Accurately Capturing Speech Feature Distributions by Extending Supervectors for Robust Speaker Recognition

Kevin Wilkinghoff

Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE
Fraunhoferstraße 20, 53343 Wachtberg, Germany
Email: kevin.wilkinghoff@fkie.fraunhofer.de

Abstract

Supervectors represent speaker-specific Gaussian Mixture Models which are enrolled from a Universal Background Model (UBM) and approximate the unknown, underlying speech feature distributions. But as supervectors only consist of the stacked means of the Gaussian components, low-dimensional i-vectors which are derived from them do not completely capture the true feature distributions. In this work, the classical supervectors are extended with additional parameters before reducing their dimension to capture the feature distributions more accurately and complement the i-vectors more effectively. To extend a supervector, the mixture weights, the log-likelihood values of the UBM, a Bhattacharyya-distance based kernel and the Hellinger distance between each enrolled Gaussian component and the corresponding one of the UBM are used. In closed-set speaker identification experiments conducted on the NTIMIT corpus which consists of telephone quality speech, the extended supervectors provide significantly lower error rates than the standard supervectors, even after fusing them with i-vectors and the UBM.

1 Introduction

The first step when recognizing speakers is to extract many short-time features from given speech segments. Most commonly, Mel-Frequency Cepstral Coefficients (MFCCs) [1] or Perceptual Linear Prediction (PLP) [2] features are used for that purpose. Naturally, these features vary heavily from frame to frame especially when considering noisy or low-quality speech. To obtain a single speaker-dependent mathematical object which is robust to these variations, the unknown distributions of the features are approximated with Gaussian Mixture Models (GMMs) obtained by enrolling a Universal Background Model (UBM) [3]. This UBM is a GMM trained on features extracted from a large amount of unlabeled speech data called "development set". Supervectors serve as a compact representation of the underlying feature distributions and are obtained by stacking the means of the Gaussian components of the speaker-specific GMMs. As these supervectors are very high-dimensional and thus are difficult to handle effectively, the next step is to reduce their dimension with the speaker-independent i-vector model [4]. Probabilistic Linear Discriminant Analysis (PLDA) [5] is applied afterwards to split the i-vectors into speaker- and channel-dependent components of low dimension.

Clearly, the supervectors do not completely capture the feature distributions as they only consist of the means of the Gaussian components. When extracting the i-vectors, the log-likelihood of the mixture components given the data and therefore also their weights and covariance matrices come into play but still some information about the feature distribution is inherently lost. Hence, it is apparent that i-vectors are certainly good but not optimal representations of given utterances.

Prior to the rise of i-vectors other attempts using all the parameters of the enrolled GMMs and UBM have been made to deal with the high-dimensional supervectors. Most prominently, Support Vector Machines (SVMs) have been applied with kernels based on the Kullback-Leibler (KL) divergence [6] and the closely related Bhattacharyya Distance [7, 8]. But usually, the combination of i-vector and PLDA tends to result in a much better performance. Also, directly comparing the enrolled GMMs with f-divergences does not lead to lower error rates [9, 10].

To incorporate all the relevant information about the feature distributions into the supervector, we propose to extend the supervectors by concatenating the classical ones with a) the weights of the Gaussian components, b) the log-likelihood values of the UBM's mixture components, c) a Bhattacharyya-distance based SVM kernel and d) the Hellinger distance of each enrolled Gaussian component to the corresponding one of the UBM. In Section 2, the resulting supervectors' structure is described in detail. The dimension of these extended supervectors is reduced with PCA and the computational requirements are compared to those of i-vectors. Furthermore, PLDA will be used afterwards because this greatly increases the recognition accuracy. As shown in closed-set speaker identification experiments conducted on the NTIMIT corpus in Section 3, the extended supervectors are greatly improving upon the standard supervectors in terms of identification error rate. Moreover, combining them with i-vector and UBM leads to a further reduction in error rate over the standard supervectors.

2 Extending the Supervectors

2.1 Standard Supervectors

Before reviewing the definition of standard supervectors, the notation used throughout the paper will be presented. Let $\lambda_{\mathrm{U}} := (w_m^U, \mu_m^U, \Sigma_m^U)_{m=1,\dots,M}$ denote the parameters of a UBM with $M \in \mathbb{N}$ Gaussian components for a feature dimension of $D \in \mathbb{N}$. Here, $w_m^U \in [0,1]$ is the mixture weight, $\mu_m^U \in \mathbb{R}^D$ the mean and $\Sigma_m^U \in \mathbb{R}^{D\times D}$ the diagonal covariance matrix of component $m \in \{1,\dots,M\}$. Furthermore, let $\mathscr{S}$ be a finite set of speakers. For each speaker $S \in \mathscr{S}$, $\lambda_S := (w_m^S, \mu_m^S, \Sigma_m^S)_{m=1,\dots,M}$ denotes the parameters of the GMM obtained by enrolling the UBM λ_U.

Classically, the normalized supervector corresponding to speaker $S \in \mathscr{S}$ is defined as

$$\overline{\mathfrak{V}}_S := \begin{pmatrix} (\Sigma_1^U)^{-1}(\mu_1^S - \mu_1^U) \\ \vdots \\ (\Sigma_M^U)^{-1}(\mu_M^S - \mu_M^U) \end{pmatrix} \in \mathbb{R}^{DM}.$$

It is immediately visible that not all parameters of the Gaus-

sian Mixture Models are contained in the supervectors and therefore some information about the underlying speech feature distributions is inherently missing.

2.2 Extended Supervectors

The extended supervector for any given speaker $S \in \mathscr{S}$ is defined as

$$\mathfrak{X}_S := \begin{pmatrix} \mathfrak{V}^S \\ w^S \\ \log \mathrm{P}(X^S|\lambda^U) \\ \mathrm{BCK}(\lambda^S, \lambda^U) \\ \mathrm{H}(\lambda^S, \lambda^U) \end{pmatrix} \in \mathbb{R}^{(3D+4)M}$$

where $X^S \in \mathbb{R}^{D\times N}$ for some $N \in \mathbb{N}$ denotes the features used for enrolling speaker S and $\mathfrak{V}^S \in \mathbb{R}^{DM}$ denotes the unnormalized supervector, i.e. just the stacked means of all Gaussian components. Furthermore, $w^S \in \mathbb{R}^M$ consists of all the mixture weights and $\log \mathrm{P}(X^S|\lambda^U) \in \mathbb{R}^M$ contains the stacked log-likelihood values of the UBM's components given the enrollment data X^S.

The supervector and weights have not been normalized with respect to the UBM because the normalized versions also appear in the Bhattacharyya distance based SVM kernel. Thus, cases can be found where the unnormalized entries contain complementing information whereas normalizing them ensures that the entries are redundant. In any case, applying PCA to the extended supervectors will only preserve indispensable information which allows us to add both, the normalized and unnormalized entries without harming the performance.

The last two entries of the extended supervector will now be described in detail. The first one is the extended SVM kernel based on the Bhattacharyya distance given in [8] which has been chosen because it performed best among the kernels tested by the authors. For mixture component $m \in \{1, ..., M\}$, it is defined as

$$\mathrm{BCK}_m(\lambda^S, \lambda^U) := \begin{pmatrix} \left(\frac{\Sigma_m^S + \Sigma_m^U}{2}\right)^{\frac{1}{2}} (\mu_m^S - \mu_m^U) \\ \mathrm{diag}\left(\left(\frac{\Sigma_m^S + \Sigma_m^U}{2}\right)^{\frac{1}{2}} \left(\Sigma_m^S\right)^{-\frac{1}{2}}\right) \\ \frac{w_m^U}{w_m^S} \end{pmatrix}.$$

and appended to the supervector via

$$\mathrm{BCK}(\lambda^S, \lambda^U) := \begin{pmatrix} \mathrm{BCK}_1(\lambda^S, \lambda^U) \\ \vdots \\ \mathrm{BCK}_M(\lambda^S, \lambda^U) \end{pmatrix} \in \mathbb{R}^{(2D+1)M}.$$

The Hellinger distance (see e.g. [11]) for a single Gaussian mixture component $m \in \{1, ..., M\}$ can be computed as

$$\mathrm{H}_m(\lambda^S, \lambda^U) := \Bigg[1 - \frac{\det(\Sigma_m^S)^{\frac{1}{4}} \det(\Sigma_m^U)^{\frac{1}{4}}}{\det\left(\frac{\Sigma_m^U + \Sigma_m^S}{2}\right)^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{8}(\mu_m^S - \mu_m^U)^\top \left(\frac{\Sigma_m^S + \Sigma_m^U}{2}\right)^{-1} (\mu_m^S - \mu_m^U)\right)\Bigg]^{\frac{1}{2}}.$$

These distances are added to the supervector by setting the last entry to

$$\mathrm{H}(\lambda^S, \lambda^U) := \begin{pmatrix} \mathrm{H}_1(\lambda^S, \lambda^U) \\ \vdots \\ \mathrm{H}_M(\lambda^S, \lambda^U) \end{pmatrix} \in \mathbb{R}^M.$$

The Hellinger distance is a bounded metric, opposed to the Bhattacharyya distance (BC) (see e.g. [7]) which does not obey the triangular inequality, and therefore induces more structure on the space of probability distributions. This is the reason why it as also been added to the supervector although we already added a kernel based on the Bhattacharyya distance and both distances are strongly connected through the identity

$$\mathrm{H}(\lambda^S, \lambda^U) = \sqrt{1 - \mathrm{BC}(\lambda^S, \lambda^U)}.$$

Note that BC denotes the Bhattacharyya distance and not the kernel BCK from above. Furthermore, the kernel is derived from an upper bound of the Bhattacharyya distance and not the distance itself. This may be another reason that the kernel behaves differently than the Hellinger distance and using both of them ensures that all information is covered.

As a last preprocessing step, the extended supervector is standardized by subtracting the mean and dividing by the standard deviation of the extended supervectors obtained from the development data. The reason for this is that the components have entirely different scales leading to problems when applying PCA which requires similarly scaled variables (see [12]). As a side effect, the dimensionally reduced, extended supervectors do not need to be "Gaussianized" i.e. projected to the unit sphere before applying PLDA which needs to be done when using i-vectors.

2.3 Comparing Computational Complexities

By definition (and as the name already indicates), the dimension of the extended supervector is much higher than the dimension of the standard supervector. More precisely, it is $(3D+4)M$ instead of DM i.e. the dimension is more than three times higher and thus it is clear that the runtime also increases. Still, using extended supervectors with PCA has a lower computational complexity in terms of runtime than using i-vectors as we will show now.

Let $R \in \mathbb{N}$ denote the reduced target dimension of applying PCA as well as the dimension of the i-vectors and $N \in \mathbb{N}$ the number of supervectors used for training both models. When stating the computational complexities, we will make use of the fact that usually $3D + 4 < R$ (typical values are $R = 400, D = 60$) and thus MR serves as an upper bound of the extended supervector's size. Therefore, the computational complexity of extracting a single i-vector is $\mathscr{O}(MDR + MDR^2 + R^3)$ (see [13])[1] whereas reducing the dimension via PCA is a simple matrix multiplication with a computational complexity of $\mathscr{O}(MR^2)$. However, we also need to take into account that the components of the extended supervector have to be computed. But as we assumed diagonal covariance matrices, all operations for a single component of the extended supervectors can be executed in $\mathscr{O}(D)$ or even $\mathscr{O}(1)$. Thus, $\mathscr{O}(MDR)$ is needed for calculating all components. This leads to a total

[1] In the paper, the computational complexity is incorrectly stated as $\mathscr{O}(MDR + MR^2 + R^3)$ which is probably just a typo. Nevertheless, the given argumentation still holds.

computational complexity of $\mathcal{O}(MDR + MR^2)$ for computing the extended supervectors and reducing their dimension. Hence, extracting extended supervectors and applying PCA is indeed much faster than extracting i-vectors. Although it is possible to lower the computational costs (see e.g. [13]), doing so sacrifices some performance because this slightly increases the resulting error rates.

Training the models can be done offline and needs to be done only once. Therefore, the computational complexities for computing the models are not that important, but from our experience training the PCA model is also much faster than training the i-vector model.

3 Experiments

3.1 Experimental Setup

The NTIMIT corpus [14] consists of speech sent over telephone channels of 630 speakers each with 10 utterances of about 3 seconds length. For each speaker, the 5 phonetically compact "SX" and 3 phonetically diverse "SI" utterances were labeled as training data and the other 2 remaining "SA" utterances, which are the same two sentences spoken by each speaker, were used for testing. Furthermore, we downsampled all data from 16 kHz to 8 kHz to save memory. Since the signals are band-limited anyway, this does not degrade the performance.

As features, 19 dimensional MFCCs as well as PLP coefficients of the same dimension have been extracted from the utterances by using the HTK toolkit [15]. For calculating the features, Hamming-weighted frames with a length of 25 ms and an overlap of 10 ms have been used. In addition to that, Spectral Subband Centroids (SSCs) [16], Glottal Mixture Model (GLOMM) [17, 18] and pitch features have been evaluated to show that extending the supervector is beneficial regardless of the speech features being used. The algorithm that has been used to extract the pitch features is the following: First, the signal is divided into Hanning-weighted frames and the highest peak $r_\tau \in \mathbb{R}_{>0}$ in the autocorrelation-function of each frame which is in the range of human pitch is detected. The two-dimensional pitch features consist of the logarithm of the pitch period $\tau \in \mathbb{R}_{>0}$ which is the position of that peak and the pitch amplitude which is given by $\frac{r_\tau}{r_0} \in (0, 1)$.

To simplify the experimental setup, the features of the training data were also used as the development data. Due to the lack of validation data, none of the parameters has been fine-tuned but all are set to reasonable values suitable for measuring identification error rates. It should be noted, that in practical applications it is rarely the case that validation data is available at all and therefore this approach is much more realistic.

Using MFCC, PLP and SSC features, we trained diagonal-covariance UBMs with 256 Gaussian components for 5 iterations with a minimum standard deviation of 0.0001. For the GLOMM and pitch features we only used 32 components as they usually perform better when using only a few components. The i-vector models have been trained for 50 iterations and their dimension is 400. When applying PCA to the standard or extended supervector, the same dimension of 400 has been used to be able to compare the results. Since the pitch features are only 2 dimensional and therefore the corresponding supervector is only of size 64, we used a target dimension of size 25 for the i-vectors and standard supervectors instead. But after extending the

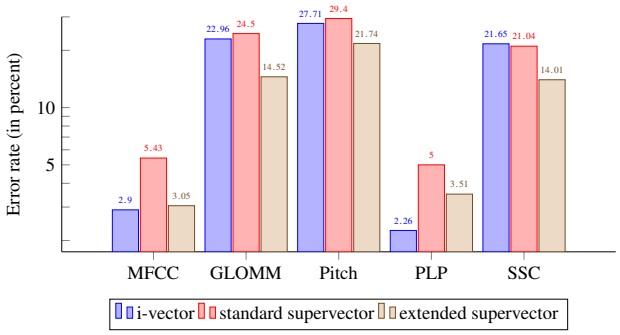

Figure 1: Identification error rates of i-vector, the standard and the extended supervector for each of the 5 features.

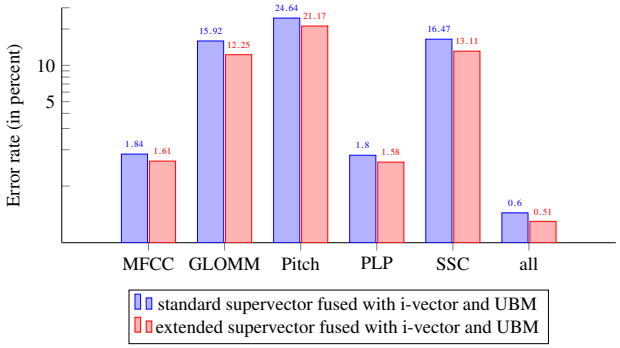

Figure 2: Identification error rates of the standard supervector and the extended supervector obtained after fusing with the scores of UBM and i-vector for each of the 5 features alone and all of them combined.

pitch supervectors, the increased dimension allowed us to use a larger target dimension of size 200. In any case, the extended supervectors have been standardized with the mean and standard deviation of the development/training data before reducing their dimension with PCA. For all three representations of the supervectors, a PLDA model with a latent variable dimension of 200 has been trained for 20 iterations using the fastPLDA toolkit [19].

To evaluate a method, we conducted 500 independent trials in which 10 speakers were chosen at random and the two test files of each speaker were evaluated. Thus, each experiment consisted of 10000 individual 10 speaker classification trials. The same fixed sets were used for all the tests to have comparable results.

3.2 Experimental Results

In Fig. 1, the identification error rates obtained with i-vectors, standard and extended supervectors are depicted for all 5 features. Without exception, the error rates of the extended supervector are significantly lower than the ones of the standard supervector. On average, the reduction is about 32.5%. Thus, extended supervectors cover more information about the speech feature distributions.

When comparing the performance of the i-vector models with the extended supervectors, no general statement can be made about their relation in terms of identification error rate. For both well-performing features (MFCC and PLP), i-vectors lead to lower error rates and on other occasions (pitch, GLOMM and SSC) extendended supervectors are superior. One observation to be made is the difference

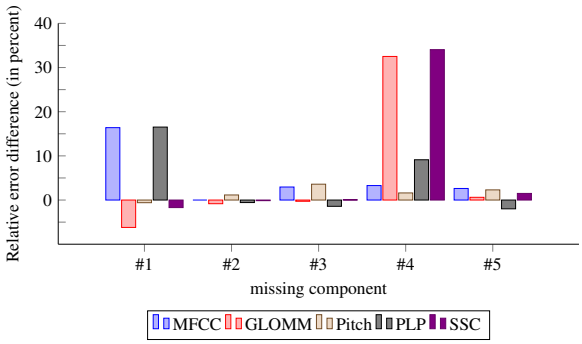

Figure 3: Difference between the identification error rates obtained with the full extended supervector and the case that one of its five components is missing. The missing components are numbered in order of their appearance in the definition of the extended supervector. To give an illustrative example, the error rate increased from 3.05% to 3.55% when evaluating the performance of the extended supervector based on MFCCs without its first component which corresponds to a relative difference of 16.39% and is depicted by the first bar.

in dimension, as a smaller number of Gaussian Components has been used for pitch and GLOMM features which both fall into the latter category. Therefore, this indicates that the extended supervector leads to a smaller error rate when only a few Gaussian components are used.

It should be recalled to memory that the pitch i-vectors have a dimension of size 25 and for the extended supervectors a PCA-dimension of size 200 has been used due to the low dimensional nature of the pitch features. If the extended supervectors are also reduced to a dimension of size 25, the identification error rate is only 28.41% which makes them worse than the i-vectors. Hence, it can be concluded that extending the supervector may even be beneficial if the feature dimension is very low which forces the corresponding i-vector dimension to be small, too.

As the performance of extended supervectors compared to i-vectors differs greatly, one essential step is to combine both approaches. By doing so, one always gets a performance which is better than the best individual one. In addition to that, the results of the enrolled UBM can also be used for combination in case that it covers different information about the speech feature distributions. The combination is done by fusing the resulting log-likelihood values and scores with a weighted sum. To find the optimal fusion coefficients an Evolutionary Algorithm (EA) [20] has been applied. Note that these coefficients may be a bit too optimistic as they are difficult to obtain in real life applications but are suitable for the purpose of comparing the performances. Fig. 2 shows the resulting identification error rates obtained after fusing i-vectors and the UBM with standard supervectors as well as fusing them with extended supervectors. It is clearly visible that the extended supervector still outperforms the standard supervector, even when all 5 features are combined. On average, the error rates are 16.21% lower. Hence, the extended supervectors contain information that has not yet been covered by the i-vectors and the standard supervectors.

In a last experiment we tried to measure the information content of all five components of the extended supervector. For that purpose, the identification error rates have been determined using the same dimension of size 400 in case one of the extended supervector's components is missing. The new error rates obtained this way have been compared to the error rates where all components are used. In Fig. 3, the relative difference between both error rates is depicted for all of the five features. Values above zero percent indicate that the corresponding missing component contains complementing information whereas values below show that the error rate actually decreased when this component has not been included. In order of magnitude, the mean differences are 16.11% for the Bhattacharyya distance based kernel, 4.88% for the Standard Supervector, 1.01% for the Hellinger distance, 0.98% for the log-likelihood values of the UBM and -0.08% for the weights. Hence, it can be argued to omit the weights being the only component which does not lead to a significant improvement. This makes sense, as the normalized weights are contained in the Bhattacharrya distance based kernel. But on the other hand, they also helped to decrease the error rate obtained with the pitch features and therefore may also be helpful in other cases which have not been covered in this experiment. Anyway, the other four components do contain complementing information and thus are essential parts of the extended supervector.

4 Conclusions and Future Work

In this work, an extended supervector for the purpose of capturing all available information about the approximated speech feature distributions has been defined. It consists of the means and weights of all Gaussian components, the log-likelihoods of the UBM, a kernel based on the Bhattacharyya distance and the Hellinger distance between all mixture components of the enrolled UBM and the UBM itself.

Despite the fact that extended supervectors are more than 3 times larger than standard supervectors, it has been shown that the computational complexity of using them is still lower than the one of using i-vectors. As seen in closed-set speaker identification experiments conducted on the NTIMIT corpus, this extended supervector has a performance which is on average 32.5% better than the one of the standard supervector. When combining the presented approach with i-vector and UBM via score-based fusion, on average 16.25% lower identification error rates are obtained compared to a fusion with the standard supervector.

The performance in relation to the widely used i-vector greatly varies as in some cases the extended supervector has a lower error rate and in other cases i-vectors yield better results. But after the fusion, significantly lower identification error rates are obtained compared to i-vector alone. Therefore, the extended supervector fulfills its purpose and acts as a complementary source of information about the speech feature distribution. Hence, the extended supervector is a useful supplement to every speaker recognition system based on i-vectors.

One possibility to further increase the performance may be to reduce the dimension of the extended supervector in a more sophisticated way than applying PCA and should be worth investigating. In addition to that, the extendend supervector should be fused not only with i-vectors but also with the recently proposed speaker embeddings, also called x-vectors, [21–23] which are extracted with deep neural networks. Doing that and evaluating the performance on a NIST Speaker Recognition Evaluation task is subject to future work.

References

[1] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.

[2] H. Hermansky, "Perceptual linear prediction (plp) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[3] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digit. Signal Process.*, vol. 10, pp. 19–41, Jan. 2000.

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

[5] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[6] 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.

[7] 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, pp. 49–52, 2009.

[8] C. H. You, K. A. Lee, and H. Li, "Gmm-svm kernel with a bhattacharyya-based distance for speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1300–1312, 2010.

[9] M. Ben, G. Gravier, and F. Bimbot, "A model space framework for efficient speaker detection," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[10] A. N. Iyer, U. O. Ofoegbu, R. E. Yantorno, and B. Y. Smolenski, "Speaker distinguishing distances: a comparative study," *International Journal of Speech Technology*, vol. 10, no. 2-3, pp. 95–107, 2007.

[11] A. L. Gibbs and F. E. Su, "On choosing and bounding probability metrics," *International statistical review*, vol. 70, no. 3, pp. 419–435, 2002.

[12] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions. Series A, Mathematical, physical and engineering sciences*, vol. 374, no. 2065, 2016.

[13] O. Glembek, L. Burget, P. Matějka, M. Karafiát, and P. Kenny, "Simplification and optimization of i-vector extraction," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 4516–4519, IEEE, 2011.

[14] W. M. Fisher, G. R. Doddington, K. M. Goudie-Marshall, C. Jankowski, A. Kalyanswamy, S. Basson, and J. Spitz, "NTIMIT LDC93S2." Web Download, 1993. Philadelphia: Linguistic Data Consortium.

[15] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, *et al.*, *The HTK book, Version 3.4*. Cambridge University Engeneering Department, 2006.

[16] N. P. H. Thian, C. Sanderson, and S. Bengio, "Spectral subband centroids as complementary features for speaker authentification," in *Biometric Authentification*, pp. 631–639, Springer, 2004.

[17] P. M. Baggenstoss, "Combining the glottal mixture model (glomm) with ubm for speaker recognition," in *Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*, pp. 2156–2160, IEEE, 2016.

[18] P. M. Baggenstoss, K. Wilkinghoff, and F. Kurth, "Glottal mixture model (glomm) for speaker identification on telephone channels," in *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, pp. 2803–2807, IEEE, 2017.

[19] A. Sizov, K. A. Lee, and T. Kinnunen, "Unifying probabilistic linear discriminant analysis variants in biometric authentication," in *Proc. S+SSPR*, pp. 464–475, Springer, 2014. Software available at https://sites.google.com/site/fastplda/.

[20] T. Bäck, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[21] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*, pp. 165–170, IEEE, 2016.

[22] 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, IEEE, 2017.

[23] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*, IEEE, 2018.