Variational Bayesian Model Selection for GMM-Speaker Verification using Universal Background Model.

Timur Pekhovsky, Alexandra Lokhanova

Department of Speaker Verification and Identification, Speech Technology Center, St. Petersburg, Russia
{tim, lohanova}@speechpro.com

Abstract

In this paper we propose to use Variational Bayesian Analysis (VBA) instead of Maximum Likelihood (ML) estimation for Universal Background Model (UBM) building in GMM text independent speaker verification systems. Using VBA estimation solves the problem of the optimal choice of the UBM mixture dimensionality for the training data set, as well as the problem of noise Gaussians which are typical for ML estimation. Experiments using the NIST 2006 and 2008 SRE datasets (cellular channels only) demonstrate superior efficiency of baseline verification systems with a UBM trained using the VBA method compared to standard ML training. Verification error was reduced by almost 8%, compared to a baseline system with standard ML training for the UBM.

Index Terms: speaker verification, gaussian mixture model, universal background model, variational bayesian analysis

1. Introduction

In recent years, Gaussian mixture models (GMM) based on Joint Factor Analysis (JFA) [1,2,3] became the dominant approach to text independent speaker verification tasks. One of the key elements of all these GMM systems is a Universal Background Model (UBM) [4].

Currently, a standard procedure for UBM building based on Maximum Likelihood (ML) estimation is traditionally used. Several variations of this procedure exist, starting with the original work [4], where the mixture initially contains 512, 1024 or 2048 Gaussians, which are either initialized using, for example, the K-means algorithm, or simply randomized within the limits of the training set. More complicated procedures are borrowed from the field of speech recognition, where the initial number of Gaussians is equal to 256 Gaussians, then after several EM iterations each Gaussian is split in two, then four, etc., so that the final number of Gaussians reaches 2048 [5].

The most critical problem for UBM building is finding an exact match between the number of UBMs Gaussians and the amount of training material. The ML-criterion tries to train all the Gaussians, even when there are few training data, which leads to overfitting of the GMM to the training set and low system efficiency on an independent test set, demonstrating the poor generalization potential of the system. On the other hand, all large contemporary GMM verification systems that participate in the NIST competition use very large training sets, so that the typical number of Gaussians for a UBM (2048) may in any case be lower than the optimal number of Gaussians allowed by the size of the training set which would guarantee the best efficiency for the verification system.

The easiest way of solving this problem is the cross-validation approach (see example in [6]), where several UBMs with different numbers of Gaussians are built, then each corresponding verification system is tested on an independent test set. However, this approach is very computationally expensive and time-consuming.

In this paper we propose to use Variational Bayesian Analysis (VBA) instead of Maximum Likelihood (ML) estimation when building a UBM. This approach has several advantages over ML-estimation. First, it is possible to establish the optimal number of components in the mixture without resorting to the crossvalidation approach. Besides, there is no overfitting if we choose more components than is necessary for the given amount of training data. Moreover, the singularities that arise in Maximum Likelihood when a Gaussian component ‘collapses’ onto a specific data point, are absent in Bayesian treatment.

2. VBA estimation for a GMM model

2.1. Bayesian approach

The main goal of the Bayesian approach to Gaussian mixture model learning on observed data \( X = \{x_n\}_{n=1}^N \) is to find a good approximation both for the true posterior \( p(Z|X) \) and for the evidence \( p(X) \):

\[
p(X) = \int p(X|Z) \cdot p(Z) \cdot dZ
\]

Here the vector \( Z = [\theta, z] \) is the joint set of parameters \( \theta \) and latent variables \( z \). It is possible to demonstrate that for every normalized distribution \( q(Z) \), it is true that

\[
\ln p(X) = L(q) + KL(q \parallel p),
\]

where

\[
L(q) = \int q(Z) \cdot \ln \left( \frac{p(X, Z)}{q(Z)} \right) dZ,
\]

\[
KL(q \parallel p) = -\int q(Z) \cdot \ln \left( \frac{p(Z|X)}{q(Z)} \right) dZ \geq 0.
\]

Since the values of the Kullback-Leibler divergence \( KL(q \parallel p) \) are non-negative, \( L(q) \) is the lower bound for the evidence \( p(X) \). \( L(q) \) is maximized by means of optimization with respect to the distribution \( q(Z) \). The maximum of the lower bound for \( q(Z) \) is reached when the Kullback-Leibler divergence \( KL(q \parallel p) \) equals zero. This corresponds to the case when \( q(Z) \) equals the true posterior distribution \( p(Z|X) \).

However, often realization with a true posterior distribution is intractable. Instead, we take a restricted distribution family \( q(Z) \) and then search for an element of this family for which the lower bound is maximized.

2.2. Variational inference

The key and only assumption in variational inference is the assumption of the distribution factorization of the joint variational posterior:

\[
q(Z) = \prod_n q(z_n)
\]
\[ Q(Z) = Q(z, \theta) = q(z) \cdot q(\theta) \]  

VBA does not need any assumptions about the form of the distributions \( q(\theta) \) and \( Q(Z) \). Among all the distributions \( Q(Z) \) that have the form (2) we now search for the distribution for which the lower bound \( L(q) \) is the greatest. We do that by maximizing \( L(q) \) with respect to each distribution \( q(\theta) \) and \( Q(Z) \) separately. First \( q(\theta) \) is fixed and we search for the maximum \( L(q) \) for \( q(z) \). By substituting (2) in (1) we get:

\[
L(q) = \int \left[ \int \left[ q(\theta) \ln P(X, Z | \theta) q(z) \cdot q(\theta) \right] d\theta \right] dz =
\]

\[ = -KL(q(z) || \tilde{Q}(\theta)) \geq 0, \]

where

\[
\ln \tilde{Q}(\theta) = \int q(\theta) \ln \frac{P(X, Z | \theta)}{q(\theta)} d\theta =
\]

\[ = E_p \left[ \ln \frac{P(X, Z | \theta)}{q(\theta)} \right] + \text{const} = E_p [P(X, Z | \theta)] + \text{const}' \]

The expression for \( L(q) \) is simply the negative KL-divergence. Thus, \( L(q) \) is maximal when the KL-divergence equals 0 or when \( q(z) = \tilde{Q}(\theta) \). Consequently, the update formula for \( q(z) \) is:

\[ \ln q(z) = E_p \left[ \ln P(X, Z | \theta) \right] + \text{const} \]  

(3)

The update formula for \( q(\theta) \) is the same:

\[ \ln q(\theta) = E_p \left[ \ln P(X, Z | \theta) \right] + \text{const} \]

(4)

where the constants are defined by the corresponding norming conditions.

2.3. VBA-GMM modeling

In this paper we will not use an incomplete Bayesian approach [8] where mixture coefficients \( \pi \) are found using the maximum likelihood method of type II. Instead, we use a complete Bayesian treatment where all parameters \( \theta = \{ \pi, \mu, \Lambda \} \) and latent variables \( \{ z \} \) are given prior distributions [7].

Following [7], we take as a starting point the logarithm of the Likelihood ML-structure function for the complete data set:

\[
L_{\text{ML}} = \sum_{n=1}^{N} \sum_{m=1}^{M} \ln \left[ \sum_{a=1}^{K} \pi_{a,m} \cdot N(x_n | \mu_{a,m}, \Lambda_{a,m}) \right] =
\]

\[ \ln \left[ \prod_{n=1}^{N} \prod_{m=1}^{M} \left( \pi_{a,m} \right)^{r_{a,m}} \right] + \ln \left[ \prod_{n=1}^{N} \prod_{m=1}^{M} \left( N(x_n | \mu_{a,m}, \Lambda_{a,m}) \right)^{r_{a,m}} \right]. \]

Here each data vector \( x_n \) is connected to a set of binary latent variables \( z_{n,m} \in \{0,1\} \), where \( m = 1, \ldots, M \) - number of mixture component, and \( \sum_m z_{n,m} = 1 \). Latent variables \( z_{n,m} \) are given discrete distributions which are governed by the mixture coefficients \( \pi_{a,m} \). Instead of the covariance matrix \( \Sigma \) it is more convenient to work with its inverse called the precision matrix \( \Lambda \). As follows from (3) and (4), we are interested in the joint distribution \( p(X, Z | \theta) = P(X | Z) \cdot P(\theta) \), which for VBA-GMM is the following:

\[ p(X, Z | \theta) = p(X | z, \mu, \Lambda) \cdot p(z | \pi) \cdot p(\pi) \cdot p(\mu, \Lambda). \]

Where it follows from (5) that:

\[
p(z | \pi) = \prod_{n=1}^{N} \prod_{m=1}^{M} \left( \pi_{a,m} \right)^{r_{a,m}} \]

(6)

\[ p(X | z, \mu, \Lambda) = \prod_{n=1}^{N} \prod_{m=1}^{M} (N(x_n | \mu_{a,m}, \Lambda_{a,m}))^{r_{a,m}} \]

(7)

Prior for the \( \theta \) parameters are chosen from the conjugate distributions of the exponential family. For the prior \( p(\mu, \Lambda) \) we take an independent Gaussian-Wishart prior governing the mean \( \mu_{a,m} \) and precision \( \Lambda_{a,m} \) of each Gaussian:

\[ p(\mu, \Lambda) = \prod_{n=1}^{N} (N(\mu_{a,m} | m_{a} (\beta_{a} \Lambda_{a,m})^{-1}), \Lambda_{a,m}) \]

For the prior \( p(\pi) \) for mixture coefficients we take a Dirichlet distribution:

\[ p(\pi) = C(\alpha_{a}) \prod_{a=1}^{K} (\alpha_{a}-1) = \text{Dir}(\pi | \alpha_{a}), \]

(9)

where for the sake of symmetry the same parameter \( \alpha_{a} \) is chosen for each component, and \( C(\alpha_{a}) \) is the normalization constant of the Dirichlet distribution [7]. In (8) \( w \) and \( W \) are the Wishart distribution and the Wishart matrix, respectively [7].

2.4. Variational posteriors

Using (3), for the variational posteriors of the latent variables we get:

\[ \ln q'(z) = E_p \left[ \ln p(X, Z | \theta) \right] + \text{const} =
\]

\[ = \sum_{n=1}^{N} \sum_{m=1}^{M} z_{n,m} \cdot \ln \rho_{n,m} + \text{const} \]

Here the constants contain mathematical expectations independent of \( z_n \) and the values:

\[ \ln \rho_{n,m} = E_p \left[ \ln \pi_{a,m} \right] + \frac{1}{2} E_p \left[ \ln \Lambda_{a,m} \right] -
\]

\[ - \frac{1}{2} E_p \left[ \ln \Lambda_{a,m} \right] \cdot \frac{D}{2} \ln(2\pi). \]

(10)

Here \( D \) is a feature dimension. The variational posterior of the latent variables will have the same form as (6):

\[ q'(z) = \prod_{n=1}^{N} \prod_{m=1}^{M} \left( r_{a,m} \right)^{r_{a,m}} \]

\[ r_{n,m} = \rho_{n,m} / \sum_{k=1}^{M} \rho_{n,k}. \]

The \( r_{n,m} \) values for VBA are fully analogous to the responsibilities in ML-GMM.

In the same way, using (4) it is possible to demonstrate that the variational posteriors of the parameters \( q'(\theta) \) have the same functional form as their corresponding priors in the joint distribution. For the variational posterior of the mixture coefficients we get a Dirichlet distribution:

\[ q'(\pi) = C(\alpha_{a}) \prod_{a=1}^{K} (\pi_{a}-1) = \text{Dir}(\pi | \alpha_{a}) \]

where \( \alpha_{a} = (\alpha_{1,a}, \ldots, \alpha_{a,n}, \ldots, \alpha_{a,M} \). For the variational posterior of the means \( \mu \) and \( \Lambda \) we get the result in the form of a Gaussian-Wishart distribution, as in (7):

\[ q'(\mu_{a,m}, \Lambda_{a,m}) = N(\mu_{a,m} | m_{a} (\beta_{a} \Lambda_{a,m})^{-1}), \Lambda_{a,m}) \]
This is the result of choosing the prior from the conjugate distributions of the exponential family. The hyperparameters \( \alpha, \beta, m, W, V \) of this distributions will be updated on M-step of Variational Bayesian Expectation Maximization (VB EM) algorithm.

2.5. VB EM Algorithm

First we define the variational E-step for the responsibilities \( \mathbb{E}[z_{mn}] r_{mn} \). Using the standard properties of the Wishart and Dirichlet distributions we get:

\[
E_{\pi} (L_{x} \mid \Lambda_{n}) = \frac{1}{2} \ln | \Lambda_{n} | + \frac{1}{2} \ln | W_{n} | + D \ln 2
\]

(11)

Then in the next M-step we hold these responsibilities fixed and use them to calculate the variational distribution of the parameters, using:

\[
\bar{\alpha} = \sum_{n=1}^{N_{\text{n}}} \alpha_{n}
\]

(13)

where \( \psi(\cdot) \) is the digamma function. We find \( r_{mn} \) by substituting (11), (12), (13) into (10) and normalizing them. Afterwards it is necessary to count all the statistics of the \( 0^{\text{th}}, 1^{\text{st}}, 2^{\text{nd}} \) order, using responsibilities \( r_{mn} \):

\[
N_{\alpha} = \sum_{n=1}^{N_{\text{n}}} \frac{r_{mn}}{x_{n}}
\]

\[
F_{n} = \frac{1}{N_{\alpha}} \sum_{n=1}^{N_{\text{n}}} r_{mn}.x_{n}
\]

\[
S_{n} = \frac{1}{N_{\alpha}} \sum_{n=1}^{N_{\text{n}}} r_{mn} (x_{n} - F_{n})^{T}(x_{n} - F_{n})
\]

Then in the next M-step we hold these responsibilities fixed and use them to calculate the variational distribution of the parameters, using:

\[
\alpha_{n} = \alpha_{0} + N_{n}
\]

\[
\beta_{n} = \beta_{0} + F_{n}
\]

\[
V_{n} = V_{0} + N_{n}
\]

\[
m_{n} = \frac{1}{\beta_{n}} \left[ (\beta_{n} m_{n} + N_{n} F_{n}) \right]
\]

(14)

\[
W_{n} = W_{0} + N_{n} S_{n} + \frac{\beta_{n} N_{n}}{\beta_{n}^{2} + \beta_{0} N_{n}} \left[ (F_{n} - m_{n})^{T}(F_{n} - m_{n}) \right]
\]

The equations in (14) are the equations of the variational M-step, during which hyperparameters of the distributions are updated.

3. VBA modeling of the UBM. Implementation Issues.

One of the apparent difficulties of applying VBA to UBM learning is the fact that if we are considering the limit \( N \to \infty \), Bayesian processing with a fixed number of Gaussians \( M \) must converge to the EM algorithm of maximum likelihood. However, in contemporary speaker verification systems, where typically a very large number of mixture components can be used, it is not obvious that VBA model selection becomes meaningless. On the contrary, it is obvious that when an arbitrarily large \( N \) is fixed, and when the number of components \( M \) is increased indefinitely, we will almost arrive at superfluous noise Gaussians. On the other hand, as we have mentioned in the Introduction, the contemporary limit of 2048 Gaussians for a UBM can be safely lowered, which raises the question of optimal model selection even with a large amount of training data.

In VBA method the most robust way of choosing the optimal number of components for a UBM is computation of the lower bound \( L(q) \) for evidence, with a relatively broad noninformative prior \( \alpha_{0} \). For lack of space, we cannot provide the full expression for \( L(q) \) here and refer the reader to [7] (chapter 10). This strategy leads to a very expensive procedure, only slightly better than the crossvalidation approach in terms of time and resources, since we need to learn a set of UBMs with different number of Gaussians and compare their lower bounds to find its maximum.

In our work the criterion for model selection is the property of mixture coefficients for the components that are in fact not responsible for the explanation of the data point – i.e., they have \( r_{mn} \approx 0 \) and \( N_{n} \approx 0 \). Then, as can be seen from (14), the parameters of such weakly data-adapted Gaussians return to their prior values. Based on the properties of the Dirichlet distribution, the VBA-GMM model gives for the expected values of the mixture coefficients in posterior distribution the next expression:

\[
E[\pi_{n}] = \frac{\alpha_{n}}{\bar{\alpha}} = \frac{\alpha_{0} + N_{n}}{M \bar{\alpha} + \sum_{j=1}^{M} N_{j}}
\]

(15)

If we take a Gaussian component fitted slightly to the data points, for which \( N_{n} \approx 0 \) and \( \alpha_{n} \approx \alpha_{0} \), then if the initial prior is wide (\( \alpha_{0} \to 0 \)), it holds that \( E[\pi_{n}] \to 0 \). While if the prior \( \alpha_{0} \to \infty \), then \( E[\pi_{n}] \to 1/M \). In this way we get a fast method for model selection: only one pass of the VB-EM-algorithm over the training data. The choice of the finite initial prior \( \alpha_{0} \) means that the mixture coefficients of the uninformative Gaussians converge to the same small value \( \xi \), where \( 0 < \xi < 1/M \):

\[
\xi \Rightarrow \frac{(\alpha_{0}/N)}{M (\alpha_{0}/N) + 1} \Rightarrow 1/M + N/\alpha_{0}
\]

(16)

We have conducted experiments with different amounts of training data and have found that if the amount is small, uninformative Gaussians very quickly decrease their mixture coefficients down to \( \xi \), and they are easy to distinguish from the rest of the Gaussians. In the case of large amounts of training data, it was necessary to perform relatively many iterations (hundreds) for the uninformative Gaussians to behave in this fashion, so the process converged very slowly. Such behaviour in the latter case is explained not as much by the large \( N \) as by an insufficiently large initial \( M \) which only slightly exceeded the optimal number of Gaussians.

In our realization of the VBA-training of UBM, we start with a typically large number of Gaussians \( M=4096 - 8192 \). Then, in order to speed up the convergence of the VB-EM-algorithm, we initialize the vector \( m_{0} \) for the Gaussians, using \( K \)-means for the training data.

We choose large initial covariance matrices \( \tilde{W}_{0} = 10^{-I} \), so that in the very beginning each Gaussian cannot stay in the initial local maximum. All covariance and precision matrices as well as Wishart matrices are diagonal, since the UBM technology only supports diagonal matrices, which does not influence the fundamental difference between the ML and VBA approaches in any way. The initial values of the other hyperparameters we chose \( \alpha_{0} = \beta_{0} = V_{0} = 1 \).

4. Experimental results and discussion

We used the NIST speaker recognition evaluation (SRE) datasets for evaluating the efficiency of the proposed method.

We used a Baseline GMM system which is a pure GMM system, where speaker models are MAP-adapted from the UBM on speaker models. We used the classic maximum a
posteriori (MAP) adaptation algorithm [4] with the relevance factor \( \tau = 16 \). Only Gaussian means \( \mu_m \) were adapted. In the test we used the standard LLR-estimation of GMM-verification without any channel compensation for the test utterance and without the ZT-norm.

We also used the GMM-SVM system by Speech Technology Center, Ltd (STC), which participated in NIST SRE 2010 [9]. In this system, the GMM block is an ML variant [10] of Vogt’s reduced JFA version [11], and the SVM block, which is based on linear KL-kernel in the mean supervector space.

Both systems used the same preprocessing and VAD-segmentation procedure, as well as MFCC vectors of the STC’s GMM-SVM system described in [9], and UBMs produced by two methods: ML training and VBA raining.

Table 1 shows the results (EER) of our “SMALL” experiment, in which the UBM was trained on a small amount of data: 515 recordings of 103 men from the NIST 2004 telephone database. The values of mixture coefficients of weakly data-adapted components converge to \( \xi = 2.5 \cdot 10^{-7} \).

Testing was performed on the NIST SRE 2006 database (men, telephone). Table 1 demonstrates that the model VBA selection for UBM preserved 708 components out of 1024. The advantage of this selection is more pronounced for the Baseline GMM system than for the hybrid GMM-SVM system, in which the GMM subsystem used 30 eigenchannels for the eigenchannel matrix and 200 male impostors were used for SVM block. This can be explained by the fact that in the latter case, the UBM is used indirectly, only via the GMM subsystem, and the discriminative SVM subsystem strongly disguises the effect in question.

Table 2 shows the results of our “LARGE” experiment (\( \xi = 1.25 \cdot 10^{-4} \)), where the UBM was trained on typical amounts of data used by NIST leaders: about 10000-14000 recordings for one gender.

<table>
<thead>
<tr>
<th>System</th>
<th>UBM training approach</th>
<th>ML</th>
<th>VBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline-GMM</td>
<td></td>
<td>6.38%  (M=1024)</td>
<td>5.89% (M=708)</td>
</tr>
<tr>
<td>GMMSVM</td>
<td></td>
<td>5.17%  (M=1024)</td>
<td>5.05% (M=708)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>UBM training approach</th>
<th>ML</th>
<th>VBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline-GMM</td>
<td></td>
<td>15.55% (M=2048)</td>
<td>14.24% (M=3062)</td>
</tr>
<tr>
<td>GMMSVM</td>
<td></td>
<td>5.78%  (M=2048)</td>
<td>5.15% (M=3062)</td>
</tr>
</tbody>
</table>

Here the training was done on 10179 recordings of SwitchBoard II, NIST SRE 2004, 2005, 2006 (men, telephone). Testing was performed on the NIST SRE 2008 database (men, telephone). Table 2 demonstrates that the model VBA selection for UBM preserved 3062 components out of 4096. So this amount of training data makes it possible to use more than the typically accepted 2048.

Again, the advantage of this model selection is greater for the Baseline GMM system than for the hybrid GMM-SVM system, where now 100 eigenchannels for the eigenchannel matrix and 1000 SVM-imposters were used.

5. Conclusions

In this paper we have proposed to use VBA-estimations instead of ML-estimations for UBM creating for GMM speaker verification systems. Thus the problem of the optimal choice of the number of UBM mixture components has been solved and we have avoided two types of problems which appear during UBM learning. The first one is overfitting that appears in ML learning if we excessively increase the number of mixture components on a fixed amount of training data. The second one is noise Gaussians - mixture components which ‘collapse’ onto a specific data point.

We have proposed a fast Gaussian selection criterion based on the values of the mixture coefficients analysis. Using this criterion makes it possible to find the optimal number of components without resorting to resource-intensive methods such as cross-validation in the Bayesian approach. EER was reduced by almost 8% compared to the baseline system that uses the standard ML training for UBM.

We have showed that the traditional UBM with 2048 Gaussians used in systems participating in NIST SRE is less powerful than it could be with the available training data.

6. References


