INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Statistical Modeling of Speaker's Voice with Temporal Co-Location for Active Voice Authentication

Zhong Meng, Biing-Hwang (Fred) Juang

School of Electrical and Computer Engnineering, Georgia Institute of Technology
75 5th Street NW, Atlanta, GA 30308, USA
zhongmeng@gatech.edu, juang@ece.gatech.edu

Abstract

Active voice authentication (AVA) is a new mode of talker authentication, in which the authentication is performed continuously on very short segments of the voice signal, which may have instantaneously undergone change of talker. AVA is necessary in providing real-time monitoring of a device authorized for a particular user. The authentication test thus cannot rely on a long history of the voice data nor any past decisions. Most conventional voice authentication techniques that operate on the assumption that the entire test utterance is from only one talker with a claimed identity (including i-vector) fail to meet this stringent requirement. This paper presents a different signal modeling technique, within a conditional vector-quantization framework and with matching short-time statistics that take into account the co-located speech codes to meet the new challenge. As one variation, the temporally co-located VQ (TC-VQ) associates each codeword with a set of Gaussian mixture models to account for the co-located distributions and a temporally co-located hidden Markov model (TC-HMM) is built upon the TC-VQ. The proposed technique achieves an window-based equal error rate in the range of 3-5% and a relative gain of 4-25% over a baseline system using traditional HMMs on the AVA database.
Index Terms: vector quantization, co-located frames, hidden Markov model, active voice authentication

1. Introduction

An active voice authentication (AVA) system is intended to actively and continuously validate the identity of a person by taking advantage of his/her unique voice characteristics without prompting the user for credentials. AVA is significantly different from the conventional speaker verification in that its goal is to make a decision on the speaker identity at each time instant rather than to make a final decision after the entire test utterance is obtained because the test utterance may instantaneously undergo change of speaker in the scenario of AVA. To satisfy both the real-time requirement and statistical reliability, AVA slides a "test window" with about one second of speech data over the test utterance at the rate of 100 per second and provides a decision for each window about the speaker identity as in Fig. 1.

For AVA, we use window-based equal error rate (WEER) as the performance metric. A window-based miss detection error (WMDE) occurs if an "true speaker" decision is made while the impostor is speaking within that window. A window-based false alarm error (WFAE) occurs if a "impostor" decision is made while the "true speaker" is actually speaking within that window. With all the decisions made for the windows sliding over all the test speech signal, the window-based miss detection rate (WMDR) and the window-base false alarm rate (WFAR) can be computed. The WEER is reached when the WMDR and WFAR are equal.

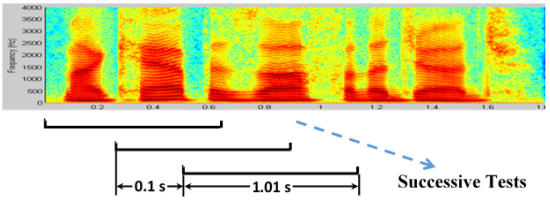

Figure 1: An illustration of the successive tests performed with data windows on a short-time spectrogram.

A large number of statistical modeling techniques have been proposed to characterize a speaker's voice. In 1980s, a vector quantization (VQ) codebook was used to characterize the short-time spectral features of a speaker and recognize the identity of an unknown speaker from his/her speech based on a minimum distance classification rule [1]. During the 1990s, continuous ergodic hidden Markov models (HMM) were used for text-independent speaker verification [2]. In the 2000s, the speaker-specific voice characteristics were modeled statistically by the maximum a-posterior (MAP) adapted speaker-independent Gaussian mixture models (GMMs) [3, 4, 5]. Based on this, the application of support vector machines (SVM) in a GMM supervector space [6, 7, 8] modeled the speaker voice by performing a nonlinear mapping from the input space to an SVM kernel space. Recently, factor analysis methods such as joint factor analysis (JFA) [9, 10, 11, 12] and i-vectors [13, 14, 15, 16] have achieved state-of-the-art performance in NIST speaker recognition evaluations (SRE). These approaches model the speaker and channel variability by projecting speaker-dependent GMM mean supervectors onto a space of reduced dimensionality.

Although these traditional methods are able to capture long-term characteristics of a speaker's voice, they fail to robustly model the short-time statistics. In Section 3, we show that the AVA system based on i-vector achieves a perfect authentication performance when the duration of the test window is long enough (above 2.01 s). But the performance degrades rapidly as the test window duration decreases. In general, many other i-vector based systems exhibit sharp performance degradation [17, 18], when they are tested with short duration (below 5s) utterances. This is understandable as the covariance matrix of the i-vector is inversely proportional to the number of speech frames per speaker utterance and the variance of the i-vector estimate grows directly as the number of frames in the utterance decreases [19].

The authors would like to thank Chao Weng and M Umair Bin Altaf at Georgia Institute of Technology for their help on AVA system.

http://dx.doi.org/10.21437/Interspeech.2016-650

The test statistic in AVA consists of $p(X|\Lambda^{\text{target}})$ and $p(X|\Lambda^{\text{anti}})$ where X is the speech data (or its spectral representations) in a test window and Λ^{target} , Λ^{target} are the pair of target and anti-target statistical models that correspond to X. This is remarkably different from the conventional test statistic based on X of unspecified duration. Therefore, the duration of the training speech segments should be equal to that of the test window. Further, $p(X|\Lambda^{\text{target}})$ and $p(X|\Lambda^{\text{anti}})$ conventionally model the spectral characteristics encapsulated in one single frame and the sequential constraints between frames through transition probabilities. However, in the case of AVA, the sequential constraints do not play an effective role in characterizing the voice of a speaker because of the limited number of frames included in each short-duration test window. It is thus necessary to expand the richness of the statistics along the sequence of speech frames via modeling a set of *temporally co-located frames (TCFs)*, i.e, modeling the spectral characteristics within a speech segment that is longer than one signal frame.

Therefore, we propose a *VQ-conditioned model*, in which a set of local models are built over a block of TCFs anchored on each VQ codeword. Each local model characterizes the probability distribution of one TCF anchored at a certain codeword. Many types of the VQ-conditioned models can be constructed with this approach. Each VQ codeword can be associated with a set of GMMs to model the TCFs or we can use an HMM to model the TCFs anchored at each VQ codeword. By introducing the transition probabilities, the VQ codebook can be recast into an HMM in which each state corresponds to one codeword in the original VQ codebook. A set of GMMs serve as the probability output of each HMM state to model the TCFs. We call this HMM the *temporally co-located HMM (TC-HMM)*. In this work, we focus on the statistical modeling of the speaker voice characteristics using TC-HMM. The parameters of TC-HMM are re-estimated and adapted using an *expectation-maximum (EM) algorithm* [20].

To validate the proposed framework, AVA database is recorded. The AVA system based on the TC-HMM framework achieves an average WEER of 3-5% and a relative gain of 4-25% over the baseline system using traditional HMM.

In Section 2, we introduce the AVA database used for performance evaluation. In Section 3, we introduce how i-vector applied to AVA and analyze its performance. In Section 4, we define the VQ-conditioned model. In Section 5, we introduce how the VQ-conditioned models are trained and adapted for the AVA task and how the sequential testing is conducted. In Section 6, the experimental results on AVA database are analyzed.

2. AVA Database

The NIST SRE Training and Test Sets are widely used to evaluate the performance of speaker verification systems [21, 22]. However, it is not suited for the AVA system because even though a test utterance in NIST SRE is labeled as coming from a certain speaker, some portions of the utterance may be actually from other speakers. These crosstalk components make the evaluation results of a real-time system meaningless as we are not able to know real identity for each time instant. This necessitates the collection of a completely new data base for the performance evaluation of AVA system.

We collect a voice database to train and validate the AVA user models from 25 volunteers (14 females, 11 males). A Microsoft Surface Pro tablet was used to record the data. The data was recorded from the built-in microphone on the tablet at 8000 samples per second and it includes about 2.25 hours of voice recordings. The data that we collected from each person consists of four parts: a rainbow passage [23], a user-chosen pass-phrase, 20 randomly selected sentences from phonetically balanced Harvard sentences [24] (5.5s on average) and 30 digit pairs (each digit is randomly selected from 0 to 9). The speaker repeats the same pass-phrase 8 times.

Number of Mixtures	Test Window Duration (s)				
	1.01	1.51	2.01	2.51	3.01
128	13.72	7.24	3.72	1.56	0.58
256	13.89	7.29	**3.69**	**1.43**	**0.35**
512	**12.91**	**6.92**	3.79	1.44	0.52
1024	14.54	8.02	3.99	1.62	0.64

Table 1: WEER (%) of AVA using i-vector on AVA database with different test window durations and UBM configurations.

For the performance evaluation, the Rainbow passage, the pass-phrases and digits are used for training while the Harvard sentences are used for testing. The audio signal is converted to the conventional 39-dimension MFCC features using a 25 ms Hamming window and a 10 ms frame advance. The cepstral mean is subtracted to minimize the channel variability. The training data is 240 seconds long on average for each speaker.

3. AVA with I-Vector

In this section, we investigate if i-vector, the state-of-the-art technique in conventional speech verification, can also achieve extraordinary performance for AVA. Within the i-vector framework, it is assumed that a linear dependence exists between the speaker adapted GMM supervectors μ and the speaker-independent GMM supervector m [13].

$$\mu = m + Tw \tag{1}$$

where T is a low rank factor loading matrix estimated through EM algorithm [19] and w is a standard normal distributed random vector. The i-vector is an MAP estimate of w.

We first apply i-vector to the conventional speaker verification task under the assumption that each test utterance is from only one speaker. At the training stage, we train a GMM universal background model (UBM) with all the training data in the AVA database. With EM algorithm, a speaker-independent factor loading matrix T_{SI} is trained by using the sufficient statistics collected from the UBM. Then an i-vector is extracted for each speaker using his or her training data and T_{SI}. During testing, an i-vector is extracted from the each test utterance using T_{SI}. The cosine distance between the i-vector of each test utterance and that of the hypothesized speaker is used as the decision score. An EER is computed with all the utterance-level decision scores and the ground truth. In AVA database, the i-vector achieves 0.00% EER for the utterance-based speaker verification under all UBM configurations.

Then we use i-vector in the AVA task. We apply the same training procedure as in the traditional case, but during testing, a test window of prescribed duration is slided over the test utterance at the rate of 100 per second and an i-vector is extracted from the speech signal within each test window using T_{SI}. The cosine distance between the i-vector of each test window and that of the hypothesized speaker is used as the decision score. We show the WEER results with respect to the test window duration and the number of mixtures in the UBM in Table 1.

For each UBM configuration, the i-vector based AVA system achieves perfect performance when the duration of the test window is above 2.51 s. However, the performance degrades

tremendously as the test window duration falls below 2.01 s. Note that when the test window is at 1.01 s, the WEER degrades to around 13.00%. This trend of WEER performance indicates that i-vector technique works perfectly when the duration of the test segment is long enough, but does not work well for extremely short test segments.

4. VQ-conditioned Model

To better model the short-time speaker characteristics, training speech segments with the same duration of the test window are extracted from the training speech data. This is done by sliding a window at the rate of approximately 100 per second. The speech segment within the sliding window at each time will serve as the training token.

Further, *VQ-conditioned model* is proposed to overcome the insufficiency of speech statistics within the short-duration speech segment which represents speaker characteristics. The VQ-conditioned model is a codebook and a set of probability distributions of the anchor frame with its TCFs conditioned on each codeword.

Let $X = \{x_1, x_2, \ldots, x_T\}$ denote a training token from a certain speaker. Given x_t as the *anchor (local) frame* at discrete time t, $\{x_{t+k} | -K \leq k \leq K, k \neq 0\}$ is the set of TCFs of x_t and x_{t+k} is the k^{th} TCF of x_t. Further, if x_t is quantized to codeword j through a codebook Q, x_{t+k} is the k^{th} TCF of codeword j. $k < 0$ indicates that the TCF is ahead of x_t in time. With the above notation, the VQ-conditioned model is given by

$$\begin{aligned}\Lambda = \{&p(x_{t+k}|Q(x_t) = j),\\ &k = 0, \pm 1, \ldots, \pm K, j = 1, 2, \ldots, L.\}\end{aligned} \tag{2}$$

where $Q(x_t) = j$ means that x_t is quantized to codeword j through a codebook Q. $p(x_{t+k}|Q(x_t) = j)$ is the conditional distribution of the k^{th} TCF of codeword j, and is called the k^{th} *temporally co-located distribution (TCD)* of codeword j.

In this work, we use GMMs to model the TCDs and name this model *temporally co-located VQ (TC-VQ)*. As a variation of the VQ-conditioned model, TC-HMM is built up upon the TC-VQ by mapping the VQ codewords to the HMM states, embedding the $(2K + 1)L$ GMMs as the state probability outputs and introducing the transition probabilities.

5. AVA with VQ-conditioned Model

For the AVA task, we need to train a pair of models (target and anti-target models) for each speaker and use them to verify the claim of each test speech segment. With VQ-conditioned modeling, we first use all the training data in AVA database to train a *speaker-independent TC-VQ (SI-TC-VQ)* in which each codeword is associated with a set of $(2K + 1)$ speaker-independent GMMs. Then we recast the SI-TC-VQ into a *speaker-independent TC-HMM (SI-TC-HMM)* that is used as the anti-target model for all the speakers during sequential testing. Finally, the SI-TC-HMM is trained and then adapted to the speech signal of each individual speaker to generate a set of *speaker-dependent TC-HMMs (SD-TC-HMMs)* which are used as the target models.

5.1. SI-TC-VQ Training

In this section, a SI-TC-VQ is trained to serve as the anti-target model for sequential testing of AVA. At first, K-means clustering algorithm [25] is used to generate a speaker-independent codebook Q of size L with a set of codewords $\mathcal{C} = \{1, 2, \ldots, L\}$ using the speech signal from all speakers. We quantize training frames from all the speakers with Q so that each training frame is assigned a codeword. Q is also used to quantize the test frames in the testing stage. Then we estimate the distributions of the TCFs given each codeword in Eq. (2). The k^{th} TCD is given by the following GMM-UBM,

$$\begin{aligned}p(x_{t+k}|Q(x_t) = j) &= \sum_{m=1}^{M} w_{jkm}\mathcal{N}_m(x_{t+k}|\mu_{jkm}, \Sigma_{jkm})\\ j = 1, 2, \ldots, L, k &= 0, \pm 1, \ldots, \pm K, m = 1, 2, \ldots, M\end{aligned} \tag{3}$$

where d is the dimension of each frame and μ_{jkm}, Σ_{jkm}, w_{jkm} is the mean vector, covariance matrix and weight of the m^{th} mixture component of the UBM that models the k^{th} TCF of codeword j, respectively.

We pool all the k^{th} TCFs of codeword j in codebook Q to train the UBM $p(x_{t+k}|Q(x_t) = j)$ via EM algorithm.

5.2. SI-TC-HMM Training

In this section, SI-TC-HMM is trained to model the temporal structure in the speakers' voice. The SI-TC-HMM is initialized from the SI-TC-VQ trained in Section 5.1 as follows. First, the set of L codewords $\mathcal{C} = \{1, \ldots, L\}$ of SI-TC-VQ is mapped to a set of L states $\mathcal{S} = \{1, \ldots, L\}$ of the SI-TC-HMM such that the frames that were quantized to codeword j are now aligned with state j. Then, the UBM that models the distribution of the k^{th} TCF of codeword j now serves as the probability output of state j, i.e.,

$$p(x_{t+k}|s_t = j) = p(x_{t+k}|Q(x_t) = j). \tag{4}$$

Assume that N_i is the number of frames aligned with the SI-TC-HMM state i and N_{ij} is the number of frames aligned with state i with its next frame aligned with state j. The initial transition matrix $A = [a_{ij}]$ of the SI-TC-HMM is

$$a_{ij} = N_{ij}/N_i \tag{5}$$

where $i, j = 1, \ldots, L$. We further generate the new alignment of speech frames against the SI-TC-HMM states through *Viterbi algorithm* as follows. If we have a training token $X = \{x_1, \ldots, x_T\}$ and $\phi_t(j)$ represents the maximum likelihood of observing speech vectors x_1 to x_t being in state j at time $t (1 \leq t \leq T)$, that is,

$$\phi_t(j) = \max_{s_1, \ldots, s_{t-1}} P(x_1, \ldots, x_t, s_1, \ldots, s_{t-1}, s_t = j|\Lambda), \tag{6}$$

where s_t is the codeword that frame x_t is aligned with. The optimal state sequence $\hat{S} = \{\hat{s}_1, \ldots, \hat{s}_T\}$ that X is aligned with can be obtained using the following recursion

$$\phi_t(j) = \max_i\{\phi_{t-1}(i)a_{ij}\}\left[\prod_{k=-K}^{K} p(x_{t+k}|s_t = j)\right]^{\frac{1}{2K+1}} \tag{7}$$

a_{ij} and $p(x_{t+k}|s_t = j)$ have been initialized in Eqs. (4), (5).

Then we re-estimate the parameters A and Θ in SI-TC-HMM. For any training token X, if x_t is aligned with $\hat{s}_t$, its k^{th} TCF x_{t+k} is used to train the UBM $p(x_{t+k}|s_t = \hat{s}_t)$.

5.3. SI-TC-HMM Adaptation

The SD-TC-HMM is generated by adapting the UBMs embedded in the SI-TC-HMM states to the training data of the speaker

with MAP estimation. The SD-TC-HMM is used as the target model in sequential testing.

For an adaptation token $X^a = \{x_1^a, \ldots, x_{T^a}^a\}$ from a certain speaker, if speech frame x_t^a is aligned with the state $\hat{s}_t$ of the SI-TC-HMM trained in Section 5.2, its TCF x_{t+k}^a is used to adapt the UBM $p(x_{t+k}|s_t = \hat{s}_t)$ with EM algorithm as follows.

1) E-step: We compute the posterior of mixture m given the k^{th} TCF x_{t+k}^a of state j within adaptation data,

$$p(m|x_{t+k}^a) = \frac{w_m \mathcal{N}(x_{t+k}^a|\mu_{jkm}, \Sigma_{jkm})}{\sum_{i=1}^{M} w_i \mathcal{N}(x_{t+k}^a|\mu_{jki}, \Sigma_{jki})}, t \in \mathcal{T}_j^a \quad (8)$$

$$\mathcal{T}_j^a = \{t|x_t^a \text{ is aligned with state } j \text{ of the SI-TC-HMM}\} \quad (9)$$

2) M-step: The speaker-adapted mean vector $\hat{\mu}_{jkm}$, variance matrix $\hat{\Sigma}_{jkm}$ and weight $\hat{w}_{jkm}$ of the m^{th} component of GMM $\hat{p}(x_{t+k}|s_t = j)$ is updated as

$$N_{jkm}^a = \sum_{t \in \mathcal{T}_j^a} p(m|x_{t+k}^a), \qquad \alpha = \frac{N_{jkm}^a}{N_{jkm}^a + \tau} \quad (10)$$

$$\hat{\mu}_{jkm} = (1-\alpha)\mu_{jkm} + \frac{\alpha}{N_{jkm}^a} \sum_{t \in \mathcal{T}_j^a} p(m|x_{t+k}^a) x_{t+k}^a \quad (11)$$

$$\hat{\Sigma}_{jkm} = (1-\alpha)\Sigma_{jkm} + \frac{\alpha}{N_{jkm}^a} \sum_{t \in \mathcal{T}_j^a} p(m|x_{t+k}^a) (x_{t+k}^a - \mu_{jkm})(x_{t+k}^a - \mu_{jkm})^\top \Sigma_{jkm}^a \quad (12)$$

$$\hat{w}_{jkm} = (1-\alpha)w_{jkm} + \alpha \frac{N_{jkm}^a}{|\mathcal{T}_j^a|} \quad (13)$$

where $|\mathcal{T}_j^a|$ is the number of frames in X^a that are aligned with state j. For simplicity, the transition probabilities in SD-TC-HMM remain the same as in SI-TC-HMM.

5.4. Sequential Testing

In the testing stage, the AVA sequentially takes in a sliding window of speech frames and calculates the log-likelihood with respect to both the target and anti-target models for the registered speaker. Assume that we have the target SD-TC-HMM and the anti-target SI-TC-HMM with parameters Λ^{target} and Λ^{anti} respectively. The LLR score for the test window is given by

$$\Gamma(X|\Lambda^{\text{target}}, \Lambda^{\text{anti}}) = \frac{1}{T}\left[\log p(X|\Lambda^{\text{target}}) - \log p(X|\Lambda^{\text{anti}})\right] \quad (14)$$

$$p(X|\Lambda) = \max_S p(X, S|\Lambda) = \max_i \phi_T(i) \quad (15)$$

where $\Lambda = \{\Lambda^{\text{target}}, \Lambda^{\text{anti}}\}$ and $\phi_T(i)$ can be obtained by the recursion in Eq. (7). Then we compare $\Gamma(X|\Lambda^{\text{target}}, \Lambda^{\text{anti}})$ with threshold γ to make a decision on the speaker identity for that window. By varying γ, WEER can be calculated.

6. Experiments

We use the training and test data in the AVA database described in Section 2 to evaluate the performance of the AVA system based on VQ-conditioned models. Training of the needed models has been described in Section 5. The number of mixture components for each GMM is fixed at 4 and the duration of the test window is fixed at 1.01 s.

First, we show the WEER results with respect to the number of TCF modeled in each TC-HMM state ($2K$) and the number of states in the TC-HMM in Table 2. Note that only the anchor frame is modeled when $2K = 0$, which is equivalent to the traditional HMM case. As is observed from Table 2, the AVA system based on TC-HMM framework achieves an average WEER of 3-5% and a relative gain of 4-25% over the baseline system using traditional HMM. This indicates that VQ-conditioned model successfully meets the real-time requirement of AVA which most of the conventional speaker verification techniques (including i-vector) fail to satisfy.

Number of States	Number of Co-Located GMMs in Each State ($2K$)					
	0	2	4	6	8	10
64	4.867	4.828	4.260	4.230	**4.172**	4.360
128	4.513	4.308	4.015	4.006	**3.950**	4.618
256	4.308	3.779	3.889	**3.238**	3.814	-
512	3.749	3.941	3.639	**3.609**	3.736	-
1024	4.963	**4.194**	4.745	4.451	4.671	-

Table 2: WEER (%) of AVA system on AVA database under different TC-HMM configurations.

Number of Codewords	Number of Co-Located GMMs for Each Codeword($2K$)				
	0	2	4	6	8
64	5.129	**5.099**	5.173	5.265	5.422
128	4.775	**4.736**	4.845	5.068	4.968
256	4.845	4.946	4.933	4.819	**4.749**
512	4.391	**3.958**	4.151	4.295	4.273
1024	5.260	4.513	4.548	**4.177**	4.229

Table 3: WEER (%) of AVA system on AVA database under different TC-VQ configurations.

The performance gain comes from the co-located GMMs in each TC-HMM state since, for a fixed number of states, the WEER first decreases gradually as $2K$ grows and then decreases when $2K$ becomes too large. This is because, during Viterbi alignment, the state output probability of each anchor frame is evaluated by a set of $(2K+1)$ GMMs which models both the anchor frame and the TCFs instead of using one single GMM as in the traditional HMM case and the best state sequence obtained in this way is thus more accurate. However, the far-away TCFs are loosely correlated with the anchor frame and provide inaccurate statistics that degrade the performance when $2K$ continues to increase.

From Table 2, we can also see that the value of $2K$ at which the lowest WEER is achieved becomes smaller as the number of states in TC-HMM grows . This is because, as the number of states in TC-HMM increases, the amount of data used to train or adapt the GMMs that model far-away TCFs becomes insufficient, which makes the estimation of these GMMs less accurate.

In Table 3, we show that the AVA performance is improved by recasting TC-VQs into TC-HMMs. The reason is that, with TC-HMMs, transition probabilities are introduced to model the sequential constraints of the states and, through Viterbi alignment, the state sequence that the speech frames are aligned with are sequentially optimal rather than locally optimal as in the TC-VQ case.

7. Conclusions

In this work, the VQ-conditioned modeling framework is introduced to model the short-time characteristics of the speaker voice as is required by AVA. The proposed framework achieves consistent and significant performance gain over the systems using traditional HMMs on the AVA task. The gain comes from the temporally co-located GMMs and the sequential constraints introduced by the transition probabilities.

8. References

[1] F. Soong, A. Rosenberg, L. Rabiner, and B. Juang, "A vector quantization approach to speaker recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '85.*, vol. 10, Apr 1985, pp. 387–390.

[2] T. Matsui and S. Furui, "Comparison of text-independent speaker recognition methods using vq-distortion and discrete/continuous hmm's," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 3, pp. 456–459, Jul 1994.

[3] D. Reynolds and R. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, Jan 1995.

[4] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 13, pp. 19 – 41, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1051200499903615

[5] J. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 2, pp. 291–298, Apr 1994.

[6] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proceedings of ICASSP*, 2006, pp. 97–100.

[7] N. Dehak, P. Kenny, R. Dehak, O. Glembek, P. Dumouchel, L. Burget, V. Hubeika, and F. Castaldo, "Support vector machines and joint factor analysis for speaker verification," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, April 2009, pp. 4237–4240.

[8] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for svm-based speaker recognition," in *Proc. of ICSLP*, 2006, p. 14711474.

[9] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," CRIM, Tech. Rep., 2005.

[10] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[11] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 3, pp. 345–354, May 2005.

[12] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in gmm-based speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1448–1460, May 2007.

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

[14] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 5, pp. 980–988, July 2008.

[15] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems." in *Interspeech*, 2011, pp. 249–252.

[16] P. Matejka, O. Glembek, F. Castaldo, M. Alam, O. Plchot, P. Kenny, L. Burget, and J. C?ernocky, "Full-covariance ubm and heavy-tailed plda in i-vector speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May 2011, pp. 4828–4831.

[17] A. Kanagasundaram, R. Vogt, D. B. Dean, S. Sridharan, and M. W. Mason, "I-vector based speaker recognition on short utterances," in *INTERSPEECH*, 2011, pp. 2341–2344.

[18] A. K. Sarkar, D. Matrouf, P.-M. Bousquet, and J.-F. Bonastre, "Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification." in *INTERSPEECH*, 2012.

[19] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 3, pp. 345–354, May 2005.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. pp. 1–38, 1977. [Online]. Available: http://www.jstor.org/stable/2984875

[21] M. Przybocki and A. F. Martin, "Nist speaker recognition evaluation chronicles," in *ODYSSEY04-The Speaker and Language Recognition Workshop*, 2004.

[22] N. I. of Standards and Technology, "Speaker recognition evaluation," http://www.itl.nist.gov/iad/mig/tests/spk/, [Online; Accessed: 30-Sep-2014]. [Online]. Available: \url{http://www.itl.nist.gov/iad/mig/tests/spk/}

[23] G. Fairbanks, *Voice and articulation drillbook*. Harper & Brothers, 1940.

[24] "IEEE recommended practice for speech quality measurements," *Audio and Electroacoustics, IEEE Transactions on*, vol. 17, no. 3, pp. 225–246, Sep 1969.

[25] B.-H. Juang and L. Rabiner, "The segmental k-means algorithm for estimating parameters of hidden markov models," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 9, pp. 1639–1641, Sep 1990.