

# On the Number of Gaussian Components in a Mixture: An Application to Speaker Verification Tasks

Mijail Arcienega and Andrzej Drygajlo

Signal Processing Institute  
Swiss Federal Institute of Technology, Lausanne

mijail.arcienega@epfl.ch, andrzej.drygajlo@epfl.ch

## Abstract

Despite all advances in the speaker recognition domain, Gaussian Mixture Models (GMM) remain the state-of-the-art modeling technique in speaker recognition systems. The key idea is to approximate the probability density function (*pdf*) of the feature vectors associated to a speaker with a weighted sum of Gaussian densities. Although the extremely efficient Expectation-Maximization (EM) algorithm can be used for estimating the parameters associated with this Gaussian mixture, there is no explicit method for predicting the best number of Gaussian components in the mixture (also called order of the model). This paper presents an attempt for determining the “optimal” number of components for a given feature database.

## 1. Introduction

When using GMMs for modeling speakers, the *pdf* for a multi-dimensional feature vector  $\vec{x}$  is given by:

$$p(\vec{x}|\lambda_M) = \sum_{k=1}^M w_k \cdot \mathcal{N}(\vec{\mu}_k, \Sigma_k), \quad (1)$$

where  $\vec{\mu}_k$  is the mean vector,  $\Sigma_k$  the covariance matrix and  $w_k$  is the weight for the  $k^{\text{th}}$  Gaussian,  $k = 1, \dots, M$ . The model  $\lambda_M$  is said to be the set of parameters:

$$\lambda_M = \{w_k, \vec{\mu}_k, \Sigma_k\}, \quad k = 1, \dots, M. \quad (2)$$

The GMMs were introduced into speaker recognition tasks by Reynolds. In one of his first works [1], he conducted several identification experiments with different model orders ( $M = 2, 4, 8, 16, 32, 64$ ), different training set sizes as well as different test set sizes. He concluded that when using 6000 25-dimensional Mel-cepstral vectors, at least 16 Gaussian components should be used for capturing speaker specific characteristics. These results were obtained by observing the percentage of correct identifications only after the identification experiments were performed.

From a mathematical point of view, it is important to point out that the order of the mixture for representing a generic *pdf* is finite only if the stochastic process is defined by a *pdf* as in Equation 1; otherwise, the number of components  $M$  necessary to represent the *pdf* is equal to  $\infty$  [2].

Numerous papers suggest iterative methods for finding this optimum number of components [3, 4, 5, 6]. Most of these works aim at finding the order of the mixture that have previously generated a given set of samples, often called *training samples*. In [6], the Markov Chain Monte Carlo (MCMC) method is used to find an *a posteriori* distribution of the order

$k$  of the mixture. However, this method requires prior distributions assumptions for which the parameters may not be available. In [3], filtered kernel estimates and mixture density estimates are alternated in an iterative process where  $M$  is incremented until a distance (computed between two consecutive approximations) falls below a fixed threshold. The drawback in this method is that kernel density estimates need some *a priori* knowledge about the *pdf* for fixing the bandwidth  $h$ , also known as the “smoothing parameter”. *Pdf* approximations are indeed strongly dependent on this parameter.

When using GMMs for modeling speakers, one more constraint must be taken into account: estimated models will be used to classify further observed data. Therefore the order of the mixture that actually generated the training data may be different from the order that will give the best classification performance. For example, over-fitting may allow a precise representation of training data, but degrades the performance of the recognizer.

In this paper, an attempt for finding an optimum order in an automatic way is explored. This order is optimum in the sense that it provides the best recognition results.

In the sequel of the paper, section 2 presents the characteristics of the EM likelihood optimization approach. Section 3 introduces the proposed method for finding the best  $M$  and Section 4 presents the results as a consequence of this choice in the framework of speaker verification experiments.

## 2. Expectation-Maximization and Log-Likelihood

Following the EM algorithm [7], the goal is to maximize the log-likelihood

$$L(\lambda_M|X) = \log(p(X|\lambda_M)), \quad (3)$$

where  $X$  is the set of vectors  $\vec{x}$  that form the training data. This maximization is done in an iterative manner, finding a better estimate of  $\lambda_M$  at each step. One of the most important properties of the algorithm is that it guarantees that the log-likelihood increases monotonically at each iteration and that it will eventually converge. Initialization is primordial since the algorithm converges towards a local maximum.

In order to understand the evolution of this maximized log-likelihood as a function of the number of components, the following experiment has been performed. Artificial training sets have been generated from Gaussian mixtures of orders  $M_t(l) = 2^l$ ;  $l = 1, \dots, T$ . Then, the EM algorithm has been used in order to fit the training set generated by the mixture of order  $M_t$  with Gaussian mixtures of orders  $M_e(j) = 2^j$ ;  $j = 0, \dots, l + 4$ . Figures 1, 2 and 3 present these results for

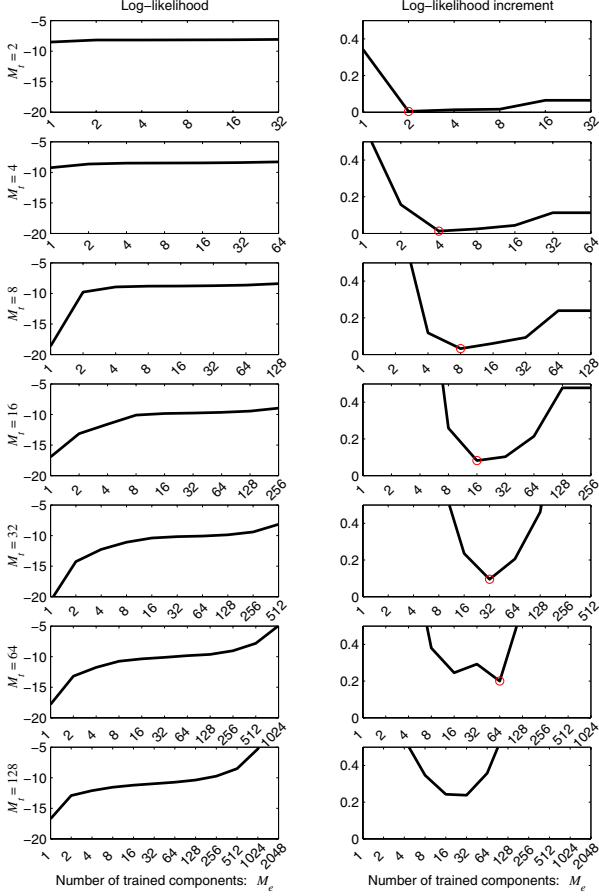


Figure 1: The log-likelihood  $L(\lambda_{M_e}|X)$  and the log-likelihood increment  $\delta_{M_e}$  for a training set of  $Q = 2000$  samples

training sets of size  $Q = 2000, 4000$  and  $6000$  samples respectively. The difference between two consecutive log-likelihoods is also represented.

In the first column of graphs in Figures 1, 2 and 3, the log-likelihood starts growing fast and then seems to reach a certain value in the flat part of the curves, but then instead of remaining constant, it starts increasing again. This phenomenon is independent of the order of the mixture that generated the training samples. There is however a strong correlation between this phenomenon, the number of components during the training procedure and the size of the training set. The increase of the log-likelihood after the flat part is, in fact, the result of over-fitting the model. Hence, we can already see that in order to avoid over-fitting,  $M$  should not be greater than  $\sim Q/100$ .

### 3. The Proposed Algorithm

The second column of graphs in Figures 1, 2 and 3 displays the increment of the likelihood for two successive maximized estimations

$$\delta_{M_e(j)} = L(\lambda_{M_e(j+1)}|X) - L(\lambda_{M_e(j)}|X). \quad (4)$$

The circled point reminds the reader of the number of components that generated the training set. As we can see, this point corresponds, as far as there is no over-fitting, to the minimum of the function  $\delta_{M_e}$ . When the model over-fits the data, the  $\delta_{M_e}$

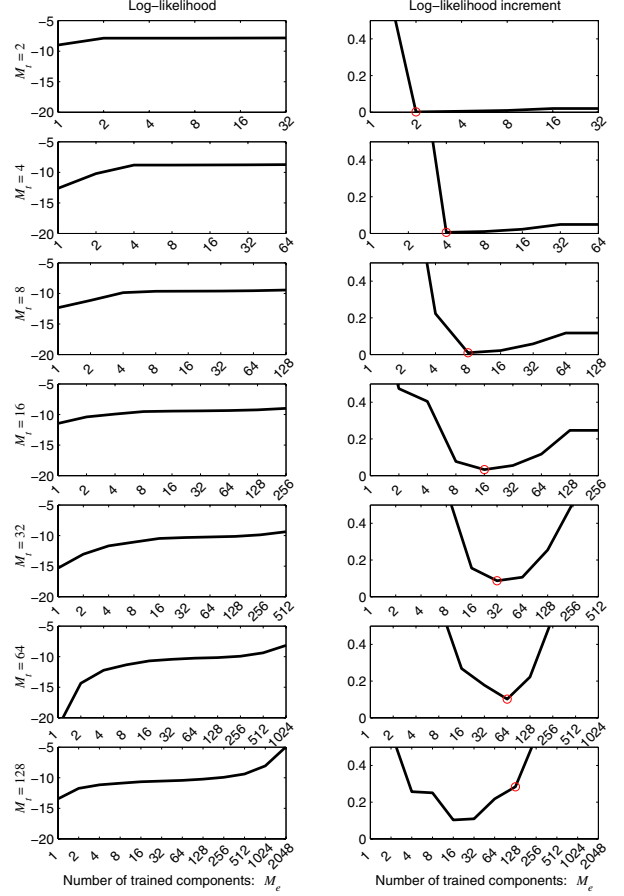


Figure 2: The log-likelihood  $L(\lambda_{M_e}|X)$  and the log-likelihood increment  $\delta_{M_e}$  for a training set of  $Q = 4000$  samples

function, after decreasing, starts to increase again; however, its minimum remains at the  $M_e(j_b)$  which will prevent the undesired over-fitting. In this case, the minimum  $M_e(j_b)$  depends only on the training set size and we can observe that it moves toward higher values when this size increases. The function  $\delta_{M_e}$  is the sampled version of the Kullback-Leibler distance [8], which measures the relative entropy, a measure of the difference between two probability distributions.

Another example presented in Figure 4 shows that the minimum of  $\delta_{M_e}$  also exists when the distribution of  $X$  is not a mixture of Gaussians. In this example, 4000 points from an uniform distribution have been used to train a model of order  $M_e(j)$ . The second graph in this Figure shows 15 dashed curves corresponding to 15 different initializations of the EM algorithm. By taking the average (dark line) we see that the minimum is around 16 components, just before the over-fitting phenomenon starts.

From the log-likelihood observations we can suggest that taking the minimum of the  $\delta_{M_e}$  function will lead us to the “best”  $M_b$  order of the mixture in the sense that it will not over-fit the model and catch the correct number of Gaussians, if this number is finite. Hence,

$$M_b = M_e(j_b) = \arg \min_{M_e(j)} \{\delta_{M_e(j)}\}. \quad (5)$$

An algorithm for finding  $M_b$  can be represented as follows:

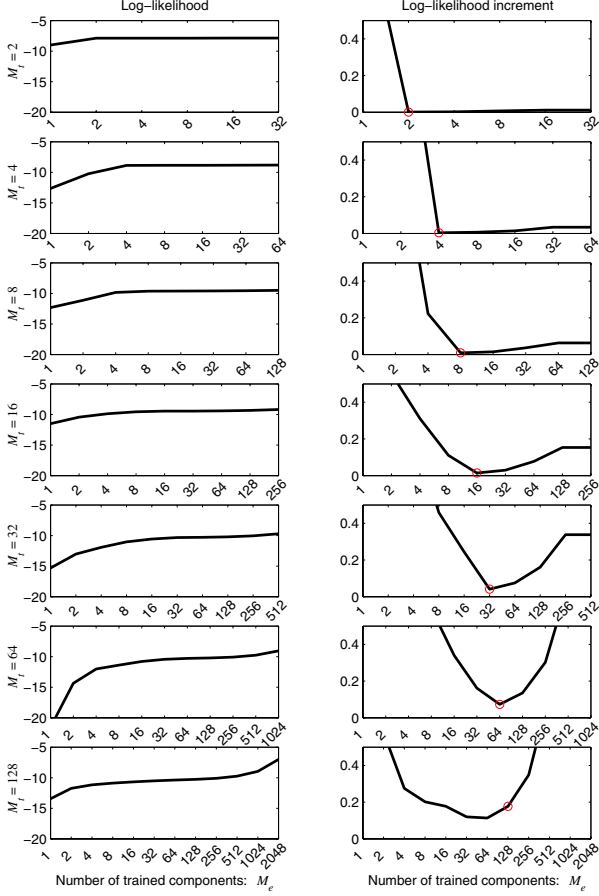


Figure 3: The log-likelihood  $L(\lambda_{M_e}|X)$  and the log-likelihood increment  $\delta_{M_e}$  for a training set of  $Q = 6000$  samples

```

1  begin initialize  $j \leftarrow j_0, \delta_{M_e(j_0-1)} \leftarrow \infty$ 
2    EM algorithm:
       $L(\lambda_{M_e(j_0)}|X) \leftarrow \max_{\lambda_{M_e(j_0)}} L(\lambda_{M_e(j_0)}|X)$ 
3    do  $j \leftarrow j+1$ 
4      EM algorithm:
         $L(\lambda_{M_e(j)}|X) \leftarrow \max_{\lambda_{M_e(j)}} L(\lambda_{M_e(j)}|X)$ 
5      compute  $\delta_{M_e(j-1)}$ 
6      until  $\delta_{M_e(j-1)} > \delta_{M_e(j-2)}$ 
7      return  $M_e(j-1)$ 
8  end

```

The EM depending on the initialization parameters, a non fully maximized  $L(\lambda_{M_e(j)}|X)$ , may stop iterations before we reach the minimum. In order to avoid this,  $\delta_{M_e(j)}$  can be replaced by an expectation  $E[\delta_{M_e(j)}]$ , obtained for example by initializing the model  $\lambda_{M_e(j)}$  in different manners.

#### 4. Application: Speaker Verification System

In section 3,  $M_b$  has been proposed as the optimum number of components for modeling a training data set.  $M_b$  is optimum in the sense that it will correspond to the effective order of the mixture that generated the training data set, or, it will suggest the order to take into account so that the mixture does not overfit the data.

In this section, the results of several speaker verification

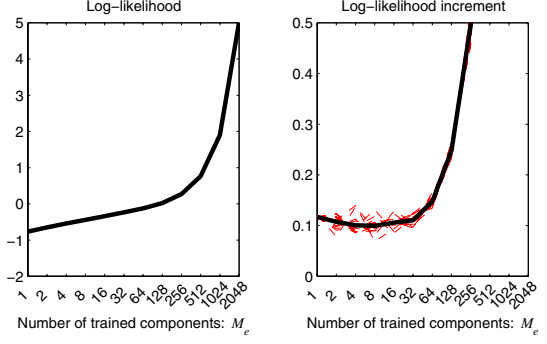


Figure 4: The log-likelihood  $L(\lambda_{M_e}|X)$  and the log-likelihood increment  $\delta_{M_e}$  when fitting a uniform distribution with a mixture of Gaussians

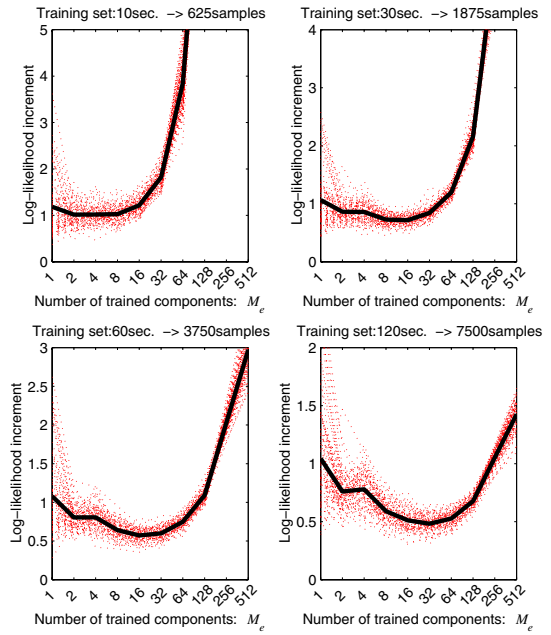


Figure 5: The log-likelihood increment  $\delta_{M_e}$  for different training set sizes of 13-dimensional MFCCs

tests are presented. 40 speakers from the Switchboard database were used as clients and 48 speakers were used to build a world model. The order of the model changes from 1 to 512. Utterances of 10, 30, 60, and 120 seconds have been used to fit the models. The acoustic feature vectors, 13-dimensional Mel Frequency Cepstral Coefficients(MFCC), were extracted every 16 ms, which leads to training sets of  $Q = 625, 1875, 3750$  and 7500 samples. During the exploitation phase the length of the test utterances was fixed to 3, 6, 10, or 15 seconds.

Before performing the tests, let us use the function  $\delta_{M_e}$  to predict the optimum order. Figure 5 shows the shape of this function for different training set sizes. The dashed curves represent the  $\delta$  function for a speaker and the dark line is the overall average. These graphics tell us that, for example, when having only 30 seconds of speech for training a model, no more than 16 Gaussian components should be used.

Figure 6 presents the DET curves obtained in these experiments. The test duration was set to 6 seconds. These results

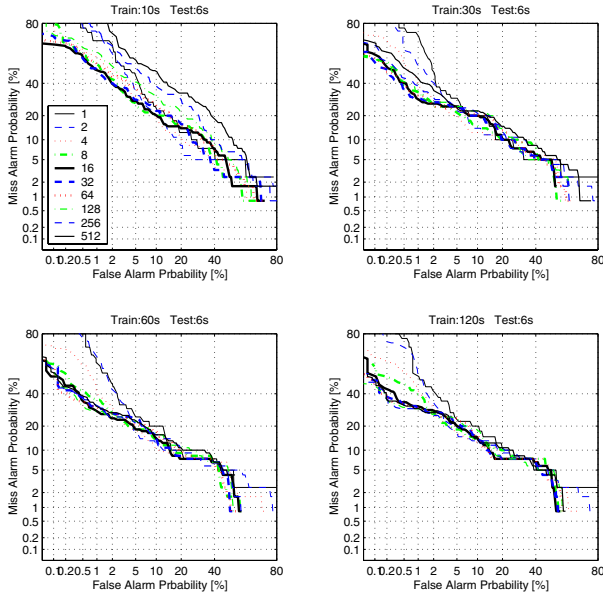


Figure 6: DET curves for different training set sizes and different number of Gaussian components in the model

are representative of all other experiments performed with different test utterance lengths. We can clearly see the effects of over-fitting in the first graph, where models of higher orders give worse performances. In order to better compare these results, a subset of these curves has been chosen and plotted in Figure 7. In particular, the performances for the suggested order  $M_b$  are compared to the higher order in our tests, namely  $M_e(9) = 512$ . As shown in this Figure, the model with the suggested number of Gaussians always performs better than a model with a larger number of Gaussians. When a large amount of data is available for training, the degradations (resulting from using too high of an order) nearly disappear but there is still no improvement in the verification score.

## 5. Conclusions

The proposed method suggests the order of the Gaussian mixture that optimally models a set of training data. The experiments of speaker verification have corroborated this suggestion. The chosen order is optimum in the sense that it optimizes the performance of the classifier (here, the speaker verification system) while keeping complexity of the model as low as possible. It has been shown that the correct order does not depend on the nature of the training data set, and takes into account the number of samples of such a set in order to avoid over-fitting. Moreover, the optimum order depends only on the log-likelihood maximized estimates and does not make any assumptions about the classifier. In particular, this order has been chosen without having any feedback about the performance of the classifier.

It seems that, for the specific set of features that we have used, there is no limit on the choice of this order, as far as we possess enough training data. In the speaker verification experiments, when training the world model, the  $\delta$  function presents a minimum at the values 64, 128, 256, and 512 for 8, 24, 48 and 96 minutes of training data, respectively.

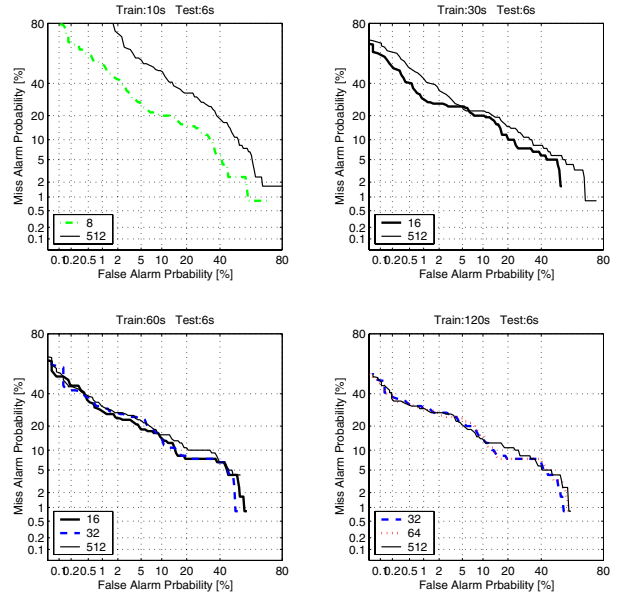


Figure 7: DET curves for different training set sizes. The results for the optimal number of Gaussian components is compared to the results when using 512 Gaussian components

## 6. References

- [1] D. A. Reynolds and R.C. Rose, “Robust text-independent speaker identification using Gaussian mixture models,” *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 72–83, 1995.
- [2] L. F. James, C. E. Priebe, and D. J. Marchette, “Consistent estimation of mixture complexity,” *Annals of Statistics*, vol. 29, no. 5, pp. 1281–1296, August 2001.
- [3] C. E. Priebe and D. J. Marchette, “Alternating kernel and mixture density estimates,” *Computational Statistics and Data*, vol. 35, no. 1, pp. 43–65, 2000.
- [4] C. Kerebin, “Consistent estimation of the order of mixture models,” *Sankhyā, The Indian Journal of Statistics*, vol. 62, no. 2, pp. 49–66, 2000.
- [5] C. E. Priebe and D. J. Marchette, “Adaptive mixture density estimation,” *Pattern Recognition*, vol. 26, no. 5, pp. 771–785, 1993.
- [6] S. Richardson and P. J. Green, “On Bayesian analysis of mixtures with an unknown number of components,” *Journal of the Royal Statistical Society*, vol. 59, no. 4, pp. 731–792, 1997.
- [7] A.P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum-likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.
- [8] S. Kullback and R. A. Leibler, “On information and sufficiency,” *Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 1951.