Efficient Speaker Recognition Using Approximated Cross Entropy (ACE)

Hagai Aronowitz and David Burshtein, *Senior Member, IEEE*

***Abstract*—Techniques for efficient speaker recognition are presented. These techniques are based on approximating Gaussian mixture modeling (GMM) likelihood scoring using approximated cross entropy (ACE). Gaussian mixture modeling is used for representing both training and test sessions and is shown to perform speaker recognition and retrieval extremely efficiently without any notable degradation in accuracy compared to classic GMM-based recognition. In addition, a GMM compression algorithm is presented. This algorithm decreases considerably the storage needed for speaker retrieval.**

***Index Terms*—Speaker identification, speaker indexing, speaker recognition, speaker retrieval, speaker verification.**

I. Introduction

STATE-OF-THE-ART text-independent speaker recognition algorithms often use Gaussian mixture models (GMMs) [1] for acoustic modeling. Introduced in the 1990s [2]–[4], GMM-based speaker recognition has been the state of the art for more than a decade. A GMM-based system computes the log-likelihood of a test utterance given a target speaker by fitting a parametric model (a GMM) to the target training data and computing the average log-likelihood of the test-utterance feature vectors assuming frame independence.

Recently, other novel approaches for speaker recognition [5]–[9] have been developed and applied successfully. Nevertheless, GMM modeling is still a major tool in speaker recognition, used by improved algorithms such as [10]–[14]. Furthermore, GMMs are also a standard tool for language identification [15], and channel detection [1].

Lately, accuracy of automatic speaker recognition systems has improved dramatically thanks to channel compensation and intraspeaker variability modeling [10]–[14]. Therefore, other aspects, such as complexity, gain importance. Speaker recognition technology may be used in various scenarios, including speaker verification, speaker identification, and speaker retrieval. Speaker verification, i.e., deciding whether a target speaker is the speaker of a given audio file usually requires using normalization techniques such as Z-norm [1], T-norm [16], or a combination of both, which necessitates computation of a GMM score between many pairs of speaker models and audio files. Speaker identification, i.e., searching for the identity of a speaker of an audio file within a possibly large open-set (where the unknown speaker may not exist in the set) of speakers may also require many computations of GMM scores.

Retrieval in large audio archives has emerged recently [17], [18] as an important research topic as large audio archives now exist. Speaker retrieval is an essential component of a speech retrieval system. The goal of a speaker retrieval system is to be able to efficiently retrieve occurrences of a given speaker in an audio archive. This can be achieved by dividing the speaker recognition process into two stages. The first one is an indexing phase which is usually done online as audio is recorded and archived. In this stage, there is no knowledge about the target speakers. The goal of the indexing stage is to execute all possible precalculations in order to make the search as efficient as possible when a query is presented. The second stage is activated when a target speaker query is presented. At this point, the precalculations of the first stage are used.

The bottleneck in terms of time complexity of the GMM-based speaker recognition algorithm is the calculation of the log-likelihood of an utterance given a speaker model. A first step towards improving the time complexity is to speed up the likelihood calculation by exploiting redundancy in the time domain (frame decimation) or in the GMM domain (top-N decoding) [19].

A different approach which is more suitable for speaker retrieval is anchor modeling [20]–[23]. Under the anchor modeling framework, each utterance, both training and test utterances, is projected into an anchor space defined by a set of anchor models which are nontarget speaker models. Each utterance is represented in the anchor space by a vector of distances between the utterance and each anchor model. A distance between two utterances is defined as the distance (not necessarily Euclidean) in anchor space. Anchor modeling is highly suitable for speaker retrieval due to the fact that most of the computation burden for comparing two utterances is in the process of projecting them into anchor space, a projection that can be done in an indexing stage for the utterances in the audio archive, and only the query must be projected during the query stage. The disadvantage of anchor modeling is that some speaker information is lost by the projection into anchor space. Indeed, the accuracy reported in [20] and [21] for the anchor models framework is degraded compared to conventional GMM scoring. In [23], the performance of anchor-modeling-based speaker recognition was improved using probabilistic modeling in anchor space. However, more accurate recognition can be achieved by applying similar probabilistic modeling directly to

Manuscript received February 15, 2007; revised May 15, 2007. This work was supported in part by Muscle, a European network of excellence funded by the EC 6th Framework IST Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jean-François Bonastre.

H. Aronowitz was with the Computer Science Department, Bar-Ilan University, Ramat-Gan 52900, Israel. He is now with the Advanced LVCSR Group, T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: haronow@us.ibm.com; aronowitzh@yahoo.com).

D. Burshtein is with the School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel (e-mail: burstyn@eng.tau.ac.il).

Digital Object Identifier 10.1109/TASL.2007.902059

the GMM framework [10], [12], [13]. Furthermore, the anchor modeling approach by itself performs a considerable amount of GMM score calculations, which preferably should be done efficiently.

Our suggested approach for efficient speaker recognition is based on the assumption that a GMM extracts the entire speaker information from an utterance, i.e., the GMM parameters comprise a sufficient statistic for estimating the identity of the speaker. Our novelty is to exploit the same modeling assumption for test utterances in order to derive a computationally efficient score instead of the standard GMM score. We began to explore this approach in [24]–[26]. In this paper, we report a complete description and experimental analysis.

Parameterization of both training and test utterances in a symmetric framework was done in [27], where both target speakers and test utterances were treated symmetrically by being modeled by a covariance matrix. The distance between a target speaker and a test utterance was defined as a symmetric function of the target model and the test utterance model. Unfortunately, a covariance matrix lacks the modeling power of a GMM, which results in low accuracy. In [28], cross likelihood ratio was calculated between the GMM representing a target speaker and a GMM representing a test utterance. This was done by switching the roles of the training and test utterances and averaging the likelihood of the test utterance given the GMM parameterization of the training utterance with the likelihood of the training utterance given the GMM parameterization of the test utterance, but the inherent asymmetry of the GMM scoring remained.

Parameterization of a test utterance by a GMM may be beneficial for improving complexity but may also improve robustness by estimating a GMM parameterization for a test utterance using maximum *a posteriori* (MAP) adaptation with a universal background model (UBM) as a prior.

According to our suggested approach, a GMM is fitted for a test utterance, and the likelihood is calculated by using only the GMM of the target speaker and the GMM of the test utterance.

This paper is organized as follows. The proposed speaker recognition algorithm is presented in Section II. Section III describes how to improve the time complexity of the proposed algorithm. Section IV presents an algorithm for fast GMM decoding and fast GMM MAP adaptation. Section V describes the experimental setup and results. In Section VI, we analyze the time complexity of the proposed system for identification of a large population of speakers and for speaker retrieval in large audio archives. Section VII describes a GMM compression algorithm used for compressing the index of our speaker retrieval system. Finally, Section VIII concludes the paper.

II. Speaker Recognition Using Approximated Cross Entropy (ACE)

In this section, we describe our proposed speaker recognition algorithm. Our goal is to approximate the calculation of a GMM score without using the test utterance raw data. Instead, a GMM fitted to the test utterance is used. We first show in Section II-A that the average log-likelihood of a test utterance can be approximated by the negative cross entropy of the target GMM and the true model for the test utterance. In Sections II-B and II-C, we describe methods for estimating the cross entropy given an estimated GMM for the test utterance. In Sections II-D and II-E we analyze the special cases of using speaker-independent diagonal covariance matrices and global diagonal covariance matrices.

A. Approximating the Likelihood of a Test Utterance

The average log-likelihood of a test utterance $X = x_1, \ldots, x_n$ according to some GMM denoted by Q (which represents some target speaker) is defined as

$$\text{score}\left(X\,|Q\right) = \frac{1}{n}\sum_{i=1}^{n}\log\left(\Pr\left(x_i\,|Q\right)\right). \tag{1}$$

The vectors $x_1, \ldots, x_n$ of the test utterance are acoustic observation vectors generated by a stochastic process. Let us assume that the true model that generated the vectors $x_1, \ldots, x_n$ is a GMM denoted by P. The average log-likelihood of an utterance $x_1, \ldots, x_n$ of asymptotically infinite length n drawn from model P, is

$$\begin{aligned}\frac{1}{n}LL\left(X\,|Q\right) &= \frac{1}{n}\sum_{i=1}^{n}\log\left(\Pr\left(x_i\,|Q\right)\right)\\ &\xrightarrow[n\to\infty]{} \int_x \Pr\left(x\,|P\right)\log\left(\Pr\left(x\,|Q\right)\right)dx\\ &= -H\left(P,Q\right)\end{aligned} \tag{2}$$

where $H(P,Q)$ is the cross entropy between GMMs P and Q. Equation (2) follows by an assumed ergodicity of the speech frame sequence and the law of large numbers. According to (2), the log-likelihood of a test utterance given GMM Q is a random variable that asymptotically converges to the negative cross entropy of P and Q. Therefore, by calculating the log-likelihood of a test utterance X given GMM Q, one is actually trying to estimate the negative cross entropy between P and Q. A different approach would be to estimate the negative cross entropy between P and Q directly using a MAP estimation of P as in (3)

$$\begin{aligned}&\int_x \Pr\left(x\,|P\right)\log\left(\Pr\left(x\,|Q\right)\right)dx\\ &\qquad\cong \int_x \Pr\left(x\,\middle|\,\hat{P}\right)\log\left(\Pr\left(x\,|Q\right)\right)dx\\ &\hat{P} = \arg\max_P \Pr\left(P\,|X\right)\end{aligned} \tag{3}$$

or to estimate the expected negative cross entropy as follows:

$$E_P\int_x \Pr\left(x\,|P\right)\log\left(\Pr\left(x\,|Q\right)\right)dx \tag{4}$$

where E_P is an expectation under the distribution P.

B. Approximating $\int_x \Pr\left(x\,|P\right)\log\left(\Pr\left(x\,|Q\right)\right)dx$

GMMs P and Q are defined as

$$\begin{aligned}\Pr\left(x\,|P\right) &= \sum_{g=1}^{n_g^P} w_g^P N\left(x;\mu_g^P,\sum_g^P\right)\\ \Pr\left(x\,|Q\right) &= \sum_{g=1}^{n_g^Q} w_g^Q N\left(x;\mu_g^Q,\sum_g^Q\right)\end{aligned} \tag{5}$$

where w_g^P, w_g^Q, μ_g^P, μ_g^Q, Σ_g^P and Σ_g^Q are the weights, means, and covariance matrices of the gth Gaussian of GMMs P and Q, respectively, n_g^P, and n_g^Q are the GMM orders of P and Q, respectively, and $N(x;\mu,\Sigma)$ is the probability density function (pdf) of x given a normal distribution with mean μ and covariance Σ.

Exploiting the linearity of the integral and the mixture model, we get

$$\int_x \Pr(x|P)\log(\Pr(x|Q))dx = \sum_{g=1}^{n_g^P} w_g^P \int_x N\left(x;\mu_g^P,\Sigma_g^P\right)\log(\Pr(x|Q))dx. \quad (6)$$

As no closed-form expression for (6) exists, an approximation must be used. A review of several such approximations can be found in [29]. Note that

$$\begin{aligned}
&\int_x N\left(x;\mu_g^P,\Sigma_g^P\right)\log(\Pr(x|Q))\,dx \\
&= \int_X N\left(x;\mu_g^P,\Sigma_g^P\right)\log\left[\sum_{j=1}^{n_g^Q} w_j^Q N\left(x;\mu_j^Q,\Sigma_j^Q\right)\right]dx \\
&\geq \int_X N\left(x;\mu_g^P,\Sigma_g^P\right)\log\left[w_j^Q N\left(x;\mu_j^Q,\Sigma_j^Q\right)\right]dx \\
&= \log w_j^Q - \frac{1}{2}\left(\mu_g^P-\mu_j^Q\right)^T\left(\Sigma_j^Q\right)^{-1}\left(\mu_g^P-\mu_j^Q\right) \\
&\quad - \frac{1}{2}\log\det\left(\Sigma_j^Q\right) - \frac{1}{2}tr\left(\Sigma_g^P\left(\Sigma_j^Q\right)^{-1}\right) - \frac{D}{2}\log 2\pi
\end{aligned} \quad (7)$$

where D denotes the dimension of the feature space. The inequality in (7) holds for every Gaussian j; therefore, we have n_g^Q closed-form lower bounds. The tightest lower bound is achieved by setting j to (8), as shown at the bottom of the page.

The tightest lower bound is used to approximate the integral $\int_x N\left(x;\mu_g^P,\Sigma_g^P\right)\log(\Pr(x|Q))dx$. The final approximation for $\int_x \Pr(x|P)\log(\Pr(x|Q))\,dx$ is therefore shown in (9) at the bottom of the page. Note that for the case where we define P as an exact representation of test utterance X, our approximation for the negative cross entropy coincides with the exact expression for the average log-likelihood with the exception of using for each frame the most probable Gaussian in Q instead of a summation over all Gaussians. More precisely, in this case, GMM P is defined as $P = \left\{w_i^P = 1/n, \mu_i^P = x_i, \Sigma_i^P = \varepsilon I\right\}_{i=1}^n$, and the negative cross entropy is shown in (10) at the bottom of the page.

C. Estimating the Cross Entropy

Knowing GMM P, the cross entropy between P and Q could be approximated using the technique described in the previous subsection. However, P is unknown. Our first approach, MAP-ACE, is to estimate P from the test utterance, as Q is estimated from the training data of the target speaker, i.e., by estimating a GMM using MAP adaptation from a UBM, though the order of the model may be tuned to the length of the test utterance.

Our second approach, expected ACE (E-ACE), is to calculate the expected negative cross entropy conditioned on the observed test data X as follows:

$$E_P\int_x \Pr(x|P)\log(\Pr(x|Q))\,dx$$

$$\arg\max_j\left\{\log w_j^Q - \frac{1}{2}\left(\mu_g^P-\mu_j^Q\right)^T\left(\Sigma_j^Q\right)^{-1}\left(\mu_g^P-\mu_j^Q\right) - \frac{1}{2}\log\det\left(\Sigma_j^Q\right) - \frac{1}{2}\mathrm{tr}\left(\left(\Sigma_j^Q\right)^{-1}\Sigma_g^P\right)\right\} \quad (8)$$

$$\int_x \Pr(x|P)\log(\Pr(x|Q))dx \cong \sum_{g=1}^{n_g^P} w_g^P \max_j\left\{\log w_j^Q - \frac{1}{2}\left(\mu_g^P-\mu_j^Q\right)^T\left(\Sigma_j^Q\right)^{-1}\left(\mu_g^P-\mu_j^Q\right) - \frac{1}{2}\log\det\Sigma_j^Q - \frac{1}{2}tr\left(\left(\Sigma_j^Q\right)^{-1}\Sigma_g^P\right) - \frac{D}{2}\log 2\pi\right\} \quad (9)$$

$$\begin{aligned}
&\lim_{\varepsilon\to 0}\int_x \Pr(x|P)\log(\Pr(x|Q))dx \\
&= \frac{1}{n}\sum_{i=1}^n \max_j\left\{\log w_j^Q - \frac{1}{2}\left(x_i-\mu_j^Q\right)^T\left(\Sigma_j^Q\right)^{-1}\left(x_i-\mu_j^Q\right) - \frac{1}{2}\log\det\Sigma_j^Q - \frac{D}{2}\log 2\pi\right\} \\
&= \frac{1}{n}\sum_{i=1}^n \max_j \log\left(w_j^Q N\left(x_i;\mu_j^Q,\Sigma_j^Q\right)\right) \cong \mathrm{score}(X|Q)
\end{aligned} \quad (10)$$

$$= \int_P \Pr(P|X) \int_x \Pr(x|P) \log(\Pr(x|Q))\, dx dP. \quad (11)$$

A reasonable assumption would be that the covariance matrices of P are known (fixed for all utterances as $\{\Sigma_g^{\mathrm{UBM}}\}$, the corresponding covariance matrices in the UBM) and that the weights of P are estimated accurately from the test utterance. The mean vector μ_g^P is assumed to be drawn from a normal distribution with known mean μ_g^{UBM} taken from the UBM and covariance matrix $1/r\Sigma_g^{\mathrm{UBM}}$, where r is the relevance factor used for MAP adaptation. The posterior pdf of μ_g^P is therefore

$$\mu_g^P \sim N\left(\frac{n\hat{w}_g^P \hat{\mu}_g^P + r\mu_g^{\mathrm{UBM}}}{n\hat{w}_g^P + r}, \frac{\Sigma_g^{\mathrm{UBM}}}{n\hat{w}_g^P + r}\right) \quad (12)$$

where $\hat{w}_g^P$ denotes the maximum-likelihood (ML) estimated weight of Gaussian g of GMM P from the test utterance, and $\hat{\mu}_g^P$ denotes the corresponding ML estimated mean. The expected negative cross entropy in (11) using the posterior distribution of $\hat{\mu}_g^P$ derived in (12) can be calculated as follows:

$$\begin{aligned} & E_P \int_x \Pr(x|P) \log(\Pr(x|Q))\, dx \\ &= \int_P dP \Pr(P|X) \int_x \Pr(x|P) \log(\Pr(x|Q)) dx \\ &= \int_x \log(\Pr(x|Q)) dx \int_P \Pr(x|P) \Pr(P|X) dP \\ &= \int_x \log(\Pr(x|Q)) \Pr\left(x \middle| \tilde{P}\right) dx \end{aligned} \quad (13)$$

where $\tilde{P}$ is the convolution of the posterior pdf of P ($\Pr(P|X)$) and GMM P which turns out to also be a GMM

$$\begin{aligned} & \Pr\left(x \middle| \tilde{P}\right) \\ &= \sum_{g=1}^{n_g^P} \hat{w}_g^P N\left(x; \frac{n\hat{w}_g^P \hat{\mu}_g^P + r\mu_g^{\mathrm{UBM}}}{n\hat{w}_g^P + r}, \Sigma_g^{\mathrm{UBM}} \left(1 + \frac{1}{n\hat{w}_g^P + r}\right)\right). \end{aligned} \quad (14)$$

The expected negative cross entropy can therefore be approximated by (15), shown at the bottom of the page, with $\tilde{\mu}_g^P = (n\hat{w}_g^P \hat{\mu}_g^P + r\mu_g^{\mathrm{UBM}})/n\hat{w}_g^P + r$ and $\tilde{\Sigma}_g^P = \Sigma_g^{\mathrm{UBM}}\left(1 + 1/(n\hat{w}_g^P + r)\right)$.

D. Special Case #1: Fixed Diagonal Covariance GMMs

In speaker recognition, it is customary to use fixed diagonal covariance matrices GMMs, i.e., diagonal covariance matrices which are trained for the UBM and are not retrained for each speaker. Using fixed diagonal covariance GMMs has the advantages of lower time and memory complexity and also improves robustness. Applying the fixed diagonal covariance assumption results in simpler approximations for the negative cross entropy. The MAP-estimation of the negative cross entropy is shown in (16) at the bottom of the page, where μ_g^P is estimated through MAP-adaptation and is equal to $(n\hat{w}_g^P \hat{\mu}_g^P + r\mu_g^{\mathrm{UBM}})/n\hat{w}_g^P + r$ (using the definitions from previous subsection), and Σ_j is the diagonal covariance matrix for Gaussian j.

The expected negative cross entropy is shown in (17) at the bottom of the page.

$$E_P \int_x \Pr(x|P) \log(\Pr(x|Q))\, dx \cong \sum_{g=1}^{n_g^P} \hat{w}_g^P \max_j \left\{ \log w_j^Q - \frac{1}{2}\left(\tilde{\mu}_g^P - \mu_j^Q\right)^T \left(\Sigma_j^Q\right)^{-1} \left(\tilde{\mu}_g^P - \mu_j^Q\right) - \frac{1}{2}\log\det\Sigma_j^Q - \frac{1}{2}\mathrm{tr}\left(\left(\Sigma_j^Q\right)^{-1} \tilde{\Sigma}_g^P\right) - \frac{D}{2}\log 2\pi \right\} \quad (15)$$

$$\int_x \Pr(x|P) \log(\Pr(x|Q))\, dx \cong \sum_{g=1}^{n_g^P} w_g^P \max_j \left\{ \log w_j^Q - \frac{1}{2}\left(\mu_g^P - \mu_j^Q\right)^T \Sigma_j^{-1} \left(\mu_g^P - \mu_j^Q\right) - \frac{1}{2}\log\det\Sigma_j - \frac{D}{2}(\log 2\pi + 1) \right\} \quad (16)$$

$$E_P \int_x \Pr(x|P) \log(\Pr(x|Q))\, dx \cong \sum_{g=1}^{n_g^P} \hat{w}_g^P \max_j \left\{ \log w_j^Q - \frac{1}{2}\left(\tilde{\mu}_g^P - \mu_j^Q\right)^T \Sigma_j^{-1} \left(\tilde{\mu}_g^P - \mu_j^Q\right) - \frac{1}{2}\log\det\Sigma_j - \frac{\frac{D}{2}}{n\hat{w}_g^P + r} - \frac{D}{2}(\log 2\pi + 1) \right\} \quad (17)$$

E. Special Case #2: Global Diagonal Covariance GMMs

Global diagonal covariance GMMs share a single diagonal covariance matrix among all Gaussians and among all speakers. Using global diagonal covariance GMMs has the advantages of lower time and memory complexity and also may improve robustness when training data are sparse. The reduced modeling power of using a global variance can be compensated by moderately increasing the number of Gaussians. Robustness may be especially important when modeling short test utterances. Applying the Global diagonal covariance assumption results in simple approximations for the negative cross entropy. The MAP-estimation of the negative cross entropy is

$$\begin{aligned}\int_x \Pr(x|P)\log(\Pr(x|Q))\,dx &\cong \sum_{g=1}^{n_g^P} w_g^P \max_j \left\{\log w_j^Q - \frac{1}{2}\left\|\widehat{\mu}_g^P - \widehat{\mu}_j^Q\right\|^2\right\} \\ &\quad - \frac{1}{2}\log\det\Sigma - \frac{D}{2}(\log 2\pi + 1)\end{aligned} \tag{18}$$

where $\widehat{\mu}_g^P$ and $\widehat{\mu}_g^P$ are mean vectors normalized by corresponding global standard deviations:

$$\widehat{\mu}_{g,d}^P = \frac{\mu_{g,d}^P}{\sqrt{\Sigma_{d,d}}} \quad \widehat{\mu}_{g,d}^Q = \frac{\mu_{g,d}^Q}{\sqrt{\Sigma_{d,d}}} \tag{19}$$

and μ_g^P is estimated through MAP-adaptation.

The expected negative cross entropy is

$$\begin{aligned}E_P \int_x \Pr(x|P)\log(\Pr(x|Q))\,dx &\cong \sum_{g=1}^{n_g^P} \hat{w}_g^P \max_j \left\{\log w_j^Q - \frac{1}{2}\left\|\widehat{\mu}_g^P - \widehat{\mu}_j^Q\right\|^2 - \frac{\frac{D}{2}}{n\hat{w}_g^P + r}\right\} \\ &\quad - \frac{1}{2}\log\det\Sigma - \frac{D}{2}(\log 2\pi + 1)\end{aligned} \tag{20}$$

with $\widehat{\mu}_g^P$ and $\widehat{\mu}_g^Q$ defined similar to (19) (except that $\tilde{\mu}_{g,d}^P$ replaces $\mu_{g,d}^P$).

III. Reducing Time Complexity of Approximated Cross Entropy-Based Speaker Recognition

In order to approximate the log-likelihood of a test utterance given a target speaker using ACE, GMMs must be estimated for both the training data and the test utterance. This stage may become a bottleneck and is addressed in Section IV. All the variants presented in Section II for ACE can be generalized as

$$\int_x \Pr(x|P)\log(\Pr(x|Q))dx \cong \sum_{g=1}^{n_g^P} w_g^P \max_j \mathrm{sim}(P_g, Q_j) \tag{21}$$

where P_g and Q_j are the gth Gaussian of P and the jth Gaussian of Q, respectively. The difference between the various variants lies in the definition of function $\mathrm{sim}(P_g, Q_j)$ which measures the similarity between P_g and Q_j.

The time complexity of approximating the cross entropy between two pretrained GMMs [(9), (15)] is $O\left(n_g^P n_g^Q D^3\right)$ for the general case (n_g^P and n_g^Q are the GMM order of P and Q, respectively, D is the dimension of the feature space). For the special cases of diagonal covariance matrices [(16)–(18), (20)], the time complexity is $\mathrm{O}\left(n_g^P n_g^Q D\right)$. We use the following techniques to improve the time complexity of the proposed ACE methods.

A. Top-N Pruning

Top-N pruning exploits the property that both GMM P and GMM Q are adapted from UBMs U_P and U_{Q}, respectively, and therefore prior knowledge on the parameters of P and Q exists. Furthermore, if the mean of the gth Gaussian of U_P is very distant from the mean of the jth Gaussian of U_{Q}, then the gth Gaussian of P and the jth Gaussian of Q are most probably distant. This property can be used by creating a Gaussian short-list [a list which specifies the subset of the Gaussians which are most likely to maximize (8)] for every Gaussian in U_P. The Gaussian short-list of the gth Gaussian of U_P points to the top-N closets (in the sense of function sim) Gaussians in U_{Q}. Given GMMs P and Q, (21) can be approximated by

$$\int_x \Pr(x|P)\log(\Pr(x|Q))dx \cong \sum_{g=1}^{n_g^P} w_g^P \max_{j\in Lg} \mathrm{sim}(P_g, Q_j) \tag{22}$$

where Lg is the Gaussian short-list of the gth Gaussian of U_P. The time complexity of the approximation using the top-N technique is $\mathrm{O}\left(n_g^P N S\right)$, where S is the time complexity of function sim, which is D^3 for the general case and D for diagonal covariance matrices. Note that similar principles are used for GMM frame-based top-N scoring [19].

B. Gaussian Pruning

Equation (21) may be viewed as approximating the negative cross entropy by an empirical expectation of function $\max \mathrm{sim}(P_g, Q_j)$ with respect to a sample of Gaussians (the components of P). In order to obtain a quick approximation to the negative cross entropy, a subset of the components of P can be used to calculate an empirical expectation. The optimal subset would be the set of Gaussians with the highest weights

$$\int_x \Pr(x|P)\log(\Pr(x|Q))dx \cong \frac{1}{\sum_{g=1,w_i>K/n_g^P}^{n_g^P} w_g^P} \sum_{g=1,w_i>K/n_g^P}^{n_g^P} w_g^P \max_{j\in Lg} \mathrm{sim}(P_g, Q_j). \tag{23}$$

for some appropriate value of K.

C. Two-Phase Recognition

Excessive Gaussian pruning may degrade recognition accuracy. This degradation may be mended by a second phase of verification which is performed by rescoring (without Gaussian pruning) a small subset of the test sessions which pass the first phase with a relatively high score. For example, if we are interested in a false acceptance rate of 1%, we can set the false

acceptance rate of the first phase to a somewhat higher rate and reduce it using a rescoring phase.

IV. Fast GMM–UBM Decoding and Fast GMM–UBM MAP Adaptation

In this section, we describe a technique for accelerating the procedure of finding the top-N best scoring Gaussians for a given frame. This technique is used by both classic GMM scoring [19] and by MAP-adaptation of a GMM which is used by the ACE algorithm. The goal of the top-N best scoring technique, given a UBM and a frame, is to find the top-N scoring Gaussians. Note that a small fraction of errors may be tolerated. Finding the top-N best scoring Gaussians is usually done by scoring all Gaussians in the UBM and then finding the N maximal scores. Our technique introduces an indexing phase in which the Gaussians of the UBM are examined and associated with clusters defined by a vector-quantizer. During recognition, every frame is first associated with a single cluster, and then only the Gaussians mapped to that cluster are scored. Note that a Gaussian is usually mapped to many clusters. In order to be able to locate the cluster quickly, we design the vector-quantizer to be structured as a tree (VQ-tree). A similar approach which does not exploit a tree structure was investigated in [30] and in [31]. Hierarchical GMM approaches [32] do exploit a tree structure and achieve considerable complexity reduction. However, using a VQ-tree enables extremely fast decoding and can be tuned to achieve any desired level of accuracy.

A. VQ-Tree Training

Following is a description of the VQ-tree training procedure.

1) Initialize the tree by inserting all the development set vectors into a single leaf (the root).
2) Until the number of leaves reaches a requested threshold: Split the most distorted leaf by performing the following steps.
   a) Initialize a k-means VQ by picking randomly two training vectors ($k = 2$).
   b) Train the k-means VQ using the Mahalanobis distance with the covariance matrix of the whole training dataset.
   c) Partition the training vectors according to the VQ into two leaves.

The distortion of a leaf is defined as the sum of the squared Mahalanobis distances between every vector in the leaf and the center of the leaf.

B. Mapping Gaussians to Clusters

The goal of this stage is to create for each cluster C a short-list of Gaussians G_c defined as

$$G_c = \left\{ g \left| \frac{\int_{x \in C} 1_{g \in \text{top}N(x)} dx}{\int_{x \in C} dx} > \varepsilon \right. \right\} \tag{24}$$

where x is a feature vector. Equation (24) assigns a Gaussian g to the short-list of cluster C if the probability for a random feature vector associated to cluster C to have Gaussian g in its top-N Gaussians exceeds a predefined threshold ε.

The following is a description of the algorithm for mapping Gaussians to clusters.

1) For every feature vector in development dataset:
   a) compute the top-N scoring Gaussians;
   b) locate the matching leaf in VQ-tree;
   c) accumulate the top-N scoring Gaussians in the statistics of the matching leaf.
2) Create for every leaf a Gaussian short-list according to the accumulated statistics and (24).

C. Finding Top-N Gaussians for a Feature Vector

Following is a description of the top-N decoding procedure.

1) Given a feature vector x, find its cluster in VQ-tree (C).
2) Score all the Gaussians in the Gaussian short-list of cluster C.
3) Find the top-N scoring Gaussians.

D. Time and Memory Complexity

Given a UBM of order g and a VQ-tree with l leaves, the expected leaf depth in the tree denoted by e is $\text{O}(\log(l))$. Let s denote the expected size of a Gaussian short-list, and let d denote the dimension of the feature space. The time complexity of the baseline is $\text{O}(gd)$. The time complexity of the VQ-tree based algorithm is $\text{O}((e+s)d)$. The speedup factor achieved is therefore $g/(e+s)$. Experiments in Section V indicate that a speedup factor of about 40 can be achieved compared to a standard baseline.

The amount of memory required for storing of the index is $O(ld)$ for storing the VQ-tree and $O(ls)$ for storing the Gaussian short-lists.

E. Accuracy

The expected miss probability of the VQ-tree algorithm for a Gaussian in the top-N list is guaranteed to be not greater than ε (24). Therefore, the algorithm can be tuned to have a requested level of accuracy.

V. Experiments

A. Datasets and Protocol

The development dataset consists of a subset of the switchboard-2 corpus [33] and a subset of the NIST-2003-SRE dataset [34]. Development dataset was used to train the UBM, the VQ-tree, and for T/Z/ZT-norm modeling. The core set of the NIST-2004-SRE dataset [35] was used for evaluation, but contrary to the NIST protocol, all male target models were scored against all available male test files, and all female target models were scored against all available female test files. This was done in order to increase the number of trials. The data set consists of 616 one-sided single conversations for training 616 target models, and 1174 one-sided test conversations, resulting in 127 968 male trials (of which 1066 are the same speaker trials) and 242 144 female trials (of which 1314 are the same speaker trials). All conversations are about five minutes long and originate from various channels, handset types, and languages. Most of the experiments were conducted using only male models and test sessions. Selected results were validated on the female dataset as well.

In order to validate the success of the proposed techniques for scoring short test sessions, three additional testing conditions of 30-, 10-, and 3-s utterances (after silence removal) were defined.

We report two performance measures. The first one is equal error rate (EER) and the second one is min-DCF [35] which is the minimal value of the detection cost function (DCF) defined as

$$\mathrm{DCF} = 0.1 * \Pr(\text{Misdetection}) + 0.99\ \Pr(\text{False acceptance}). \tag{25}$$

For selected experiments, we present DET curves [36], which represent the tradeoff between speaker misdetection probability and false-acceptance probability.

For the male subset, assuming all trials are independent, the 95% confidence interval for the EER measure is approximately 5% relative for the relevant range of EER values (10%–20%). The corresponding confidence interval for min-DCF is experiment-dependent, but in practice is on the order of 5% as well.

For all systems, the evaluated raw scores are normalized by applying Z-norm followed by T-norm (ZT-norm), which proved to be superior to T-norm, Z-norm, and TZ-norm. The gender of the normalization models/sessions is matched to the gender of the test utterances. 250 normalization sessions per gender were chosen from the development data and used for both Z-norm and T-norm.

B. Baseline GMM System

The baseline GMM system was inspired by the GMM-UBM system described in [1] and [37]. The front-end of the recognizer consists of calculation of Mel-frequency cepstrum coefficients (MFCCs) according to the ETSI standard [38]. An energy-based voice activity detector is used to locate and remove nonspeech segments, and the cepstral mean of the speech segments is calculated and subtracted. The final feature set is 13 cepstral coefficients + 13 delta cepstral coefficients extracted every 10 ms using a 25-ms window. Feature warping with a 300 frame window is applied as described in [39].

A gender-independent (GI) UBM and two gender-dependent (GD) UBMs (adapted from the GI UBM) were trained using the first 20 s of 500 sessions from the development data. The GI-UBM is used as a prior for GMM adaptation, and the GD-UBMs are used for score normalization. The order of the GMMs used for all experiments is 2048. Both fixed diagonal covariance matrix GMMs and global diagonal covariance matrix GMMs were evaluated, with and without weight adaptation. Top-$N(N = 10)$ fast scoring was used for GMM scoring [19]. Note that the performance using top-5 was worse than using top-10. In the scoring stage, the log likelihood of each conversation side given a target speaker was normalized by the GD-UBM score and divided by the length of the conversation before ZT-normalization.

Table I shows a comparison of various configurations tested on the male subset of the evaluation dataset. The EER (10.68%) and min-DCF (0.0412) achieved for full session testing with fixed covariance matrices and Gaussian mean adaptation are competitive to comparable results published such as in [40]. Overall it can be concluded that fixed (Gaussian-dependent) covariance matrices slightly outperform global covariance matrices for long (full one-side, 30-s) test sessions, whereas the opposite is true for short (ten-sided, 3-s) test utterances. A similar phenomenon was observed for weight adaptation as it degrades performance for all but the shortest (3-s) test sessions for which it slightly improved performance.

TABLE I
EER AND MIN-DCF FOR VARIOUS CONFIGURATIONS OF THE BASELINE GMM SYSTEM FOR FULL, 30-, 10-, AND 3-S-LONG TEST SESSIONS (ON THE MALE SUBSET NIST-2004-SRE). BOLDFACE FONT IS USED TO HIGHLIGHT PERFORMANCE OF CHOSEN BASELINES

Test length (sec)	Covar. matrices	Adaptation: means only		Adaptation: means + weights	
		EER (%)	Min-DCF	EER (%)	Min-DCF
Full	Global	**10.79**	**0.0407**	11.35	0.0412
Full	Fixed	10.68	0.0412	11.35	0.0421
30sec	Global	**11.21**	**0.0423**	11.61	0.0431
30sec	Fixed	11.10	0.0421	11.91	0.0439
10sec	Global	**12.57**	**0.0491**	12.95	0.0493
10sec	Fixed	12.66	0.0488	13.51	0.0496
3sec	Global	18.30	0.0655	**17.73**	**0.0649**
3sec	Fixed	18.21	0.0669	18.04	0.0659

Under the classic GMM framework, the optimal GMM configuration may be also a function of the test session duration. Using top-1 approximation, where only the most likely Gaussian is considered, the likelihood of a session given a GMM consists of two components. The first component is a function of the Gaussian weights, and the second component is a function of the Gaussian means and covariance matrices. It is possible that the weight-related component degrades accuracy for long test sessions and improves accuracy for short test sessions, probably because for scoring short test sessions the information encapsulated in the Gaussian weights is relatively more important than for longer test sessions. GMMs with global covariance matrices are more suitable for scoring short sessions probably because their reduced modeling capability is compensated by their excess smoothness, which is important when scoring short sessions.

Taking into account that most of the differences in accuracy are statistically insignificant and considering efficiency, global covariance matrices without weight adaptation was chosen as the configuration for the GMM baseline for the three longest test conditions (full and 30 and 10 s) and global covariance matrices with weight adaptation was chosen as the configuration for the GMM baseline for the 3-s test condition.

C. ACE Compared to the GMM Baseline

A first set of experiments was carried out in order to assess the validity of the concept of ACE as an approximation for the frame-based likelihood scoring. Table II compares the accuracy of the baseline GMM system with a system based on a Monte-Carlo approximation of the cross entropy. The Monte-Carlo based algorithm approximates the cross entropy between GMMS P and Q by averaging the log-likelihood conditioned on Q of random vectors drawn from model P. P is estimated by MAP adaptation of a UBM as Q is estimated for the target speaker. The Monte Carlo approximation was found impractical as the number of random vectors per cross entropy calculation

TABLE II
EER AND MIN-DCF FOR THE BASELINE GMM SYSTEM COMPARED TO MONTE CARLO-BASED ACE FOR FULL ONE-SIDE TEST SESSIONS (ON THE MALE SUBSET NIST-2004-SRE)

System	EER (%)	minDCF
GMM baseline	10.79	0.0407
ACE, Monte-Carlo: 100,000 vectors/score	11.17	0.0425
ACE, Monte-Carlo 20,000 vectors/score	11.26	0.0427
ACE, Monte-Carlo 5,000 vectors/score	12.57	0.0441

TABLE III
EER AND MIN-DCF FOR MAP-BASED ACE COMPARED TO THE BASELINE GMM SYSTEM FOR FULL, 30-, 10-, AND 3-S-LONG TEST SESSIONS (ON THE MALE SUBSET NIST-2004-SRE)

Test length (sec)	System	EER (%)	minDCF
Full	GMM baseline	10.79	0.0407
	MAP-ACE	11.06	0.0400
	E-ACE	10.79	0.0400
30sec	GMM baseline	11.21	0.0423
	MAP-ACE	11.16	0.0419
	E-ACE	10.95	0.0419
10sec	GMM baseline	12.57	0.0491
	MAP-ACE	12.57	0.0497
	E-ACE	12.57	0.0491
3sec	GMM baseline	17.73	0.0649
	MAP-ACE	17.82	0.0646
	E-ACE	17.82	0.0646

required to get a good approximation was found to be high. Note that although the accuracy of the Monte Carlo-based approximation converges closely to the accuracy of the baseline, the correlation coefficient between the log-likelihoods produced by both systems was measured to be only 0.54. However, the correlation coefficient between the like log-likelihood ratios (after normalization with likelihood of UBM) was measured as 0.986.

Next, MAP-based estimation of the approximated cross (MAP-ACE) entropy as derived in (9) was used and compared to the GMM baseline. Table III compares the accuracy of the baseline GMM system with MAP-based ACE using top-1 pruning. Overall, no statistically significant degradation was found compared to the optimized GMM baseline. The corresponding DET curve is presented in Fig. 1. The same experiments were done for the female subset of NIST-2004-SRE. The results are listed in Table IV and indicate no overall degradation. For both subsets, no significant improvement was found when using top-N pruning with N greater than 1.

E-ACE was tested on the male subset and compared to MAP-ACE. The results are also listed in Table III and indicate an improvement compared to MAP-ACE.

D. Gaussian Pruning

Experiments with Gaussian Pruning were conducted on the male subset of NIST-2004-SRE. The results are summarized in Table V. It was found that a speedup factor of 3 can be achieved with a statistically insignificant degradation in accuracy. Larger speedup factors can be achieved with a significant degradation that can be partly recovered using a second rescoring phase.

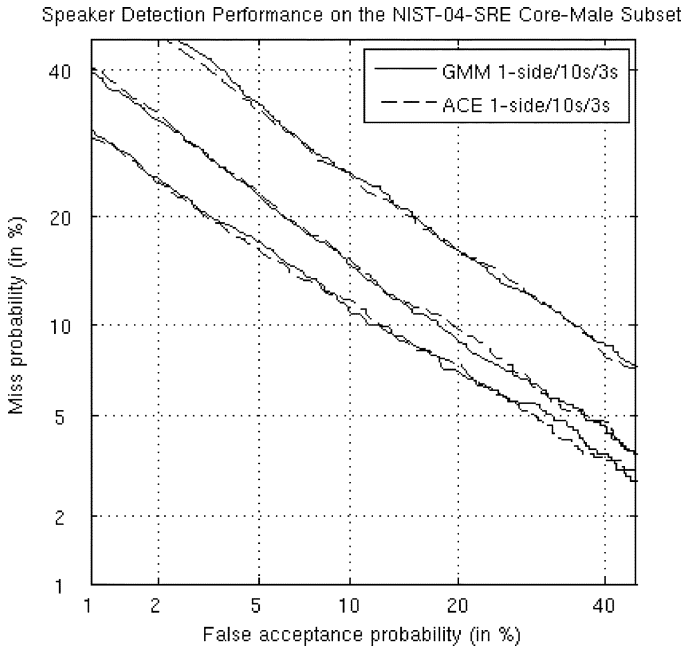

Fig. 1. GMM baseline compared to the ACE system on the NIST-2004 corpus for one-side and 10- and 3-s test utterances. Results for 30-s test utterances are close to those of one-side test utterances and therefore were omitted for the sake of clarity. No significant difference in accuracy is observed for any test condition.

TABLE IV
EER AND MIN-DCF FOR MAP-BASED ACE COMPARED TO THE BASELINE GMM SYSTEM FOR FULL, 30-, 10-, AND 3-S-LONG TEST SESSIONS (ON THE FEMALE SUBSET NIST-2004-SRE)

Test length (sec)	System	EER (%)	minDCF
Full	GMM baseline	11.35	0.0410
	MAP-ACE	11.19	0.0397
30sec	GMM baseline	12.48	0.0436
	MAP-ACE	11.80	0.0423
10sec	GMM baseline	13.63	0.0509
	MAP-ACE	13.24	0.0509
3sec	GMM baseline	19.33	0.0701
	MAP-ACE	20.55	0.0739

TABLE V
EER AND MIN-DCF FOR THE MAP-ACE GMM SYSTEM WITH VARIOUS LEVELS OF GAUSSIAN PRUNING FOR ONE-SIDE SESSIONS (ON THE MALE SUBSET NIST-2004-SRE)

Pruning factor (K)	Speedup factor	EER (%)	minDCF
0	1.0	11.06	0.0400
1	2.9	11.16	0.0420
2	7.9	11.72	0.0435
3	18.9	12.94	0.0478
4	40.9	14.07	0.0541

E. Two-Phase Scoring

A two-phase scoring system was developed based on the ACE algorithm with top-1 pruning. The first phase includes Gaussian pruning with $K = 3$, and the second phase does not include Gaussian Pruning. A threshold of 1.0 on the ZT-normalized scores was set in order to pass only a small percentage (14%) of test sessions to the second scoring phase. The results for the

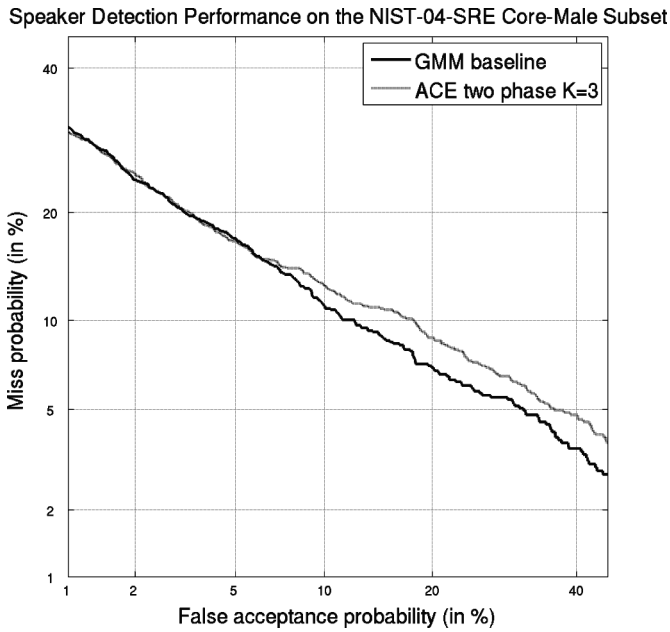

Fig. 2. GMM baseline compared to a two-phase ACE-based system with Gaussian pruning factor of $K = 3$ on the NIST-2004 corpus for one-side, test utterances. No significant difference in accuracy is observed for false acceptance probability lower than 7%

two-phase scoring system are presented in Fig. 2. A speedup factor of 5 was achieved without any degradation for false acceptance of 7% and lower.

F. Fast Top-N Decoding Using VQ-Tree

A VQ-tree with 10 000 leaves was trained on the SPIDRE corpus (a subset of switchboard I). ε was set to 0.0001. For a UBM-GMM of 2048 Gaussians, the average size of a Gaussian short-list was 40. The expected depth of a leaf in the VQ-tree is 17. The effective speedup factor is therefore 37. The amount of memory required is 1 MB for the VQ-tree and 800 kB for the Gaussian short-list.

On the NIST-2004-SRE, no degradation in accuracy was observed when using the VQ-tree either for GMM adaptation or when it was used to accelerate the baseline GMM for both MAP-adaptation and scoring.

VI. Time Complexity Analysis

In this section, two tasks are considered. The first task is speaker identification where multiple speakers may be hypothesized, and the second one is speaker retrieval. For both tasks, t denotes the average number of test frames after silence removal (12 000 on average for one-side sessions), g denotes the GMM order (2048), and d denotes the dimension of the feature space (26). The GMM baseline is assumed to use top-N decoding with $N = 10$. Other speedup techniques, reviewed in Sections I and IV, were not used by the baseline because typically these techniques (e.g., frame decimation and Gaussian clustering) have tradeoffs between accuracy and efficiency and are not standard. For the ACE-based systems, we use VQ-tree-based GMM-UBM adaptation with a speedup factor denoted by v (37). We analyze both an ACE system with top-N pruning

TABLE VI
Time Complexity Analysis: ACE Systems Compared To Baseline GMM

System	Time complexity	Ops. (x10^6)	Speedup factor[a]
Speaker Identification			
GMM baseline	gdt+Ndt	640+3.1n	1
ACE	gdt/v+ngd	17+0.05n	62
Two-phase ACE	gdt/v+ngd/p	17+0.01n	310
Speaker Retrieval			
GMM baseline	gdt	640	1
ACE	gd	0.05	12,800
Two-phase ACE	gd/p	0.01	64,000

[a] For speaker identification n$\rightarrow\infty$

$(N = 1)$ and a two-phase scoring system described in Section V with a speedup factor of $p(5)$.

For speaker identification, we assume a speaker population of size n. The front-end processing time and training time is excluded from the analysis.

For speaker retrieval, we assume only a single speaker is retrieved, and we assume that for both the baseline GMM and the ACE systems, T-norm parameters for the sessions in the archive are already precomputed in the indexing phase and therefore are not part of the retrieval complexity. The training time for the target speaker is excluded from the analysis. The time complexity is computed per single session in the archive.

The time complexity analysis is presented in Table VI. The second column in Table VI lists the time complexity of the various systems as a function of the system parameters, and the third column lists the time complexity using typical parameter values. For example, a two-phase ACE-based speaker identification system requires gdt/v operations for parameterization of test utterance with a GMM, and ngd/p operations for approximating the log-likelihoods of a test utterance given n target models. For the typical values of the parameters listed above this will result in 17 million operations for GMM parameterization and 10 000 n operations for n log-likelihood approximations.

VII. GMM Compression

Our proposed algorithm for speaker retrieval requires storing a GMM for every audio file in the archive. In order to reduce the size of the index, the GMMs must be compressed.

In [41] a GMM compression algorithm was introduced. The main idea is to exploit the fact that all GMMs are adapted from the same UBM. For a given GMM, the parameters that are significantly different from the UBM are quantized using the UBM as a reference. The quantization is done independently for every mean coefficient, variance coefficient, and every weight.

In this paper, we propose a different way to compress GMMs. We optimize our compression algorithm with respect to the speaker retrieval task and the ACE algorithm. We need only to compress the weights and the mean vectors. The weights are compressed by similar techniques as in [40]. The means however are compressed by using vector quantization. We compute GMMs for all the sessions in a development set and then subtract from each mean vector the corresponding UBM mean vector. We cluster the resulting vectors into 60 000 (in order to represent cluster indices with 2 B) clusters. In order to compress a GMM, we just replace every mean vector by its cluster index. This compression algorithm results in a 1:50

compression rate but with some degradation in accuracy. In order to eliminate the degradation in accuracy, we first locate badly quantized Gaussians (10% in average). These Gaussians are located by calculating for every mean vector the product of its weight and its quantization error. Badly quantized Gaussians are characterized by a high product. The badly quantized Gaussians are compressed by quantization of every coefficient independently into 4 bits. The compression factor of the described algorithm is 1:30 (7 kB per GMM) without any notable degradation.

VIII. Conclusion

In this paper, we have presented an algorithm for efficient and accurate speaker recognition. The algorithm is based on ACE and is useful for both identification of a large population of speakers and for speaker retrieval. For example, we get a speedup factor of 52 for identification of 100 speakers (target and T-norm) and a speedup factor of 135 for 1000 speakers. For the speaker retrieval task we get a speedup factor of 64 000. We verified that our techniques are also suitable when testing on short test sessions. Finally, we presented an algorithm for GMM compression which is used to compress the index built by our speaker retrieval algorithm.

More generally, the ACE framework may be used to replace other GMM-based algorithms. Lately, the ACE method has been successfully used for efficient language identification [15], and efficient speaker diarization [42].

This paper shows that a GMM trained for a test utterance is approximately a sufficient statistic for the classic GMM-based log-likelihood. This result is a theoretical justification for the GMM-supervector [10]–[15] framework where sessions (both training and test) are projected into a high-dimensional space using the parameters of the GMMs trained for the sessions, and modeling is done in the high-dimensional space named the GMM-supervector space. Note that when using the GMM-supervector approach, kernel-based methods can also be incorporated [14]. In addition to establishing a theoretical basis for the GMM-supervector approach, the speedup techniques presented in this paper (Gaussian pruning and fast GMM-UBM MAP-adaptation) can be used to reduce the time complexity of GMM-supervector-based approaches. Furthermore, current GMM-supervector-based approaches which usually model only the GMM means are probably suboptimal as the information encapsulated in the GMM weights has been shown to be important for speaker recognition [44]. Moreover, current GMM-supervector-based approaches are not very successful in coping with short sessions [13], probably because GMM weights are important in that case. The ACE framework gives a theoretically-based method for combining GMM means and weights and may be a good starting point for a development of an improved GMM-supervector approach.

References

[1] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process.*, vol. 10, no. 1-3, pp. 19–41, 2000.

[2] R. C. Rose and D. A. Reynolds, "Text-independent speaker identification using automatic acoustic segmentation," in *Proc. ICASSP*, 1990, pp. 293–296.

[3] D. A. Reynolds, "A Gaussian mixture modeling approach to text-independent speaker identification," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, 1992.

[4] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 1, pp. 72–83, Jan. 1995.

[5] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Comput. Speech Lang.*, vol. 20, no. 2-3, pp. 210–229, 2006.

[6] D. E. Sturim, D. A. Reynolds, R. B. Dunn, and T. F. Quatieri, "Speaker verification using text-constrained Gaussian mixture models," in *Proc. ICASSP*, 2002, pp. 677–680.

[7] A. Stolcke, L. Ferrer, and S. Kajarekar, "Improvements in MLLR-transform-based speaker recognition," in *Proc. ISCA Odyssey Workshop*, 2006.

[8] Aronowitz, D. Burshtein, and A. Amir, "Text independent speaker recognition using speaker dependent word spotting," in *Proc. Interspeech*, 2004, pp. 1789–1792.

[9] A. Hatch, B. Peskin, and A. Stolcke, "Improved phonetic speaker recognition using lattice decoding," in *Proc. ICASSP*, 2005, pp. 169–172.

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

[11] H. Aronowitz, D. Burshtein, and A. Amir, "A session-GMM generative model using test utterance Gaussian mixture modeling for speaker verification," in *Proc. ICASSP*, 2005, pp. 729–732.

[12] H. Aronowitz, D. Irony, and D. Burshtein, "Modeling intra-speaker variability for speaker recognition," in *Proc. Interspeech*, 2005, pp. 2177–2180.

[13] R. Vogt and S. Sridharan, "Experiments in session variability modeling for speaker verification," in *Proc. ICASSP*, 2006, pp. 897–900.

[14] 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 *Proc. ICASSP*, 2006, pp. 97–100.

[15] E. Noor and H. Aronowitz, "Efficient language identification using Anchor models and support vector machines," in *Proc. ISCA Odyssey Workshop*, 2006, pp. 1–6.

[16] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Process.*, vol. 10, pp. 42–54, 2000.

[17] I. M. Chagolleau and N. P. Vallès, "Audio indexing: What has been accomplished and the road ahead," in *Proc. 6th Joint Conf. Inf. Sci.*, 2002, pp. 911–914.

[18] J. Makhoul, F. Kubala, T. Leek, L. Daben, N. Long, R. Schwartz, and A. Srivastava, "Speech and language technologies for audio indexing and retrieval," *Proc. IEEE*, vol. 88, no. 8, pp. 1338–1353, Aug. 2000.

[19] J. McLaughlin, D. A. Reynolds, and T. Gleason, "A study of computation speed-ups of the GMM-UBM speaker recognition system," in *Proc. Eurospeech*, 1999, pp. 1215–1218.

[20] D. E. Sturim, D. A. Reynolds, E. Singer, and J. P. Campbell, "Speaker indexing in large audio databases using anchor models," in *Proc. IEEE ICASSP*, 2001, pp. 429–432.

[21] Y. Mami, D. Charlet, and F. Lannion, "Speaker identification by anchor models with PCA/LDA post-processing," in *Proc. ICASSP*, 2004, pp. 180–183.

[22] M. Collet, D. Charlet, and F. Bimbot, "A correlation metric for speaker tracking using Anchor models," in *Proc. ICASSP*, 2005, pp. 713–716.

[23] M. Collet, Y. Mami, D. Charlet, and F. Bimbot, "Probabilistic Anchor models approach for speaker verification," in *Proc. Interspeech*, 2005, pp. 2005–2008.

[24] H. Aronowitz, D. Burshtein, and A. Amir, "Speaker indexing in audio archives using test utterance Gaussian mixture modeling," in *Proc. ICSLP*, 2004, pp. 609–612.

[25] H. Aronowitz, D. Burshtein, and A. Amir, "Speaker indexing in audio archives using Gaussian mixture scoring simulation," in *MLMI: Proceedings of the Workshop on Machine Learning for Multimodal Interaction*. New York: Springer-Verlag LNCS, 2004, pp. 243–252.

[26] H. Aronowitz and D. Burshtein, "Efficient speaker identification and retrieval," in *Proc. Interspeech*, 2005, pp. 2433–2436.

[27] M. Schmidt, H. Gish, and A. Mielke, "Covariance estimation methods for channel robust text-independent speaker identification," in *Proc. ICASSP*, 1995, pp. 333–336.

[28] W. H. Tsai, W. W. Chang, Y. C. Chu, and C. S. Huang, "Explicit exploitation of stochastic characteristics of test utterance for text-independent speaker identification," in *Proc. Eurospeech*, 2001, pp. 771–774.
[29] J. Hershey and P. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in *Proc. ICASSP*, 2007, pp. 317–320.
[30] D. B. Paul, "An investigation of Gaussian shortlists," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding*, 1999, pp. 209–212.
[31] A. Chan, J. Sherwani, R. Mosur, and A. Rudnicky, "Four-layer categorization scheme of fast GMM computation techniques in large vocabulary continuous speech recognition systems," in *Proc. ICSLP*, 2004, pp. 289–292.
[32] B. Xiang and T. Berger, "Efficient text-independent speaker verification with structural Gaussian mixture models and neural network," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 5, pp. 447–456, 2003.
[33] "Switchboard 2 Phase II," Univ. Pennsylvania, Philadelphia, PA. [Online]. Available: http://www.ldc.upenn.edu/Catalog/docs/Switchboard2_Phase2
[34] "The NIST Year 2004 Speaker Recognition Evaluation Plan," NIST, Gaithersburg, MD. [Online]. Available: http://www.nist.gov/speech/tests/spk/2003
[35] "The NIST Year 2004 Speaker Recognition Evaluation Plan," NIST, Gaithersburg, MD. [Online]. Available: http://www.nist.gov/speech/tests/spk/2004
[36] A. Martin, D. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proc. Eurospeech*, 1997, pp. 1895–1898.
[37] D. A. Reynolds, "Comparison of background normalization methods for text-independent speaker verification," in *Proc. Eurospeech*, 1997, pp. 963–966.
[38] *Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Front-End Feature Extraction Algorithm; Compression Algorithms*, [Online]. Available: http://www.etsi.org/stq
[39] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. ISCA Odyssey Workshop*, 2001, pp. 213–218.
[40] S. S. Kajarekar, L. Ferrer, E. Shriberg, K. Sonmez, A. Stolcke, A. Venkataraman, and J. Zheng, "SRI's 2004 NIST speaker recognition evaluation system," in *Proc. ICASSP*, 2005, pp. 173–176.
[41] D. A. Reynolds, "Model compression for GMM based speaker recognition systems," in *Proc. Eurospeech*, 2003, pp. 2005–2008.
[42] H. Aronowitz, "Trainable speaker diarization," in *Proc. Interspeech*, 2007, to be published.
[43] H. Aronowitz, "Speaker recognition using kernel-PCA and intersession variability modeling," in *Proc. Interspeech*, 2007, to be published.

Hagai Aronowitz received the B.Sc. degree in computer science, mathematics, and physics from the Hebrew University, Jerusalem, Israel, in 1994, and the M.Sc. (summa cum laude) and Ph.D. degrees in computer science from Bar-Ilan University, Ramat-Gan, Israel, in 2000 and 2006, respectively.

In 2006, he joined the Advanced LVCSR Group, IBM T. J. Watson Research Center, Yorktown Heights, NY, as a Postdoctoral Fellow. His research interests include speech processing and machine learning.

David Burshtein (M'92–SM'99) received the B.Sc. and Ph.D. degrees in electrical engineering from Tel-Aviv University, Tel-Aviv, Israel, in 1982 and 1987, respectively.

From 1988 to 1989, he was a Research Staff Member in the Speech Recognition Group, IBM T. J. Watson Research Center, Yorktown Heights, NY. In 1989, he joined the School of Electrical Engineering, Tel-Aviv University, where he is currently an Associate Professor. His research interests include information theory and signal processing.