 24, pp. 1-29, 2019.

[12] V. Vestman, B. Soomro, A. Kanervisto, V. Hautamaki, and T. Kinnunen, "Who do I sound like? Showcasing speaking recognition technology by youtube voice search," in *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP)*, 2019, pp. 5781-5785.

[13] A. Mesaros *et al.*, "Detection and classification of acoustic scenes and events: outcome of the dcase 2016 challenge," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379-393, 2018.

[14] S. Chandrakala and S. L. Jayalakshmi, "Generative model driven representation learning in a hybrid framework for environmental audio scene and sound event recognition," *IEEE Trans. on Multimedia*, vol. 22, no. 1, pp. 3-14, 2020.

[15] Y. Li, M. Liu, W. Wang, Y. Zhang, and Q. He, "Acoustic scene clustering using joint optimization of deep embedding learning and clustering iteration," *IEEE Trans. on Multimedia*, vol. 22, no. 6, pp. 1385-1394, 2020.

[16] D. Roy, K. S. R. Murty, and C. K. Mohan, "Unsupervised universal attribute modeling for action recognition," *IEEE Trans. on Multimedia*, vol. 21, no. 7, pp. 1672-1680, 2019.

[17] N. Perveen, D. Roy, and C. K. Mohan, "Spontaneous expression recognition using universal attribute model," *IEEE Trans. on Image Processing*, vol. 27, no. 11, pp. 5575-5584, 2018.

[18] N. Inoue and K. Shinoda, "A fast and accurate video semantic-indexing system using fast map adaptation and GMM supervectors," *IEEE Trans. on Multimedia*, vol. 14, no. 4, pp. 1196-1205, 2012.

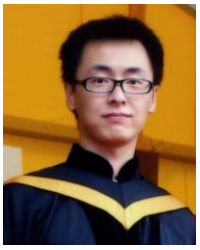
[19] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12-40, 2010.

[20] W. M. Campbell, "Using deep belief networks for vector-based speaker recognition," in *Proc. INTERSPEECH*, 2014, pp. 676-680.

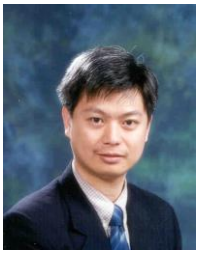
[21] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP)*, 2014, pp. 1695-1699.

[22] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. Int. Conf. on Acousitc, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052-4056.

[23] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. INTERSPEECH*, 2017, pp. 999-1003.
[24] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: robust dnn embeddings for speaker recognition," in *Proc. Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329-5333.
[25] L. Xu, K. A. Lee, H. Li, and Z. Yang, "Generalizing i-vector estimation for rapid speaker recognition," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 26, no. 4, pp. 749-759, 2018.
[26] V. Vestman and T. Kinnunen, "Supervector compression strategies to speed up i-vector system development," in *Proc. Odyssey*, 2018, pp. 357-364.
[27] P. Li, Y. Fu, U. Mohammed, J. H. Elder, and S. J. D. Prince, "Probabilistic models for inference about identity," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 144-157, 2012.
[28] L. Liu, C. Xiong, H. Zhang, Z. Niu, M. Wang, and S. Yan, "Deep aging face verification with large gaps," *IEEE Trans. on Multimedia*, vol. 18, no. 1, pp. 64-75, 2016.
[29] M. E. Wibowo, D. Tjondronegoro, V. Chandran, R. Pulungan, and J. E. Istiyanto, "Improved face recognition across poses using fusion of probabilistic latent variable models," *Telkomnika*, vol. 15, no. 4, pp. 1976-1986, 2017.
[30] G. Sell, K. Duh, D. Snyder, D. Etter, and D. Garcia-Romero, "Audio-visual person recognition in multimedia data from the IARPA Janus program," in *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP)*, 2018, pp. 3031-3035.
[31] S. E. Shepstone, Z. H. Tan, and S. H. Jensen, "Using audio-derived affective offset to enhance tv recommendation," *IEEE Trans. on Multimedia*, vol. 16, no. 7, pp. 1999-2010, 2014.
[32] M. N. Do, "Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models," *IEEE Signal Processing Letters*, vol. 10, no. 4, pp. 115-118, 2003.
[33] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 3, pp. 345-354, 2005.
[34] C. M. Bishop, "Continuous latent variables," in *Pattern Recognition and Machine Learning*, Springer, 2006, ch. 12, pp. 559-603.
[35] Z. Ghahramani and G. E. Hinton, "The EM algorithm for mixtures of factor analyzers," *Technical Report CRG-TR-96-1*, University of Toronto, 1996.
[36] KingLine Data Center, American English Speech Recognition Corpus (King-ASR-L-081), Speechocean, 2013.
[37] J. O. Garcia, J. G. Rodriguez, and V. M. Aguiar, "AHUMADA: a large speech corpus in Spanish for speaker characterization and identification," *Speech Communication*, vol. 31, no. 2, pp. 255-264, 2000.
[38] Y. Jiang and F. H. F. Leung, "A class-dependent background model for speech signal feature extraction," in *Proc. IEEE Int. Conf. on Digital Signal Processing (DSP)*, 2018.
[39] X. Huang, A. Acero, and H. W. Hon, "Speech signal representations," in *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Upper Saddle River, NJ: Prentice Hall PTR, 2001, ch. 6, pp. 273-333.
[40] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1-27, 2011.
[41] Y. Jiang and F. H. F. Leung, "The scalable version of probabilistic linear discriminant analysis and its potential as a classifier for audio signal classification," in *Proc. IEEE Int. Joint Conf. on Neural Networks (IJCNN)*, 2018, pp. 1-7.
[42] S. Cumani and P. Laface, "E-vectors: JFA and i-vectors revisited," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5435-5439.
[43] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1-8.
[44] H. Yamamoto, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura, "Parameter sharing in mixture of factor analyzers for speaker identification," *IEICE Trans. on Information and Systems*, vol. 88, no. 3, pp. 418-424, 2005.
[45] T. Hasan and J. H. Hansen, "Acoustic factor analysis for robust speaker verification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 842-853, 2013.
[46] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," in *Proc. INTERSPEECH*, 2017, pp. 2616-2620.
[47] Y. Jiang, K. A. Lee, Z. Tang, B. Ma, A. Larcher, and H. Li, "PLDA modeling in i-vector and supervector space for speaker verification," in *Proc. INTERSPEECH*, 2012, pp. 1680-1683.
[48] P. N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, "Classification: basic concepts and techniques," in *Introduction to Data Mining*, 2nd Edition. Pearson, 2018, ch. 3.
[49] C. H. You, K. A. Lee, and H. Li, "An SVM kernel with GMM-supervector based on the Bhattacharyya distance for speaker recognition," *IEEE Signal Processing Letters*, vol. 16, no. 1, 2009.
[50] M. H. Bahari *et al.*, "Non-negative factor analysis of Gaussian mixture model weight adaptation for language and dialect recognition," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 22, no. 7, pp. 1117-1129, 2014.
[51] C. H. You, H. Li, E. Ambikairajah, K. A. Lee, and B. Ma, "Bhattacharyya-based GMM-SVM system with adaptive relevance factor for pair language recognition," in *Proc. Odyssey*, 2012.



Yuechi Jiang (S'18) received the BEng degree in electronic engineering from the Chinese University of Hong Kong in 2015. He is currently working toward the PhD degree in the Department of Electronic and Information Engineering, the Hong Kong Polytechnic University. His research interests include acoustic signal processing and pattern recognition.



Frank H. F. Leung (M'92–SM'03) received the BEng and the PhD degrees in electronic engineering from the Hong Kong Polytechnic in 1988 and 1992 respectively. He is now Associate Professor in the Department of Electronic and Information Engineering, the Hong Kong Polytechnic University. He is an active researcher who has published over 210 research papers on computational intelligence, machine learning, control, and power electronics. At present, he is involved in the R&D on Intelligent Signal Processing, Systems, and Robotics. He has been serving as editor, guest editor and reviewer for international journals, and helping the organization of many international conferences. He is a Chartered Engineer, a corporate member of the Institution of Engineering and Technology (U.K.) and the Hong Kong Institution of Engineers.