System of Speaker Identification Independent of Text for Romanian Language based on Gaussian Mixture Models extract from the MFCC vectors

Marieta Gâta and Gavril Toderean

Abstract – *The speaker recognition technique used in this study is based on GMM-based approach, which is the state-of-the-art for speaker recognition. This approach consists in three phases: a parameterizations phase, a model training phase, a classification phase. We compare some unknown speech, provided from an unknown speaker, with the models of speaker already calculate through EM algorithm for GMM. We choose which speaker from a closed set produced the speech sample.*

Keywords: *Gaussian Mixtured Models, Speaker Identification, MFCC, Covariance Matrix.*

I. GAUSSIAN CLASSIFIERS

A. Euclidean distance

Having addressed the issue of turning the input speech signal into a sequence of feature vectors, we now begin to look at the problem of comparing an unknown input signal with a stored model. A feature vector might consist of 10 or 50 numbers which represent the essential properties of the input signal. Comparing two values of a single feature like pitch has an obvious solution: subtract one value from the other, the smaller the difference the closer the two values are to one another. Moving to two features, we can easily visualise a solution by plotting values on a graph. We can measure the distance between the points on the graph to give a direct comparison of the two feature vectors. This distance can be calculated from the difference between this two values and is known as the Euclidean distance. The Euclidean distance measure is the simplest measure in general use but it's not the only way of comparing two feature vectors. Since calculating the Euclidean distance involves taking square roots it is quite expensive to calculate and requires floating point calculations. An alternative is to use the city block or Manhattan distance which is just the sum of the differences in each dimension, in two dimensions this is the distance of the shortest path if you're limited to a grid as in Manhattan [1].

B. Gaussian Classifiers

Having defined a way of comparing two feature vectors we have one method of classifying an unknown feature vector: by comparing it with a set of known prototype vectors selected to be typical. The closest known vector will define the identity of the unknown vector. Two problems with the prototype approach are that it assumes that we have some way of finding a prototype that is representative of each category as a whole and it takes no account of the variability of phoneme categories. Both of these problems are solveable in various ways and the technique of Gaussian modelling addresses both of them while providing a theoretically sound method of making a decision between categories.

C. The Mean Vector

The first problem of which prototype vector to choose to compare unknowns to has a simple answer: we can take the mean of a set of vectors as the prototype. The mean is by definition maximally close to all of the set of points used to calculate it, with a suitable sample of phonemes the mean should be representative of the category. The set of vectors we use to derive the mean is known as the training set. When we come to evaluate our models we will use a second set of data, the testing set.

D. The Covariance Matrix

A Gaussian model is an extension of the one dimensional normal curve. The normal curve shows the distribution of values for some variable around a mean position. Any normal curve can be described by two parameters, the mean, which defines the centre of the curve, and the variance which defines the width of the curve. In more than one dimension, a Gaussian model is characterised again by two values a mean, which is now a vector rather than a single value, and a covariance matrix. The covariance matrix describes not only the variance of each dimension but the way that the vary together: for example if two dimensions are correlated they will have a hight covariance value.

E. Making use of the Gaussian Model

The Gaussian model can be used to characterise a group of feature vectors of any number of dimensions with two values: a mean vector and a covariance matrix. They key is to understand the Gaussian model as a probabilistic model. The basic theory of Bayesian probability theory which is based on counting the relative occurences of observations. The measure of interest is the posterior probability that a token belongs to a type given an observation (P(type|observation)). It's hard to reliably estimate this measure but we can calculate it given two other measures of the prior probabilities of the type and observation and the conditional probability P(observation|type). The Gaussian model is one way of calculating P(observation|type). Each type is characterised by a single Gaussian and the probability is calculated by measuring the height of the curve (or more generally the value of the function) at the coordinates specified by the input vector. This conditional probability measure is converted to a posterior probability via Bayes formula using the prior probabilities of the observation and the type. The posterior probability can be used in the same way as the Euclidean distance measure we developed earlier to compare an unknown against a set of models. However, now the models are Gaussian models rather than single points and the probability will be higher if the unknown is close to the model. The decision between types can be made by finding the model for which the probability measure is largest. The use of a Gaussian model gives us a solution to the second problem of taking the shape of the distribution of vectors into account. The shape of the distribution is encoded in the mean and covariance of the Gaussian curve and so the probability calculation will take account of this [2].

F. Training Statistical Models

We are estimating a mean and covariance matrix for each type under study. For a 30 feature input vector, the mean consists of 30 numbers and the covariance matrix of 30*30 or 900 numbers. A general statistical principle applies here which says that any summary statistic needs to be supported by a sufficient amount of data. For example if we have only 10 input vectors to train a particular Gaussian model then these 10 sets of 30 numbers will be the entire support for the 930 numbers required for the model; this may well cause the models to be very innacurate. We will often seek to examine the usefulness of a new feature we've developed. Adding this new feature might add another five elements to the overall feature vector, in the example above this takes us from 930 to 35+(35*35)=1260 numbers for each model. If the same number of training vectors is used this will mean that they are spread more thinly and that our new model may be less accurate and perform more poorly, rather than better after the new parameter is added. Any statistical model needs to be trained with an adequate amount of data. We should be aware of the number of free parameters in any model you are training and be mindful of the relation between the number of training vectors and the complexity of your model.

G. Summary

To summarise then, we can take a set of training vectors each representing a different type of speech sound and for each type calculate the mean and covariance matrix for a Gaussian model. This model can then be used to find the probability of any unknown vector and the unknown can be assigned to the model which provides the highest probability.

II. SPEAKER IDENTIFICATION AND VERIFICATION

A. System Outline

The structure of a speaker recogniser is similar to that of a speech recogniser in that the flow of control is divided into parameterisation, pattern matching and decision making. The speaker identification and verification problem is to verify an individual based on a sample of speech. In speaker verification we are given a claim of identity and must verify that the claim is true; in speaker identification we must choose which speaker from a closed set produced the speech sample. These problems are obviously related but have different decision criteria and different requirements for speaker modelling. The basic process is to build a model of the speech of each individual using some statistical technique and match incoming speech samples with these models. Text dependant identification uses a known utterance, for example a password or combination digit sequence, which is predefined for each speaker. Text independant identification uses a different response for each identification attempt. In this latter case, the input could be the answer to a question or might just be some speech sampled from a conversation with the user.

B. Speech Parameterisation

In speech recognition we were concerned with retaining the parts of the speech signal that conveyed information about the phonetic content of the signal. Information about the source was therefore removed since it contributes an independent information steam (pitch). In speaker recognition we wish to retain information about the speaker's identity; as it turns out the requirements are almost identical to those for speech recognition. We would like to ignore variations such as different pitch, speaking rate, environment or communication channel in the same way as we do for ASR. The end result is that the parameterisation used for speaker recognition is the same as that for ASR, typically Mel frequency cepstral coefficients. All the speech material is parameterized as follows: each signal is characterized by 12 MEL frequency cepstral coefficients (MFCC). These MFCC coefficients are obtained from 20 filter bank coefficients applied on 20ms Hamming windowed frames at a 10ms frame rate. The first derivatives of the MFCC coefficients are added to the parameter vectors.

C. Speaker Modelling

In speech recognition, we build models for the words or phonemes that we wanted to recognise. In speaker recognition, we need a model of the acoustic properties of the speaker's voice. Our requirement is the same in each case, we need to be able to compare some unknown speech with the model to produce a distance or probability measure for that speaker. While the temporal nature of speech signals is vitally important in speech recognition, it is not necessarily so important in a speaker model. A simple and effective speaker model can be made by building a probabilistic model of the distribution of input vectors for a speaker. For example, we might build a Gaussian model from the MFCC vectors corresponding to 15 seconds of each speakers speech. A simple Gaussian model is unlikely to be appropriate since the MFCC data is unlikely to follow a simple normal distribution. More appropriate is a mixture of Gaussians as commonly used within HMM states or even a neural network based model. The distance score for a section of input speech is then computed from the product of the probability densities for each input vector. The GMM models are built as follows: a generic GMM model is first estimated with EM (Expectation Maximization) algorithm, maximizing the Maximum Likelihood criterion (ML) on a Romanian read-speech corpus composed of 2 female and 1 male speech utterances of 1 minutes each. Expectation-Maximization (EM) is a well-established maximum likelihood algorithm for fitting a mixture model to a set of training data. We use EM algorithm to optimize the parameter estimation iterative. It should be noted that EM requires an a priori selection of model order. Often a suitable number may be selected by a user, roughly corresponding to the length of the training utterances.

D. Decision Making

If the application is closed to set speaker identification then the decision making process is similar to that in speech recognition: we select the candidate model with the smallest distance or largest probability measure. In speaker verification applications however, the decision as to whether to accept the claim of identity is more complicated since this is an open set problem. The simplest solution to the verification problem is to set a threshold distance and accept the claim if the distance to the claimed model is below the threshold. If this is done, the security of the system can be varied by changing the threshold. In a secure system, the threshold is set very low which should result in fewer false acceptance (FA) errors and more false rejections (FR). In a less secure but more convenient system, the threshold is set higher resulting in more FA errors and fewer FR errors. During this phase, an input signal is presented to the system and compared to the N GMM models depending on the targeted task. This comparison relies on an averaged frame-based likelihood computation between a given model and the input signal. The text used for learning classes is not used for testing. In other words, the speakers read some text in the training phase and another text in the testing phase [3].

III. DECISION RULE

The front-end used in many speaker recognition systems extracts, from the input signal, a set of coefficients based on a Mel-cepstrum technique. In order to improve the system performance, we want to include as many speakers' characteristics as possible, such as dynamic cepstrum features (delta cepstrum, etc). For identification, each speaker is represented by his/her GMM, which is parameterized by the mean vectors, covariance matrix and mixture weights from all component densities. An initial model can be obtained by the estimating of parameters from the clustered feature vectors whereas proportions of vectors in each cluster can serve as mixture weights. Means and covariances are estimated from the vectors in each cluster. After the estimation, the feature vectors can be reclustered using component densities (likelihoods) from the estimated mixture model and then model parameters are recalculated. This process is iterated until model parameters converge. This algorithm is called Expectation Maximization (EM). In identification phase, mixture densities are calculated for every feature vector for all speakers and speaker with maximum likelihood is selected as the author of a speech sample. The GMM has several forms depending on the choice of covariance matrix. The model can have covariance matrix per one component density, per one speaker or shared for all speakers. In template matching, the speaker model with smallest matching score is selected, whereas in stochastic matching, the model with highest probability is selected. Here, given feature vectors of the test utterances of an unknown speaker (placed in ith row of covariance matrix) and GMM parameters of n speakers, the recognition decision should be the jth speaker if the jth element from the ith row of covariance matrix is maxim.

A. Summary of the EM algorithm for GMM is:

Start from M initial gaussian models N($\mu_k \Sigma_k$), k=1,…, M, with equal priors set to P(q_k|θ)=1/M.

Do

Step1. Estimation: compute the probability P(q_k|xn, θ) for each data point xn to belong to the mixture q_k:

$$P(q_k \mid x_n, \theta) = \frac{P(q_k \mid \theta) * p(x_n \mid q_k, \theta)}{p(x_n \mid \theta)} = \qquad (1)$$

$$= \frac{P(q_k \mid \theta) * p(x_n \mid \mu_k, \Sigma_k)}{\sum_j p(q_j \mid \theta) * p(x_n \mid \mu_j, \Sigma_j)} \qquad (2)$$

In the algorithm:

$$c(k)=P(q_k|\theta) \qquad (3)$$

$$lBM(n,k)=\log p(x_n|q_k, \theta) \qquad (4)$$

$$lB(k)=\log p(x_n|\theta) \qquad (5)$$

$$gam\_n(n,k)=P(q_k|x_n, \theta) \qquad (6)$$

Step 2. Maximization:

-update the means:

$$\mu_k^{new} = \frac{\sum_{n=1}^{T} x_n p(q_k \mid x_n, \theta)}{\sum_{n=1}^{T} P(q_k \mid x_n, \theta)} \qquad (7)$$

-update the variances:

$$\sum_{k}^{new} = \frac{\sum_{n=1}^{T} P(q_k \mid x_n, \theta)(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T}{\sum_{n=1}^{T} P(q_k \mid x_n, \theta)} \qquad (8)$$

-update the weights: $P(q_k^{new}) = \frac{1}{T}\sum_{n=1}^{T} P(q_k \mid x_n, \theta)$ (9)

In the algorithm:

new_mu(:,k)= μ_k^{new} (10)

new_sigm(:,k)=Σ_k^{new} (11)

new_c(k)=P(q_k^{new}|θ^{new}) (12)

Step 3. Go to Step 1 until the total likelihood increase for the training data falls under some desires threshold.

IV. RESULTS

The experiments results reported in Fig. 1, 2 and 3 are from three speakers after training the three models, begin testing these three models. The length of training utterance for each speaker is about then 60 seconds and the length of training is about 15 seconds. The exact length is present in Table 1. We compared the performance for different test utterance length and different model order. The texts used for training and testing the models are different. We verified empirically the hypothesis that most of the significant correlations between elements of matrix of covariance are between elements of the same index in the low dimensional space. After reading training data, reading test data, feature extraction for the training data, feature extraction for the testing data, the results for three speaker are:

TABLE 1. Length of training and testing utterance for each speaker.

	Speaker1	Speaker2	Speaker3
testing	13.75 s	11.5 s	14.75 s
training	56.5 s	37.75 s	65.75 s

$$A = \begin{pmatrix} -13.9134 & -15.6080 & -21.0307 \\ -15.5851 & -15.5292 & -19.7973 \\ -19.5541 & -18.7163 & -11.8152 \end{pmatrix} \qquad (13)$$

We compared all the speakers. In covariance matrix A, each column i represents the test recording of speaker i, and each row i represents the training recording of speaker i. The diagonal elements corresponding to same speaker comparisons. We train all the models with the input data. The values for log-likelihood, in the training process, are presented in Table 2. The number of Gaussian is 2. The type of the file for speech sound is wav. We extract Mel-cepstral features from each speaker and we create a model for each speaker. The Fig. 1, 2 and 3 represents the distributions of coefficients for 3 different speakers (implicit the three models for that three speakers). The number of coefficients for a model of speaker is 9. Fig. 1, 2, 3 plots some normalized histograms, id est follows function likelihood of distribution. The model computes MultiGaussian likelihood. We have rectangular or Hanning or Hamming window (by default) in time domain. We applied, by choice, a triangular shaped filter (default), Hanning shaped filters or Hamming shaped filters, all in Mel domain. The Mel-spaced filterbank have 20 filters, length of FFT is 256,

TABLE 2. Values for log-likelihood after 10 iteration for each speaker.

for the first speaker	for the second speaker	for the third speaker
-28.838067 -15.916278 -15.713057 -15.578878 -15.498901	-41.885106 -15.618704 -15.474450 -15.409643 -15.367773	-29.311633 -11.691684 -11.310413 -11.205477 -11.119923

-15.441164	-15.331639	-11.066638
-15.379581	-15.296762	-11.041644
-15.266943	-15.262024	-11.025440
-15.013881	-15.227163	-11.010989
-14.639453	-15.191630	-10.999509

rate of sampling is 8000 (if is not specify 11025). The Mel cepstrum has 8000 (if it is not specify11025 as sample rate, window hamming, 12 cepstral coefficients by default (without the 0th coefficients), the length of frame is less then 30ms (power of 2). As parameters we have: filter in power domain, filter in absolute magnitude domain (default), DCT, cepstral coefficients (with 0th coefficients), log energy, delta-delta coefficients, delta coefficients. We want to verify if the variances vector (the diagonal of the covariance matrix) have the biggest of the elements from the row of the covariance matrix. The values for log-likelihood after 10 iteration are presented in Table 1. Initial log-likelihood is $-9*10^{99}$, this means infinite. The speaker who obtained the maximum score is associate to the unknown speaker.

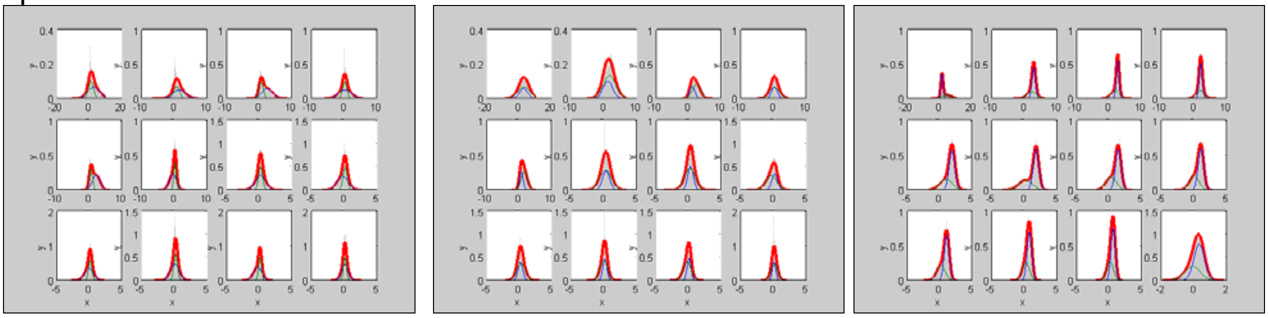

Fig. 1, 2, 3. Distribution probability for parameters of speaker1, 2, 3

V. ALTERNATIVES AND CONCLUSIONS

Our results associate the speaker1 to the model of speaker1, the speaker2 to the model of speaker2, the speaker3 to the model of speaker3. This means that our program create good models. These models recognize a speaker who has already a model create by GMM, even if the text, read by the speaker, in the training process and in the testing process are different. Our results are original, in Romanian languages the use of the GMM models stochastic processes for speaker identification, which underlie speech signal, and therefore, it produces more accurate speaker model for robust identification, are at the begining. Some results were made by Lupu E. and Pop G.P but for different situation (text dependent methods). The results presented here developed for Romanian languages are a new path which can be materialized in future with a large database.

REFERENCES

[1] S. Cassidy, Speech Recognition, Speech Hearing and Language Research Centre, Macquarie University, 2001.

[2] C. Fredouille, G. Pouchoulin, J.F. Bonastre, M. Azzarello, A. Giovanni, A. Ghio, Application of Automatic Speaker Recognition techniques to pathological voice assessment.

[3] Q. Jin, A. Waibel, Application of LDA to Speaker Recognition.

ABOUT THE AUTHORS

Lecturer Marieta Gâta, Department of Mathematics and Computer Science, University of Baia Mare, Baia Mare, Romania, Phone: 0040-262-427466, E-Mail: marietag@ubm.ro

Prof. Gavril Toderean, PhD, Department of Communications, Technical University of Cluj Napoca, Romania, Phone: 0040-264-596285, E-Mail: Gavril.Toderean@com.utcluj.ro