Fuzzy Gaussian Mixture Models for Speaker Recognition

Dat Tran and Michael Wagner
Human-Computer Communication Laboratory
School of Computing, University of Canberra, ACT 2601, Australia
E-mail: (dat, miw)@hccl.canberra.edu.au

Abstract

The Gaussian mixture model (GMM) is an important application of statistical clustering to speaker recognition. A number of prototypes are generated from the training feature vectors by representing the feature space as a mixture of Gaussian distributions. Each prototype consists of a model parameter set including mean vector, covariance matrix and mixture weight. In fuzzy clustering, the fuzzy c-means (FCM) method is the most widely used. Model parameters in each prototype include fuzzy mean vector and fuzzy covariance matrix. Both the GMM and the FCM methods have similar characteristics: using iterative optimisation algorithms, feature vectors can belong to more than one class, and degrees of belonging of a vector across classes sum to one. From these similarities, a FCM-based generalisation to the GMM called the fuzzy GMM (FGMM) is proposed in this paper. Fuzzy mixture weights are introduced by redefining the distances in the FCM functionals. The FGMM algorithm and its use in speaker recognition are considered. The experimental results show that with a suitable degree of fuzziness, the FGMMs are more effective than the GMMs in tests on 16 speakers using the T146 database and on 108 speakers using the ANDOSL database.

1. Introduction

Speaker recognition is the process of automatically recognising a speaker by using speaker-specific information included in speech waves [17]. This technique can be used to verify the identity claimed by people accessing certain protected systems; that is, it enables access control of various services by voice [11]. Voice dialling, banking over a telephone network, database access services, security control for confidential information, and remote access of computers are important applications of speaker recognition technology.

Speaker recognition can be classified into two specific tasks: identification and verification. Speaker identification is the process of determining which one of the voices known to the system best matches the input voice sample. When an unknown speaker must be identified as one of the set of known speakers, the task is known as closed-set speaker identification. If the input voice sample does not have a close enough match to anyone of the known speakers and the system can produce a "no match" decision [24], the task is known as open-set speaker identification. Speaker verification is the process of accepting or rejecting the identity claim of a speaker. An identity claim is made by an unknown speaker, and an utterance of this unknown speaker is compared with the model for the speaker whose identity is claimed. If the match is good enough, that is, above a given threshold, the identity claim is accepted. Most of the applications in which voice is used to confirm the identity claim of a speaker are classified as speaker verification.

Speaker recognition methods can also be divided into text-dependent and text-independent. When the same text is used for both training and testing, the system is said to be text-dependent. For text-independent operation, the text used to train and test the system is completely unconstrained.

Methods used for text-dependent speaker recognition are usually based on the dynamic time warping (DTW) and the hidden Markov model (HMM) methods. For text-independent speaker recognition, methods are usually based on long-term statistics, vector quantisation (VQ) and ergodic HMM methods [11]. In the above classifications, the GMM is regarded as a single-state continuous ergodic HMM. It has been reported [20] that the performance of text-independent speaker identification depends on the total number of mixture components (number of states times number of mixture components assigned to each state) per speaker model. Therefore, it can be seen that, the N-state M-mixture continuous ergodic HMM is roughly equivalent to the NM-mixture GMM in speaker recognition applications. Moreover, since GMMs are effective models capable of achieving high identification accuracy, they are currently used for speaker recognition.

In general, the GMM is a statistical clustering method. Its algorithm can be referred to as a prototype-based algorithm, that is, a number of prototypes are generated from the training feature vectors by representing the feature space as a mixture of Gaussian distributions. Each prototype consists of a set of model parameters including mean vector, covariance matrix and mixture weight. Parameters are trained in an unsupervised classification using the expectation maximisation (EM) algorithm [8]. This algorithm provides an iterative maximum likelihood estimation technique. Experiments have shown that as long as the training samples cover a sufficient variety of the speaker's speech sound, GMMs are effective models.
The FCM method is the most widely used approach in both theory and practical applications of fuzzy clustering techniques to unsupervised classification. It is an extension of the hard $c$-means algorithm [9] and was first introduced by Dunn [10]. From Dunn's classical within-groups sum of squared error function, the infinite family of FCM functionals were generalised by Bezdek [3]. The significant contributions by Bezdek are to introduce a weighting exponent $m$ on each fuzzy membership called degree of fuzziness, and a distance in $A$ norm ($A$ is any positive definite matrix). Hence a general estimation procedure for the FCM has been established and its convergence has been shown [2]. A family of the FCM algorithms with various degrees of fuzziness is used to minimise the FCM functionals, where fuzzy mean vectors are iteratively updated. Therefore, the FCM algorithms can also be referred to as the prototype-based algorithms. Gustafson and Kessel [14] have proposed a modification of the FCM algorithms, which attempts to recognise the fact that different clusters in the same data set may have differing geometric shapes. These algorithms are referred to as fuzzy covariance clustering algorithms and each prototype consists of a set of model parameters including mean vector and covariance matrix. Gath and Geva [13] have defined an exponential distance for the FCM algorithm to obtain a fuzzy approach to maximum likelihood estimation.

Both the GMM and the FCM methods have some similar characteristics. First, their algorithms are iterative optimisation algorithms. Second, every method provides a soft partitioning of a speaker's feature space, that is, every feature vector can belong to more than one class. In the GMM, the degree of belonging of vector $x$ to the class $i$ is represented by the probability density function $p(x \mid i)$, whereas in the FCM, it is represented by the fuzzy membership function $u_i(x)$. Degrees of belonging of a vector across all classes sum to one in both algorithms. Third, the least-squares functionals in the FCM and the auxiliary $Q$-functions in the GMM are similar. Based on these similarities, a FCM-based modification of the GMM is proposed in this paper. This modification begins with redefining the distances in the FCM functionals as the negative logarithms of the above-mentioned probability density functions. In the training process, to minimise these modified FCM functionals, fuzzy mixture weights are defined and are computed together with the fuzzy mean vectors and fuzzy covariance matrices in the reestimation formulas. In the testing process, alternative discriminant functions for classification are proposed instead of likelihood functions. The GMMs in this modification could be named fuzzy Gaussian mixture models (FGMMs).

The rest of the paper is organised as follows. The GMM method and its use for speaker identification and verification are summarised in the next section together with the likelihood normalisation method proposed by Matsui and Furui [19][12] for speaker verification. The FCM method is summarised in Section 3. Section 4 presents the FGMM and its use for speaker identification and verification. The experimental results in Section 5 show that the FGMMs are more effective than the GMMs in tests on 16 speakers using the T146 database and on 108 speakers using the ANDOSL database.

2. Gaussian Mixture Models

This section presents the GMM method for speaker recognition. Parameter estimation equations for training speaker models are presented first. The GMM method for speaker identification is then described as a maximum likelihood classifier. Finally, speaker verification and normalisation techniques are described.

2.1 The GMM Algorithm

Let $X = \{x_1, x_2, ..., x_T\}$ be a set of $T$ vectors, each of which is a $d$-dimensional feature vector extracted by digital speech signal processing. Since the distribution of these vectors is unknown, it is approximately modelled by a mixture of Gaussian densities, which is a weighted sum of $c$ component densities, given by the equation

$$p(x_t \mid \lambda) = \sum_{i=1}^{c} w_i N(x_t, \mu_i, \Sigma_i) \tag{1}$$

where $\lambda$ denotes a prototype consisting of a set of model parameters $\lambda = \{w_i, \mu_i, \Sigma_i\}$, $w_i$, $i = 1, ..., c$, are the mixture weights and $N(x_t, \mu_i, \Sigma_i)$, $i = 1, ..., c$, are the $d$-variate Gaussian component densities with mean vectors $\mu_i$ and covariance matrices $\Sigma_i$.

$$N(x_t, \mu_i, \Sigma_i) = \frac{\exp\left(-\frac{1}{2}(x_t - \mu_i)^T \Sigma_i^{-1}(x_t - \mu_i)\right)}{\left(2\pi\right)^{d/2} |\Sigma_i|^{1/2}} \tag{2}$$

In training the GMM, these parameters are estimated such that in some sense, they best match the distribution of the training vectors. The most widely used training method is the maximum likelihood (ML) estimation. For a sequence of training vectors $X$, the likelihood of the GMM is

$$p(X \mid \lambda) = \prod_{i=1}^{T} p(x_t \mid \lambda) \tag{3}$$

The aim of ML estimation is to find a new parameter model $\bar{\lambda}$ such that $p(X \mid \bar{\lambda}) \geq p(X \mid \lambda)$. Since the expression in (3) is a nonlinear function of parameters in $\lambda$, its direct maximisation is not possible. However, parameters can be obtained iteratively using the expectation-maximisation (EM) algorithm [8]. An auxiliary function $Q$ is used

$$Q(\lambda, \tilde{\lambda}) = \sum_{i=1}^{T} p(i \mid x_t, \lambda) \log[\bar{w}_i N(x_t, \mu_i, \Sigma_i)] \tag{4}$$

where $p(i \mid x_t, \lambda)$ is the a posteriori probability for acoustic class $i$, $i = 1, ..., c$ and satisfies

Summer 1998 Australian Journal of Intelligent Information Processing Systems
p(\mid X, \lambda) = \frac{\sum_{i=1}^{T} p(i \mid x_t, \lambda) x_t}{\sum_{t=1}^{T} p(i \mid x_t, \lambda)} \quad \ldots \quad (6)

The basis of the EM algorithm is that if \( Q(\mid \lambda, \lambda) \geq Q(\mid \lambda, \lambda) \) then \( p(X \mid \lambda) \geq p(X \mid \lambda) \) \[16\]. Setting derivatives of the \( Q \) function with respect to \( \lambda \) to zero, the following reestimation formulas are found:

\[
\frac{v_k}{i=1} = \frac{1}{\sum_{t=1}^{T} \sum_{i=1}^{T} p(i \mid x_t, \lambda) x_t} \quad \ldots \quad (7)
\]

\[
\frac{\sum_{t=1}^{T} p(i \mid x_t, \lambda) (x_t - \mu_t)}{\sum_{t=1}^{T} p(i \mid x_t, \lambda)} \quad \ldots \quad (8)
\]

The algorithm for training the GMM is described as follows:

**Algorithm 1 (the GMM algorithm):**

**Step 1:** Generate the a posteriori probability \( p(i \mid x_t, \lambda) \) at random satisfying (5)

**Step 2:** Compute the mixture weight, the mean vector, and the covariance matrix following (6), (7) and (8)

**Step 3:** Update the a posteriori probability \( p(i \mid x_t, \lambda) \) according to (5) and compute the Q function using (4)

**Step 4:** Stop if the increase in the value of the Q function at the current iteration relative to the value of the Q function at the previous iteration is below a chosen threshold, otherwise go to step 2.

### 2.2 The GMM in Speaker Identification

Let \( \lambda_k \), \( k = 1, ..., N \), denote speaker models of \( N \) speakers. Given a feature vector sequence \( X \), a classifier is designed to classify \( X \) into \( N \) speaker models by using \( N \) discriminant functions \( g_k(X) \), computing the similarities between the unknown \( X \) and each speaker model \( \lambda_k \) and selecting the model \( \lambda_k^* \) if

\[
k^* = \arg \max_{1 \leq k \leq N} g_k(X) \quad \ldots \quad (9)
\]

In the minimum-error-rate classifier, the discriminant function is the a posteriori probability

\[
g_k(X) = p(\mid X, \lambda_k) \quad \ldots \quad (10)
\]

Using the Bayes rule and assuming equally likely speakers, i.e., \( p(\lambda_k) = 1/N \), and noting that \( p(X) \) is the same for all speaker models, the discriminant function in (10) is equivalent to the following \[22\]-\[24\]

\[
g_k(X) = p(X \mid \lambda_k) \quad \ldots \quad (11)
\]

Finally, using the log-likelihood, the decision rule used for speaker identification is

\[
Select \text{ speaker } k^* \quad k^* = \arg \max_{i=1}^{T} \sum_{t=1}^{T} \log p(x_t \mid \lambda_k) \quad \ldots \quad (13)
\]

where \( p(x_t \mid \lambda_k) \) is given in (1).

### 2.3 The GMM in Speaker Verification

Let \( \eta \) be a given threshold, associated with the claimed speaker model \( \lambda_c \). A discriminant function \( g(X) \) for the unknown \( X \) and model \( \lambda_c \) used to reject or accept the claimed speaker is as follows:

\[
g(X) \begin{cases} \geq \eta & \text{accept} \\ < \eta & \text{reject} \end{cases} \quad \ldots \quad (14)
\]

The threshold \( \eta \) is taken as an equal-error-rate (EER) threshold. In many algorithms for speaker verification, \( g(X) \) is often the likelihood ratio between the claimed speaker model and the background speaker model set \( \lambda_B \)

\[
g(X) = \log p(X \mid \lambda_c) - \log p(X \mid \lambda_B) \quad \ldots \quad (15)
\]

where

\[
\log p(X \mid \lambda_c) = \frac{1}{T} \sum_{t=1}^{T} \log p(x_t \mid \lambda_c) \quad \ldots \quad (16)
\]

and the second term on the right-hand side of (15) is called the normalisation term. Various methods have been proposed to compute this term, such as the method based on the a posteriori probability proposed by Matsui and Furui [12][19], where

\[
\log p(X \mid \lambda_B) = \sum_{i=1}^{B} \sum_{t=1}^{T} \log p(x_t \mid \lambda_i) \quad \ldots \quad (17)
\]

with a set of \( B \) "cohort speakers" \( \lambda_i \) [11][20] that are representative of the population "near" the claimed speaker. It should be noted that the claimed speaker is included in the set \( B \) in this method.

### 3. Fuzzy C-Means

The FCM method for fuzzy clustering is presented in this section. The fuzzy membership function, parameter estimation equations for training fuzzy models and the use of the FCM for cluster analysis are described.
3.1 Fuzzy Membership Function

Fuzzy sets [31] are a generalisation of conventional set theory that was introduced as a new way to represent the vagueness or imprecision that is ever present in our daily experience as well as in natural language [1]. For example, an advice for a driving student approaching a red light: "Apply the brakes pretty soon" contains a type of uncertainty called fuzziness.

Consider the set \( H \) of real numbers from 5 to 9. This set is crisp, namely, every number \( x \) either is in \( H \) or is not. \( H \) can be described by its membership function \( m_H \), defined as

\[
m_H = \begin{cases} 1 & 5 \leq x \leq 9 \\ 0 & \text{otherwise} \end{cases}
\]

Next, consider the set \( F \) of real numbers that are close to 7. The property "close to 7" is fuzzy, so there is not a unique membership function for \( F \). Properties for this fuzzy set may include: (1) normality \( m_F(T) = 1 \), (2) monotonicity [the closer \( x \) is to 7, the closer \( m_F(x) \) is to 1, and conversely]; and (3) symmetry [numbers equally far left and right of 7 should have equal membership].

Fuzzy sets are always (and only) functions, from some "universe of objects," say \( X \), into \([0, 1]\), the range of \( m_F \). Therefore the fuzzy membership function is the basic idea in fuzzy set theory, its values measure degrees to which objects satisfy imprecisely defined properties [1]-[3].

3.2 The FCM Algorithm

Let \( X = \{x_1, x_2, \ldots, x_T\} \) be a set of \( T \) vectors, the structure of which is analysed by means of a cluster analysis technique. Fuzzy clustering known as unsupervised learning in \( X \) is a fuzzy partitioning of \( X \) into \( c \) fuzzy subsets or \( c \) clusters, \( 1 < c < T \). The most important requirement is to find a suitable measure of clusters, referred to as a fuzzy clustering criterion. Objective function methods allow the most precise formulation of the fuzzy clustering criterion. The most well known objective function for fuzzy clustering in \( X \) is the least-squares functional, that is, the infinite family of fuzzy c-means (FCM) functionals, generalised from the classical within-groups sum of squared error function by Bezdek [3]

\[
J_m(U, \mu; X) = \sum_{i=1}^{c} \sum_{t=1}^{T} u_{it}^m d_{it}^2 \tag{19}
\]

where \( U = \{u_{it}\} \) is a fuzzy \( c \)-partition of \( X \), each \( u_{it} \) represents the degree of vector \( x_t \) belonging to the \( i \)th cluster and is called the fuzzy membership function. For \( 1 \leq i \leq c \) and \( 1 \leq t \leq T \), we have

\[
0 \leq u_{it} \leq 1, \quad \sum_{i=1}^{c} u_{it} = 1, \quad \text{and} \quad 0 < \sum_{t=1}^{T} u_{it} < T \tag{20}
\]

\( m \geq 1 \) is a weighting exponent on each fuzzy membership \( u_{it} \) and is called the degree of fuzziness; \( \mu = (\mu_1, \ldots, \mu_c) \) are cluster centers and, \( d_{it} \) is the distance in the \( A \) norm (\( A \) is any positive definite matrix) from \( x_t \) to \( \mu_i \), known as a measure of dissimilarity

\[
d_{it}^2 = \|x_t - \mu_i\|^2_A = (x_t - \mu_i)' A (x_t - \mu_i) \tag{21}
\]

The basic idea in the FCM is to minimise \( J_m \) over the variables \( U \) and \( \mu \), on the assumption that matrices \( U \) that are part of optimal pairs for \( J_m \) identify good partitions of the data. Minimising the fuzzy objective function \( J_m \) in (19) gives

\[
u_{it} = \bigg[ \sum_{k=1}^{c} (d_{it} / d_{ik})^2 \bigg]^{-1} \tag{22}
\]

\[
u_{it} = \frac{\sum_{t=1}^{T} u_{it}^m x_t}{\sum_{t=1}^{T} u_{it}^m} \tag{23}
\]

The FCM algorithm is known as the fuzzy vector quantisation (FVQ) algorithm in speech and speaker recognition and is used to train codebooks in the VQ approach. This algorithm is described as follows

Algorithm 2 (the FCM algorithm):

Step 1: Choose any inner product norm metric for \( \mathbb{R}^d \), fix \( c \) and \( m, 2 < c < T, m > 1 \). Generate matrix \( U \) at random satisfying (20)

Step 2: For \( i = 1, \ldots, c \), compute the \( c \) fuzzy mean vectors \( \{\mu_i\} \) with (23) and the distances \( d_{it} \) with (21). If \( d_{it} = 0 \) for some \( t \), set \( u_{it} = 1, u_{is} = 0, \forall s \neq t \)

Step 3: Update matrix \( U \) using (22)

Step 4: Stop if the decrease in the value of the fuzzy objective function \( J_m \) at the current iteration relative to the value of the \( J_m \) at the previous iteration is below a chosen threshold, otherwise go to step 2.

An interesting modification of the FCM has been proposed by Gustafson and Kessel [14]. It attempts to recognise the fact that different clusters in the same data set \( X \) may have differing geometric shapes. A generalisation to a metric that appears more natural was made through the use of a fuzzy covariance matrix. Replacing (21) by an inner product induced a norm of the form

\[
d_{it}^2 = (x_t - \mu_i)' M_i (x_t - \mu_i) \tag{24}
\]

where the \( M_i \) are symmetric and positive definite and subject to the following constraints

\[
|M_i| = \rho_i \tag{25}
\]

with \( \rho_i > 0 \) and fixed for each \( i \). Define a fuzzy covariance matrix \( \Sigma_i \) by

\[
\Sigma_i = \sum_{t=1}^{T} u_{it}^m (x_t - \mu_i)(x_t - \mu_i)' / \sum_{t=1}^{T} u_{it}^m \tag{26}
\]

then we have
where \(|M_i|\) and \(|\Sigma_i|\) are the determinants of \(M_i\) and \(\Sigma_i\), respectively and \(d\) is the feature space dimension.

Step 2 in algorithm 2 is now generalised as follows. Compute the \(c\) fuzzy mean vectors \(\{\mu_i\}\) using (23), the \(c\) fuzzy covariance matrices \(\{\Sigma_i\}\) using (26) and the distances \(d_{it}\) using (24). If \(d_{it} = 0\) for some \(t\), set \(u_{it} = 1, u_{is} = 0, \forall s \neq t\).

4. Fuzzy Gaussian Mixture Models

A modification of the FCM algorithm called the fuzzy GMM (FGMM) is proposed in this paper. Our goal is to apply a FCM estimate to a Bayesian classifier in the particular case of a mixture of \(c\) Gaussian distributions. It attempts to recognise the fact that different clusters in the same data set \(X\), beyond differing geometric shapes, may have differing data densities, denoted by mixture weights (class a priori probabilities).

4.1 The FGMM Algorithm

The FCM metric is generalised through the use of a fuzzy mean vector, a fuzzy covariance matrix and a fuzzy mixture weight. To obtain these, since the density of the data in cluster \(i\) is proportional to the joint mixture density function \(f(x_i, \mid \lambda)\), we can define the dissimilarity denoted by the distance in (21) as

\[
d^2_{it} = -\log p(x_i, \mid \lambda) = -\log[\bar{w}_i N(x_i, \mu_i, \Sigma_i)]
\]  

Using the expression in (2), we have

\[
d^2_{it} = -\log \bar{w}_i + \frac{1}{2} \log(2\pi)^d |\Sigma_i|
\]  

An approximation of this distance was used in the entropy constrained VQ algorithm or generalised k-means VQ algorithm to train codebooks from the set \(X\) [3] in the VQ approach. If clusters have the same densities or are subject to the constraints \(w_i = \alpha_i\), where \(\alpha_i > 0\) and fixed for each \(i\), the term \(-\log \bar{w}_i\) in (29) will be omitted in the minimisation procedure. The distance in (29) is now equivalent to the one in (24) proposed by Gustafson and Kessel [14], subject to the constraints in (25).

The argument list of \(J_m\) is extended using \(\Sigma = \{\Sigma_1, \ldots, \Sigma_c\}\) and \(w = \{w_1, \ldots, w_c\}\) and we have

\[
J_m(U, \mu, \Sigma, w, X) = \sum_{i=1}^{c} \sum_{t=1}^{T} u_{it}^m d^2_{it}
\]  

Substituting (28) into (30) gives

\[
J_m(U, \mu, \Sigma, w, X) = -\sum_{i=1}^{c} \sum_{t=1}^{T} u_{it}^m \log \bar{w}_i
\]  

Minimising \(J_m\) is performed by minimising each term on the right hand side of (31). To minimise the first term, note that

\[
\sum_{i=1}^{c} \bar{w}_i = 1
\]  

and using the Lagrange multiplier \(\kappa\) [16], the following augmented objective function is maximised

\[
f(w) = \sum_{i=1}^{c} \sum_{t=1}^{T} u_{it}^m \log \bar{w}_i + \kappa \left( \sum_{i=1}^{c} \bar{w}_i - 1 \right)
\]  

we have

\[
\bar{w}_i = \sum_{t=1}^{T} u_{it}^m / \sum_{t=1}^{T} \sum_{i=1}^{c} u_{it}^m
\]  

The expression of \(\bar{w}_i\) in (34) is defined as the fuzzy mixture weight. Minimising the second term on the right-hand side of (31) is obtained by setting its derivatives with respect to \(\mu_i\) and \(\Sigma_i\) to zero for every \(i = 1, \ldots, c\)

\[
\sum_{t=1}^{T} u_{it}^m \Sigma_i^{-1}(x_t - \mu_i) = 0
\]  

\[
\sum_{t=1}^{T} u_{it}^m \Sigma_i^{-1}(x_t - \mu_i)(x_t - \mu_i)' = 0
\]  

To obtain (35), the following identities are used

\[
\nabla_b(b' Ab) = Ab + A'b
\]  

\[
\nabla_A(b' Ab) = bb'
\]  

\[
\nabla_A |A| = A^{-1} |A|
\]  

where \(A\) and \(b\) are a \(d\)-by-\(d\) matrix and a \(d\)-dimensional column vector, respectively. From (35) and (36) we have

\[
\bar{\mu}_i = \frac{\sum_{t=1}^{T} u_{it}^m x_t}{\sum_{t=1}^{T} u_{it}^m}
\]  

\[
\bar{\Sigma}_i = \frac{\sum_{t=1}^{T} u_{it}^m (x_t - \mu_i)(x_t - \mu_i)'}{\sum_{t=1}^{T} u_{it}^m}
\]  

where \(u_{it}\) is computed using (22) since it is derived from minimising \(J_m\) with \(u_{it}\) as variables. The algorithm based on these estimation formulas could be named the FGMM algorithm and is stated as follows

Algorithm 3 (the FGMM algorithm):

1. Fix \(c\) and \(m\), \(2 < c < T, m > 1\). Generate matrix \(U\) at random satisfying (20)
Step 2: For \( i = 1, \ldots, c \), compute the \( c \) fuzzy mixture weights \( \{w_i\} \) using (34), the \( c \) fuzzy mean vectors \( \{\mu_i\} \) using (40), the \( c \) fuzzy covariance matrices \( \{\Sigma_i\} \) with (41) and the distances \( d_{it} \) in (29). If \( d_{it} = 0 \) for some \( t \), set \( u_{it} = 1 \), \( u_{it} = 0 \), \( \forall i \neq t \).

Step 3: Update matrix \( U \) using (22)

Step 4: Stop if the decrease in the value of the fuzzy objective function \( J_m \) at the current iteration relative to the value of the \( J_m \) at the previous iteration is below a chosen threshold, otherwise go to step 2.

### 4.2 The FGMM in Speaker Identification

In the GMM algorithm, given the training sequence \( X \), the model parameters \( \lambda \) are determined such that the function \( Q(\lambda, \lambda) \) in (4) is maximised and subsequently, the likelihood function \( p(X | \lambda) \) is also maximised.

Therefore, the use of the likelihood in (12) as the discriminant function for speaker identification is suitable to determining model parameters in the training process.

In the FGMM method, we can have two cases for selecting the discriminant function in speaker identification. In the first case, for values of degree of fuzziness \( m \) that are close to one, the fuzzy \( Q \)-function obtained from substituting (28) to (30)

\[
Q_m(U, \lambda) = -J_m(U, w, \mu, \Sigma; X) = \sum_{i=1}^{c} \sum_{t=1}^{T} u_{it}^m \log(w_i N(x_i, \mu_i, \Sigma_i)) \tag{42}
\]

approaches the \( Q \)-function of the GMM in (4) in the sense that \( u_{it} \) approaches \( p(i | x_i, \lambda) \). It has been shown by Hathaway [15] that the fuzzy \( Q \)-function at \( m = 1 \) is maximised if \( u_{it} = p(i | x_i, \lambda) \). Therefore, with \( m \) close to one, minimising \( J_m(U, w, \mu, \Sigma; X) \) is equivalent to maximising \( Q_m(U, \lambda) \) and leads to maximising \( Q(\lambda, \lambda) \) in the GMM, and \( p(X | \lambda) \) also. In this case, the decision rule in (13) for the GMM is applicable to the FGMM.

In the second case, as \( m > 1 \) and not close to one, maximising \( Q_m(U, \lambda) \) does not lead to maximising \( Q(\lambda, \lambda) \), the discriminant function cannot be the likelihood as in (12). It is reasonable to choose \( Q_m(U, \lambda) \) as an alternative discriminant function. To reduce computations, the \( U \) matrix in \( Q_m(U, \lambda) \) is replaced by the expression in (22) and (28)

\[
Q_m(U, \lambda) = \sum_{i=1}^{c} \sum_{t=1}^{T} \log(p(x_i, i | \lambda))^{-1} \tag{43}
\]

Let \( \lambda_k, k = 1, \ldots, N \), denote speaker models of \( N \) speakers and given an unknown sequence \( X \), a classifier is designed to classify \( X \) into \( N \) speaker models by using \( N \) following discriminant functions

\[
g_k(X) = Q_m(\lambda_k) \tag{44}
\]

Finally, the decision rule used for speaker identification is

\[
\text{Select speaker } k^* \text{ if } k^* = \arg \max_{1 \leq k \leq N} \sum_{i=1}^{c} \sum_{t=1}^{T} [-\log p(x_i, i | \lambda_k)]^{-1} \tag{45}
\]

It should be noted that, the decision rule in (44) is applicable to the first case as \( m \) close to one.

### 4.3 The FGMM in Speaker Verification

Similarly, for values of \( m \) close to one, the likelihood ratio in (15) is applicable to speaker verification. For large values of \( m \), the likelihood functions in (15) should be changed to the fuzzy \( Q \)-function in (43) as follows

\[
g(X) = Q_m(\lambda_a) - Q_m(\lambda_B) \tag{46}
\]

### 5. Experimental Results

According to the theoretical considerations above, we present in this paper the results of GMM-based and FGMM-based speaker recognition experiments. The TI46 and the ANDOSL speech data corpora are used to compare these algorithms.

#### 5.1 The TI46 database

This corpus of speech was designed and collected at Texas Instruments (TI). It contains 16 speakers, 8 female and 8 male, labelled f1-f8 and m1-m8, respectively. There are 46 words per speaker: ten digits from 0 to 9, 26 letters from a to z, and the ten command words enter, erase, go, help, no, rubout, repeat, stop, start, and yes. Each speaker repeated the words ten times in a single training session, and then again twice in each of 8 later testing sessions. The corpus was sampled at 12500 samples per second and 12 bits per sample. The data were processed in 20.48 ms frames (256 samples) at a frame rate of 125 frames per second (100 sample shift). Frames were Hamming windowed and preemphasised with \( \mu = 0.9 \). For each frame, 46 mel-spectral bands of a width of 110 mel and 20 mel-frequency cepstral coefficients (MFCC) were determined [29].

#### 5.2 The ANDOSL database

The Australian National Database of Spoken Language (ANDOSL) [21] comprises carefully balanced material for Australian speakers, both native-born and overseas-born migrants. Current holdings are divided into those from native speakers of Australian English (born and fully educated in Australia) and those from non-native speakers of Australian English (first generation migrants having a non-English native language). There are 108 native
speakers, divided into 36 speakers of General Australian English, 36 speakers of Broad Australian English, and 36 speakers of Cultivated Australian English comprising 6 speakers of each gender in each of three age ranges (18-30, 31-45 and 46+). Each speaker contributed in a single session, 200 phonetically rich sentences. All waveforms were sampled at 20 kHz and 16 bits per sample. For the processing telephone speech purpose in our laboratory, all waveforms were converted from 20 kHz into 8 kHz (phone bandwidth). Speech processing was performed using HTK V2.0 [30]. Low and high pass cut-offs were set to 300 Hz and 3400 Hz. The data were processed in 32 ms frames at a frame rate of 10 ms. Frames were Hamming windowed and preemphasised with $\mu = 0.97$. The basic feature set consisted of the 12th-order lifted mel-frequency cepstrum coefficients (MFCCs) and the normalised short-time energy, augmented by the corresponding delta MFCCs to form a final set of feature vector with a dimension of 26 for individual frames.

5.3 Speaker Identification Experiments

For the TI46 database, the vocabulary was the set of ten command words. In the training phase, 100 training tokens (10 utterances x 1 training session x 10 repetitions) of each speaker were used to train GMMs and FGMMs of 32, 64, and 128 mixtures for 16 speakers. Speaker identification was carried out in text-independent mode by testing all 2560 test tokens (16 speakers x 10 utterances x 8 testing sessions x 2 repetitions) against the GMMs and the FGMMs of all 16 speakers in the database. The degree of fuzziness $m = 1.06$ was chosen for FGMMs, therefore the decision rule in (13) was used for both models. The experimental results are presented in Table 1. The identification error rate of FGMMs was about 2% lower than that of GMMs using 64 and 128 mixtures [25].

<table>
<thead>
<tr>
<th>Number of mixtures</th>
<th>Identification Error Rate (%) for GMMs</th>
<th>Identification Error Rate (%) for FGMMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>22.53</td>
<td>22.05</td>
</tr>
<tr>
<td>64</td>
<td>18.59</td>
<td>16.48</td>
</tr>
<tr>
<td>128</td>
<td>14.97</td>
<td>12.63</td>
</tr>
</tbody>
</table>

Table 1: Speaker Identification error rate (%) performed on 16 speakers using the TI46 database.

For the ANDOSL database, the set of 108 native speakers was used. For each speaker, the set of 200 long sentences was divided into two subsets. The training set including 20 sentences numbered from 001 to 020 was used to train GMMs and FGMMs of 4, 8, 16, 32, 64, and 128 mixtures. The test set including 180 sentences numbered from 021 to 200 was used to test the above models. Speaker identification was carried out by testing all 19440 tokens (180 utterances x 108 speakers) against GMMs and FGMMs of all 108 speakers. The degree of fuzziness for FGMMs was chosen between $m = 1.025$ and $m = 1.06$. A set of mixed models obtained from GMMs and FGMMs called generalised GMMs (GGMMs) [26] was selected and tested together with GMMs and FGMMs. The decision rule in (13) was used for these models. The experimental results are presented in Table 2. Results for the ANDOSL corpus are better than those for the TI46 corpus since a larger training data set was used.

<table>
<thead>
<tr>
<th>Number of mixtures</th>
<th>Identification Error Rate (%) for GMMs</th>
<th>Identification Error Rate (%) for FGMMs</th>
<th>Identification Error Rate (%) for GGMMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6.89</td>
<td>7.73</td>
<td>6.84</td>
</tr>
<tr>
<td>8</td>
<td>2.29</td>
<td>2.61</td>
<td>2.25</td>
</tr>
<tr>
<td>16</td>
<td>0.85</td>
<td>0.84</td>
<td>0.81</td>
</tr>
<tr>
<td>32</td>
<td>0.33</td>
<td>0.31</td>
<td>0.29</td>
</tr>
<tr>
<td>64</td>
<td>0.15</td>
<td>0.15</td>
<td>0.12</td>
</tr>
<tr>
<td>128</td>
<td>0.13</td>
<td>0.13</td>
<td>0.11</td>
</tr>
</tbody>
</table>

Table 2: Identification error rate (%) performed on 108 speakers of the ANDOSL speech data corpus.

5.4 Speaker Verification Experiments

Speaker verification was performed for the TI46 database. The ANDOSL was not used since all of 200 tokens for each speaker had been recorded in the same session. The training phase was the same as the one in speaker identification. Speaker verification was carried out by testing all 2560 test tokens (16 speakers x 10 utterances x 8 testing sessions x 2 repetitions) against the GMMs and the FGMMs of all 16 speakers in the database. The similarity normalisation method proposed by Matsui and Furui in (17) was used, where $B = 8$ was chosen (speakers who are the same gender with the claimed speaker). Experimental results are presented in Table 3.

<table>
<thead>
<tr>
<th>Number of mixtures</th>
<th>Equal error rate (%) for GMMs</th>
<th>Equal error rate (%) for FGMMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>6.45</td>
<td>6.03</td>
</tr>
<tr>
<td>64</td>
<td>4.89</td>
<td>4.12</td>
</tr>
<tr>
<td>128</td>
<td>3.75</td>
<td>3.75</td>
</tr>
</tbody>
</table>

Table 3: Equal error rate (%) performed on 16 speakers using the TI46 database.

6. Conclusion

The fuzzy Gaussian mixture model has been proposed and evaluated for text-independent speaker identification and verification in this paper. Experimental results indicate that the degree of fuzziness $m$ plays an important role for applying this model. The GMM is regarded as the FGMM with $m = 1$. With values of $m$ slightly greater than one, the FGMM shows better results than the GMM. This can be explained based on the existence of the exponent $m$ in the estimation parameter equations for the FGMM. The exponent $m$ reduces contributions (determined by the membership functions) to the cluster of vectors far from the cluster. This is similar to flattening the tails of Gaussian distributions.
7. References


