Design of Gaussian Mixture Models Using Matching Pursuit

Khaled Ben Fatma and A. Enis Çetin
Bilkent University
Department of Electrical and Electronics Engineering,
Bilkent, Ankara, TR-06533 Turkey
khaled@ee.bilkent.edu.tr cetin@ee.bilkent.edu.tr

Abstract **- In this paper, a new design algorithm for estimating the parameters of Gaussian Mixture Models is presented. The method is based on the matching pursuit algorithm. Speaker Identification is considered as an application area. The estimated GMM performs as good as the EM algorithm based model. Computational complexity of the proposed method is much lower than the EM algorithm.**

I. INTRODUCTION

A Gaussian mixture density is defined as a weighted sum of different Gaussian component densities. Gaussian Mixture Models (GMM) have been recently used in many applications as an efficient method for modeling arbitrary densities [1]. GMM was shown to provide a smooth approximation to the underlying long-term sample distribution of observations obtained from experimental measurements [1]. This is due to the fact that a linear combination of Gaussian basis is capable of representing a large class of sample distributions, in addition to the observation that most natural phenomena tend to have a Gaussian distribution.

There are several techniques available for estimating the parameters of a GMM. By far the most popular and well-established method is maximum likelihood (ML) estimation. The aim of ML estimation is to find the model parameters which maximize the likelihood of the GMM, given the training data. This usually leads to a nonlinear function of the parameters and it may not be possible to find the optimum solution. However, ML parameter estimates can be obtained iteratively using a special case of the Expectation-Maximization (EM) algorithm [2].

The EM algorithm usually leads to good estimates of the GMM parameters, however its computational complexity is very high, which makes it not suitable for real-time operations especially when the amount of training data is huge.

In this paper a new method for estimating the parameters of a GMM using a modified version of the matching pursuit algorithm is introduced. The matching pursuit based estimation method has a much lower computational cost compared to the EM algorithm, while assuring a good modeling accuracy, as good as the EM-based model.

Pursuit algorithms are generally used to decompose arbitrarily signals. Decomposition vectors are chosen depending upon the signal properties. These algorithms usually have a high computational complexity. The matching pursuit introduced by Mallat and Zhong reduces the computational complexity with a greedy strategy [4]. It is closely related to projection pursuit algorithms used in statistics and to shape-gain vector quantizations. Vectors are selected one by one from a dictionary, while optimizing the signal approximation at each step. In this paper, a modified MP algorithm is used as an alternative method for estimating the parameters of a GMM.

Speaker recognition is an important application where GMMs have proven to be very efficient [1]. The speech spectrum based parameters are very effective for speakers modeling. For example, the distribution of mel-cepstral parameters are represented by GMMs in [1]. The use of GMMs for modeling speaker identity is motivated by the interpretation that the Gaussian components represent some general speaker-dependent spectral shapes.

In this paper, speaker modeling based on GMMs is chosen as an application for evaluating the performance of the matching pursuit based method introduced here. Experimental results for text-independent speaker identification are presented and compared to results previously obtained by the EM based method.

The rest of the paper is organized as follows. In the next section, we describe briefly the general form of a GMM and the EM algorithm used for estimating its parameters, then we introduce the new idea of using a modified matching pursuit algorithm for estimating the parameters of a

GMM. Section III presents how this idea can be implemented for the task of speaker identification. Finally, in section IV experimental results are provided and compared to those obtained by the EM-based model.

II. ESTIMATION OF GMM PARAMETERS

A. Gaussian Mixture Models (GMMs)

Given an arbitrary D-dimensional random vector $\vec{x}$, a Gaussian mixture density of *M* components is defined as a weighted sum of individual *D*-variate Gaussian densities $b_i(\vec{x})$, *i* = 1, … , *M* as follows

$$p(\vec{x} \mid \lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x}) \qquad (1)$$

where p_i , i =1,…., M are the weights of the individual components and are constrained by

$$\sum_{i=1}^{M} p_i = 1 \qquad (2)$$

The *D*-variate Gaussian function, $b_i(\vec{x})$, is given by

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2}\left|\Sigma_i\right|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)'\Sigma_i^{-1}(\vec{x}-\vec{\mu}_i)\right\} \quad (3)$$

where $\vec{u}_i$ is the mean vector and $\sum i$ is the covariance matrix. Therefore a GMM can be represented by the collection of its parameters λ as

$$\lambda = \{p_i, \vec{\mu}_i, \Sigma_i\}, \quad i = 1,..., M \qquad (4)$$

B. Expectation Maximization (EM) Algorithm

The concern here is how to train an appropriate GMM which can provide a smooth estimate to the distribution of an observed random vector, given a sufficient amount of training data. The goal is to estimate the parameters λ of the GMM which best matches the distribution of the training vectors. To solve this, the maximum likelihood (ML) estimation is commonly used. This method tries to find which model parameters maximize the probability $p(\overline{X} \mid \lambda)$ of the GMM given the training vectors $\overline{X}$. This leads to a nonlinear function of the parameters λ. In [2], a special case of the EM algorithm was suggested to solve this problem. This algorithm tries to find the estimates of the ML parameters iteratively. It begins with an initial model λ, and tries to estimate a better model iteratively until some convergence is reached. According to the experimental results obtained in [1], this algorithm provides satisfying results when the training data is long enough and the model order is chosen correctly. However, the computational complexity of this algorithm is very high. This makes it not suitable for real-time operations such as speaker adaptation.

C. Matching Pursuit based estimation

Matching pursuit is a recently proposed algorithm for deriving signal-adaptive decompositions in terms of expansion functions chosen from an overcomplete dictionary – overcomplete in the sense that the dictionary elements, or atoms, exhibit a wide range of behaviors [4]. Roughly speaking, the matching pursuit algorithm is a greedy iterative algorithm which tries to determine an expansion, given a signal *s*[*n*] and a dictionary of atoms, $g_k[n]$, as follows

$$x[n] = \sum_{k=1}^{K} \alpha_k g_k[n] \qquad (5)$$

where the dictionary is a family of vectors (atoms) g_k included in a Hilbert space **H** with a unit norm $\| g_\gamma \| = 1$.

In the following, we introduce a fast method for estimating the parameters of a GMM using a modified version of the matching pursuit algorithm. We select the basis functions, $g_k[n]$, as Gaussians and fit the RHS of Equation (5) to the histogram of the data.

Given an arbitrary *D*-dimensional random vector $\vec{x}$, we want to obtain a Gaussian mixture density, which smoothly approximates the distribution of $\vec{x}$.

Let $\vec{x} = [x_1 \; x_2 \; \ldots \; x_D]^T$. Given a sequence of *T* training vectors $\overline{\boldsymbol{X}} = [\vec{x}_1, \vec{x}_2, ..., \vec{x}_T]$, we can write $\overline{\boldsymbol{X}}$ as

$$\overline{\boldsymbol{X}} = \begin{bmatrix} x_{1,1} & x_{1,2} & . & . & x_{1,T} \\ x_{2,1} & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ x_{D,1} & . & . & . & x_{D,T} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ . \\ . \\ \boldsymbol{X}_D \end{bmatrix} \qquad (6)$$

where $\boldsymbol{X_i}$, i=1,…, D are the sequences of training data corresponding to each of the *D* components of $\vec{x}$. If we can estimate an appropriate Gaussian mixture distribution for each $\boldsymbol{X_i}$, i=1,…,D , then we can obtain an overall distribution of $\vec{x}$ by multiplying the individual component densities. Thus, from each sequence $\boldsymbol{X_i}$ of the training data, we want to estimate a Gaussian mixture density corresponding to the component x_i , of the form

$$p(x_i \mid \lambda) = \sum_{m=1}^{M} p_m b_m(x_i) \qquad (7)$$

where $b_m(x_i)$, i=1,….,M are Gaussian functions.

We start by calculating the histogram of $\boldsymbol{X_i}$, we then normalize it by dividing it by T (the number of samples in $\boldsymbol{X_i}$) ,

$$H_i = \frac{\text{histogram}(\boldsymbol{X}_i)}{T} \tag{8}$$

where H_i is the discrete distribution of $\boldsymbol{X_i}$. Now, we want to estimate a Gaussian mixture distribution for x_i from H_i. If we can decompose H_i into a finite weighted sum of different Gaussian components, then the problem is solved. For this we use the matching pursuit algorithm to decompose H_i.

Matching pursuit algorithms are largely applied using dictionaries of Gabor atoms [5]. Gabor atoms are appropriate expansion functions for time-frequency signal decomposition, which are a scaled, modulated, and translated version of a single unit-norm window function, $g(\,.\,)$,

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g\left(\frac{t-\mu}{s}\right) e^{j\varepsilon t} \tag{9}$$

where $\gamma = (s, \mu, \varepsilon) \in \Gamma = \mathrm{R}^+ \mathrm{x}\, \mathrm{R}^2$. Note that $g_\gamma(t)$ is centered in a neighborhood of μ whose size is proportional to s and its Fourier transform is centered at $\omega = \varepsilon$. This parametric model provides modification capabilities for time and frequency localization properties of signals. In our application the modulation factor $e^{j\varepsilon t}$ is not necessary since the frequency localization has no meaning in this case, and thus it is dropped and we use

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g\left(\frac{t-\mu}{s}\right) \tag{10}$$

Furthermore, if we choose $g(t)$ as a Gaussian function of unit norm

$$g(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \tag{11}$$

then the resulting $g_\gamma(t)$ are valid Gaussian functions that can be used to decompose H_i.

To obtain the discrete form of (10), we sample the Gabor atoms to obtain

$$g_\gamma[n] = K_\gamma \frac{1}{\sqrt{s}} g\left(\frac{nN-\mu}{s}\right) = K'_\gamma g\left(\frac{nN-\mu}{s}\right) \tag{12}$$

where $\gamma = (s, \mu) \in \Gamma = \mathrm{R}^+ \mathrm{x}\, \mathrm{R}$, N is the sampling period, and K_γ is a constant that is adjusted so that $\| g_\gamma \| = 1$.

The method proceeds as follows. We first define our dictionary $\mathcal{D}$ as a family of vectors g_γ, $\gamma \in \Gamma$. The form of g_γ is shown in (12). The dictionary should be large enough to cover a wide range of vectors. The algorithm starts by finding $g_{\gamma_{i,0}} \in \mathcal{D}$ that best matches H_i in the sense that $|< H_i, g_{\gamma_{i,0}} >|$, which is a measure of similarity between H_i and $g_{\gamma_{i,0}}$, is maximized, i.e.,

$$|< H_i, g_{\gamma_{i,0}} >| \geq \sup |< H_i, g_\gamma >| \tag{13}$$

Then, we can write

$$H_i = < H_i, g_{\gamma_{i,0}} > g_{\gamma_{i,0}} + RH_i \tag{14}$$

where RH_i is the residual vector. The iteration then proceeds on RH_i as the initial vector. Suppose that R^nH_i denotes the n^{th} residual of H_i , at the n^{th} iteration we get

$$R^n H_i = < R^n H_i, g_{\gamma_{i,n}} > g_{\gamma_{i,n}} + R^{n+1} H_i \tag{15}$$

If we carry the iteration to order M, we obtain

$$H_i = \sum_{n=0}^{M-1} < R^n H_i, g_{\gamma_{i,n}} > g_{\gamma_{i,n}} + R^M H_i \tag{16}$$

Neglecting last residual R^MH_i, we obtain

$$H_i \approx \sum_{n=0}^{M-1} \alpha_{i,n} g_{\gamma_{i,n}} \tag{17}$$

$$\alpha_{i,n} = < R^n H_i, g_{\gamma_{i,n}} > \tag{18}$$

which gives a decomposition of H_i as a weighted sum of Gaussian components. We further normalize the weights of the individual components as

$$p_{i,n} = \frac{\alpha_{i,n}}{\sum_{m=1}^{M-1} \alpha_{i,m}}, \quad n = 0, ..., M-1 \tag{19}$$

so that $\sum_{n=1}^{M-1} p_{i,n} = 1$. In this way, we obtain a valid GMM for x_i :

$$p(x_i \mid \lambda_i) = \sum_{n=0}^{M-1} p_{i,n} g_{\gamma_{i,n}} \tag{20}$$

where $\lambda_i = \{p_{i,n}, \gamma_{i,n}\}$, $n = 0,..., M-1$.

The described procedure should be carried out for each single element x_i, i=1,…,D, of the random vector $\vec{x}$. The D resulting GMM's are multiplied to obtain the overall estimated distribution of $\vec{x}$.

$$p(\vec{x} \mid \lambda) = \prod_{i=1}^{D} p(x_i \mid \lambda_i) \tag{21}$$

where $\lambda = \{p_{i,n}, \gamma_{i,n}\}$, n = 0,…,M-1, i= 1,…,D.

D. Fast Calculations

The matching pursuit can be implemented using a fast algorithm described in [8], that computes $< R^{n+1} H_i, g_\gamma >$ from $< R^n H_i, g_\gamma >$ with a simple updating formula.

Consider (15), we can write it as

$$R^{n+1} H_i = R^n H_i - \left\langle R^n H_i, g_{\gamma_{i,n}} \right\rangle g_{\gamma_{i,n}} \tag{22}$$

Take the inner product with g_γ on each side, we obtain

$$\left\langle R^{n+1}H_i, g_\gamma \right\rangle = \left\langle R^n H_i, g_\gamma \right\rangle - \left\langle R^n H_i, g_{\gamma_{i,n}} \right\rangle \left\langle g_{\gamma_{i,n}}, g_\gamma \right\rangle \quad (23)$$

which is a simple updating formula for $< R^{n+1}H_i, g_\gamma >$. If we can calculate the inner product of all the atoms in the dictionary $< g_\alpha, g_\beta >$ and store it in a lookup table, then we can use this update formula to calculate $< R^{n+1}H_i, g_\gamma >$ at each iteration. The final algorithm is summarized in the following:

For each H_i , i=1,…,D

1. set n = 0 and compute $\{< H_i, g_\gamma >\}_{\gamma\in\Gamma}$
2. Find $g_{\gamma_{i,n}} \in \mathcal{D}$ such that
$$\left|\left\langle R^n H_i, g_{\gamma_{i,n}} \right\rangle\right| \geq \sup_{\gamma\in\Gamma} \left|\left\langle R^n H_i, g_\gamma \right\rangle\right|$$
3. Update for all $g_\gamma \in \mathcal{D}$
$$\left\langle R^{n+1}H_i, g_\gamma \right\rangle = \left\langle R^n H_i, g_\gamma \right\rangle - \left\langle R^n H_i, g_{\gamma_{i,n}} \right\rangle \left\langle g_{\gamma_{i,n}}, g_\gamma \right\rangle$$
4. If $n < M$-1 increment n and go to 2.

III. SPEAKER IDENTIFICATION

The main task of speaker identification is to extract and model the speaker-dependent characteristics of the speech signal, which can effectively distinguish one talker from another.

A. Speech analysis

The speech signal is first analyzed, and the silence periods are removed. Then the signal is divided into overlapping frames of approximately 20 ms length and a spacing of 10 ms. For each frame, cepstral coefficients derived from a mel-frequency filterbank (MFCC) are extracted [3].

B. Speaker identification

In speaker identification we a have a group of S speaker. For each speaker a training utterance is recorded. Form this utterance, a sequence of feature vectors is extracted and a GMM is estimated for that speaker. Each speaker will be represented by his own GMM λ_i , i = 1, … , S.

Whenever an test sequence $\overline{\boldsymbol{X}}$ is given, the model which maximizes the probability of observing this sequence is chosen and the corresponding speaker is identified as follows,

$$S_{identified} = \arg\max_{1\leq i\leq S} p(\overline{\boldsymbol{X}} \mid \lambda_i) \quad (24)$$

In the experiments which follow, the matching pursuit method introduced in the previous section, is used to estimate a Gaussian mixture density for the feature vectors extracted from the speech signal. Figure 1, shows Gaussian mixture distributions estimated for some MFCC parameters using the matching pursuit method and the EM method. The estimated distributions shown in Figure 1, are scaled so as to fit the histogram of the MFCC parameters. Note that the estimation is good and smooth in both methods. The matching pursuit based model performs as good as the EM based model.

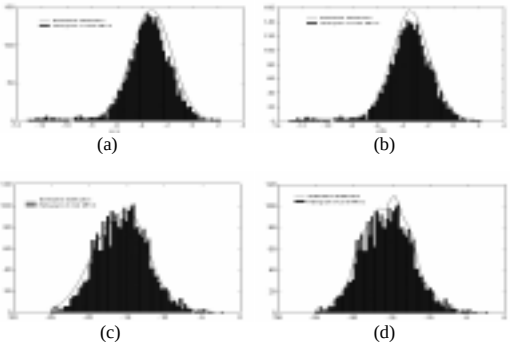

Fig. 1. The histograms of single cepstral coefficients and their corresponding distribution estimation: (a) MFCC2 using MP, (b) MFCC2 using EM, (c) MFCC12 using MP, (d) MFCC12 using EM.

C. Implementation

In this part, some algorithmic issues will be discussed. The first issue is the number the mel-cepstral coefficients used in parameterizing the speech signal. Usually, for each frame 12-MFFC coefficients are derived. Nevertheless, in some applications [6], it was shown useful to derive 24 coefficients: 12 MFCC and 12 Δ-MFCC, which are concatenated to form the feature vector.

Note also that when using the matching pursuit method developed in the previous section, a GMM model can be derived for each single feature from the feature vector separately. In speaker identification this enables us to obtain the probability of observing each element in the feature vector sequence separately. So, two methods are possible for making a decision about the identity of the speaker, given the feature vector sequence $\overline{\boldsymbol{X}} = [\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_D]^T$:

Method 1: This is the traditional method where the speaker is chosen upon the probability derived from the overall GMM model

$$S_{identified} = \arg\max_{1\leq i\leq S} p(\overline{\boldsymbol{X}} \mid \lambda_i)$$

Method 2: Obtain the probability of each feature separately for all speaker models. Then check how many features get the largest probability in each model, the model which gets the maximum probability in more features is chosen.

Calculate $p_{n,i} = p(\boldsymbol{X}_n \mid \lambda_i)$, for every feature ($n$= 1,..,$D$) and every model (i = 1,…,S).

The model λ_i which yields maximum $p_{n,i}$ for more features is chosen and the corresponding speaker is identified.
Our experiments are done using both of these methods and results are compared.

D. Database Description

The experiments were carried out on the POLYCOST 250 database (v1.0). The POLYCOST database is dedicated to speaker recognition applications [7]. The main purpose behind it is to provide a common database on which speaker recognition algorithms can be compared and validated. The database was recorded from 134 subjects coming from 14 European countries. Around 10 sessions were recorded for each subject, each session contains 14 items. The recordings were made over the telephone network with an 8 kHz sampling frequency. In [7], a set of baseline experiments is defined for which results should be included when presenting evaluations made on this database. Our experiments follow the set of rules defined in [7] under "text-independent speaker identification" .

IV. EXPERIMENTAL RESULTS AND CONCLUSIONS

Models are trained using nearly 20 sec of 8 kHz speech samples. Tests are done using utterances of about 5 sec. 10 speakers are used in these experiments. Recordings from two sessions are used for training, and about six sessions are used for testing.
Speaker modeling was carried out using the matching pursuit method and the EM method. Table 1 shows the identification rate obtained for each method. The model order in both methods is M=16. Both experiments are done under exactly the same conditions and using the same speech data. Here we used the overall probability of the feature vectors for the MP method. Table 1 also includes the necessary training time (using MATLAB in a Pentium-based PC) for each method, corresponding to a training sequence of 40 sec.

TABLE I
Identification performance of MP and EM based models. The training data corresponds to a training data of 40 sec. M=16.

	MP-based model	EM-based model
Ident. Rate	70%	71%
Training time	3 min	14 min

The performance obtained using the matching pursuit method is as good as the performance of the EM based method. However, the training time required by the MP method is remarkably shorter than that required by the EM algorithm, this shows the low computational complexity of the MP method. During training, the EM algorithm needs many iterations to reach convergence, while the matching pursuit algorithm has a fixed number of iteration which is equal the model order M, which is usually low (in the order of 16). Moreover the calculations involved in the EM algorithm are computationally costly. The MP algorithm saves a lot of calculations using the updating formula (23) and the inner product lookup table for the atoms.
Table 2 shows the identification rates obtained by the MP algorithm for different model order M using both methods described in part *C* for deciding on the speaker identity.

TABLE II
Identification performance of the MP method using speaker identity decision based on a) D-variate GMM probability and, b) probability of separate feature models.

Model order	a.D-variate model	b. Separate feat. models
M = 4	63%	58%
M = 8	69%	65%
M = 12	71%	73%
M = 16	70%	68%

It is clear that the model order M is important for the precision of the model. Nevertheless, beyond some point (around M=12) increasing the model order is useless and can decrease the performance. Note also that using separate feature models can also be as good as (or even better than) using an Overall GMM, when the model order is appropriate.

REFRENCES

[1] D. A. Reynolds and R. C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models," *IEEE trans. Speech Audio Processing*, vol. 3, no. 1, pp. 72-83, Jan. 1995.

[2] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc.*, vol. 39, pp. 1-38, 1997.

[3] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 357-366, Aug. 1980.

[4] S. Mallat and Z. Zhang, "Matching Pursuits with time-frequency Dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397-3415, Dec. 1993.

[5] R. Gribonval, Ph. Depalle, X. Rodet, E. Barcy, and S. Mallat, "Sound Signals Decomposition Using a High Resolution Matching Pursuit," *ICMC'96*.

[6] Hakan Altincay, "Experimental Work on Classifier Combination for Speaker Identification," to appear in *EUROSPEECH'99 Conference*, Budapest, Sep. 1999.

[7] Hakan Melin and Johan Lindberg, "Guidelines for Experiments on the POLYCOST Database," Version 1.0, January 8th, 1997.

[8] Stephane Mallat, *Wavelet Tour of signal Processing*, Academic Press, 1998.