**UNIVERSITE DE NICE-SOPHIA ANTIPOLIS - UFR SCIENCES**

Ecole Doctorale des Sciences et Technologies de la Communication

THESE

pour obtenir le titre de

Docteur en Sciences

de l'UNIVERSITE de Nice-Sophia Antipolis

Discipline : Automatique Traitement du signal et des images

présentée et soutenue par

AUTEUR ***Fabio VALENTE***

# Variational Bayesian Methods for Audio Indexing

Thèse dirigée par ***Christian WELLEKENS***

soutenue le *(23-09-2005)*

Jury

M. Jean-Paul Haton Professeur Rapporteur

M. Laface Pietro Professeur Rapporteur

M. Wellekens Christian Professeur Directeur de thèse

M. Drygajlo Andrzej Docteur Examinateur

# METHODES BAYESIENNES VARIOTIONELLES pour L' INDEXATIONE AUDIO

23 Septembre 2005

## Abstract

Variational Bayesian (VB) methods are relatively new techniques that allow fully Bayesian learning and fully Bayesian model selection in an approximated fashion. Fully Bayesian learning is impossible in some models that contain hidden variables (e.g. GMM or HMM). Variational methods are based on a bound of the intractable Bayesian integral. Even if those are approximated methods, they benefits of classical Bayesian properties. Thus VB algorithms allow simultaneous model learning and model selection. VB approximation permits the use of an iterative optimization algorithm (the VB Expectation-Maximization) for model learning.

In this work we study the application of VB methods to two audio indexing problems: *speaker clustering* and *speaker change detection*. Both of them are formulated as a model selection problem generally solved using the Bayesian Information Criterion (BIC). The novelty of this thesis consists in reformulating the problem in a fully Bayesian framework handled using the VB method. The VBEM is used for model learning and the variational bound over the Bayesian integral is used for model selection purposes.

Results on Broadcast News data show an interesting improvement respect to classical techniques based on Maximum Likelihood (ML) or Maximum a Posteriori (MAP) for model learning and BIC for model selection.

# Acknowledgments

**Anno I**

*Nel mezzo del cammin di nostra vita*
*mi ritrovai per una selva oscura*
*ché la diritta via era smarrita*
Dante Alighieri, Divina Commedia, Inferno, Canto I

**Anno II**

*Gran duol mi prese al cor quando lo ntesi,*
*per che gente di molto valore*
*conobbi che n quel limbo eran sospesi.*
Dante Alighieri, Divina Commedia, Inferno, Canto IV

**Anno III**

*Lo duca e io per quel cammino ascoso*
*intrammo a ritornar nel chiaro mondo;*
*e sanza cura aver d'alcun riposo,*
*salimmo sù, el primo e io secondo,*
*tanto ch'i' vidi de le cose belle*
*che porta 'l ciel, per un pertugio tondo.*
*E quindi uscimmo a riveder le stelle.*
Dante Alighieri, Divina Commedia, Inferno, Canto XXXIV

First of all, I thank the Good Lord for giving me the strength of passing through all this during last three years.

My next thanks go to my family, my father Giovanni, my mother Anna Maria and my brother Giancarlo for their unconditional support during good and bad moments i had in my life.

Naturally this work would have not been possible without the help, the support and the *infinite patience* of my thesis advisor, Professor Christian Wellekens.

I also would like to thank my thesis board for their useful comments: professor Jean-Paul Haton, professor Pietro Laface and Doctor Andrzej Drygajlo.

During my stay at the Eurecom Institute, I came across some very special people that I consider friends rather than colleagues: Vivek, Joackim, Lotta, Raphael, Carla and Melek. Thank you guys for everything!

The Italian crowd in south of France made me fell closer to home during those years: Federico, Luca, Pietro, Stefano, Pierluigi, Claudio.

I would like to thank the Turbo-HMM virtuoso Florent for all the time spent discussing the complicated world of machine learning and Hicham for all the time spent discussing in the smoking area of the Institute.

The next thanks go to the staff in Eurecom for their help with admistrative problems.

A special thank as well to the guys of the B&B of Via Volturno in Rome for their precious logistical support.

Finally i would like to thank to all the *maître d' arme* of the fencing club in Antibes for making the last year more pleasant and for introducing me to the art of fencing.

*Nunc bibendum est.*

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In the 2004 special edition of the *MIT's magazine of innovation and technology review* ([117]) a large article is dedicated to the "10 emerging technologies that will change your world". In between the most important breakthrough in term of innovation a special chapter is dedicated to Bayesian Machine Learning.

Bayesian methods are based on an intuition from reverend Thomas Bayes, an 18th century minister. The idea of what is known today as the Bayes law was described in "Essay Towards Solving a Problem in the Doctrine of Chances" (1763), published posthumously in the Philosophical Transactions of the Royal Society of London ([17]). Since than the work based on the initial idea has significantly progressed and Bayesian methods often combined with graph theory provides extremely powerful tools for many machine learning problems. Methods that use graphs together with Bayesian probability theory are also known as Bayesian Networks (BN) or Graphical Models (GM) and are now a very popular tool because of their complete generality (for a general review on BN see [72]).

Fields of application for Bayesian methods are now extremely various and heterogeneous; in between them it is possible to find language processing, microchip manufacturing, drug discovery, biology, genetic, robot navigation , data mining and many others.

Bayesian methods have the appealing property of studying dependence in between different stochastic variables using the Bayes rule between prior-posterior distributions and conditional probabilities. Furthermore Bayesian framework benefits from the interesting property of Occam's razor useful in model selection problems.

Occam's razor is a philosophical principle attributed to the 14th century logician and Franciscan friar William of Occam. The principle states that "Entities should not be multiplied unnecessarily." Sometimes it is quoted in one of its original Latin forms to give it an air of authenticity: "Pluralitas non est ponenda sine neccesitate". William used the principle to justify many conclusions including the statement that "God's existence cannot be deduced by reason alone." That one didn't make him very popular with the Pope.

Occam's razor principle is a very popular idea in science and it is widely used in statistic and probabilistic inference as well. In probabilistic model framework it can be interpreted as the choice of the simplest model in between the different possible models that can explain a given phenomenon.

The connection with Bayesian methods in model learning is extremely surprising: Bayesian model learning embeds a form of model penalty that penalizes more complex models versus to simpler models. In other words Bayesian methods prefer simpler models for explaining a given phenomena. This makes the Bayesian framework even more appealing in many machine learning problems that require model selection.

Anyway in many real data applications fully Bayesian inference is impossible in an exact way. This is sometimes a direct consequence of the extremely complicated and rich models that Bayesian theory can provide. In those cases approximated methods must be considered in order to produce a solution as close as possible to the real one.

In this work we study a family of Bayesian approximated methods known as *Variational Bayesian methods* that directly approximate the Bayesian integral. Final goal of this thesis is studying the applicability of those methods to audio file processing that needs probabilistic model selection. In the experimental part we deal with audio indexing problems. The audio indexing problem is currently formulated as statistical model selection problem. In fact when no a priori information is available about the nature of the data, a statistical model is generally build in order to represent the audio stream. The structure of the audio file is inferred from the complexity of the model e.g. the speaker number. Here comes the need for model selection methods that choose the best model. In state-of-art indexing systems the model selection criterion is generally a extremely rough approximation of the Bayesian integral. Here we propose for the first time the use of Variational Bayesian methods for speaker indexing purposes.

This thesis is organized as follows:

Chapter 2 describes some very classical concept of Bayesian machine learning like Maximum Likelihood (ML) and Maximum a Posteriori (MAP) and applications to Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM). Expectation-Maximization (EM) algorithm for ML and MAP learning is considered as well. Finally Bayesian model selection and prior distribution problem are discussed and classical approximation like the BIC are considered. Goal of chapter 2 is basically introducing notation and basic concepts that will be later compared with Variational Bayesian solution to the same problems.

Chapter 3 is the theorical heart of the thesis. Variational Bayesian methods are here introduced. An EM-like algorithm for approximated inference is here described and discussed for Conjugate-Exponential models. Approximated variational learning is also considered for the ML and MAP criterion. In the last part of the chapter model selection properties of VB framework are considered. Finally a short description of VB methods in machine learning and speech processing are considered.

Chapter 4 considers application of VB methods to two main models in audio processing: GMM and HMM. For both of them we provide details about derivation of the variational free energy, model selection, learning using the EM-like algorithm, inference on unseen data and empirical prior derivation. HMM with emission probability modeled as GMM are also considered.

Chapter 5 is an experimental chapter in which we compare classical ML and MAP learning with the VB learning in a GMM framework. Three factors are considered: amount of training data, prior distributions and model complexity (i.e. initial component number). Different experimental scenarii are considered. The adaptation task starting from a Universal Background Model using both MAP and VB is studied as well. Basically theorical intuitions are experimentally verified on synthetic and real data.

Chapter 6 deals with speaker detection change problem; a review of current state-of-the-art solution based on the Bayesian Information Criterion (BIC) or on the Log-Likelihood Ratio (LLR) is proposed. The VB solution to this problem is then introduced. Even if VB solution is sensitive to prior distributions, it outperforms the classical BIC framework.

Chapter 7 deals with speaker clustering problem. A review of classical BIC based solution is proposed. VB framework is successfully applied to the considered model. Furthermore an extensive analysis on the impact of prior distributions is done considering flat, strong and heuristic priors.

# Chapter 2

# Generalities

The goal of machine learning is the construction of a model that reflects the nature of some experimental data. The model generally is build on an hypothesis of the natural phenomenon that generated the data. The goal of the modeling process can be very different; for example the model can be required to discover some structures in the data or can be used to make prediction on some unseen data. The model can be a parametric model or a non-parametric. In a parametric model, modeling is achieved by using a parameter set that we will refer to in this work as $\theta$ while in non-parametric models there is no explicit attempt to identify any parameter. In this thesis we will focus basically on parametric models. Algorithms used to determine the best parameter set for a given task are called *training* or *learning* algorithms. Two main classes of training methods can be considered: the *generative learning* and the *discriminative learning*. In the first case, learning algorithm tries to build a model that reflects as much as possible data characteristics. Model "capability" to reflect data is generally measured in term of probability. Mathematically speaking if we denote with $Y$ the observation set and with $\theta$ model parameters that define a probability $p(Y|\theta)$ over the model, the goal of generative modeling is finding $\hat{\theta}$ such that:

$$\hat{\theta} = argmax_{\theta}\, p(Y|\theta). \tag{2.1}$$

This criterion is also known as the *maximum likelihood* criterion because it maximizes the likelihood of the data given parameters i.e. $p(Y|\theta)$. On the other hand *discriminative learning* considers a set of $m$ models with $m = \{1, \ldots, M\}$ with respectively their parameter set $\theta_m$ and labeled data set $Y_m$. The goal of *discriminative learning* is to train model parameters $\theta_m$ in order to classify a test data set as belonging to one of those models. Clearly the task is very different from the generative learning because the quality of the inferred models does not matter but just their discriminative capability is considered. Different hybrid generative/discriminative techniques have been developed (for review see [62]).

One of the main interests of learning a model is the possibility of making prediction on an unseen data set i.e. given an unseen data set $U$ computing $p(U|\bar{\theta})$ where $\bar{\theta}$ is the estimated parameter set. Prediction quality will be largely influenced by the simplification assumed in the modeling part.

In many real data problems the assumption that data are generated by some unobserved variables is done. Those variables are generally referred to as *latent* or *hidden* variables. If we denote with $X$ the hidden variables set, and $Z = \{X, Y\}$ the complete data set, it is possible to write the joint hidden and observed probability given model parameters as:

$$p(Z|\theta) = p(Y, X|\theta) = p(Y|X, \theta)p(X|\theta). \tag{2.2}$$

where $p(X|\theta)$ is the hidden variable probability conditioned to parameter $\theta$. In other words observed data probabilities are conditioned on some unseen variables. In order to find data evidence it is enough to marginalize w.r.t. the hidden variables for a given parameter set $\theta$ i.e.

$$p(Y|\theta) = \sum_X p(Y|X, \theta)p(X|\theta). \tag{2.3}$$

2.3 is thus known as incomplete-data likelihood.

Previously we have considered a probability distribution over hidden variables but in a fully Bayesian framework, a probability distribution over all model elements should be considered. More specifically model parameters too can be considered as random variables with their probability distributions that we will designate with $p(\theta)$. As before they can be marginalized as:

$$p(Y|m) = \int d\theta\, p(\theta|m) \sum_X p(Y|X, \theta, m)\, p(X|\theta, m). \tag{2.4}$$

where $m$ is the considered model and all probabilities are now considered conditioned to the given model. Expression (2.4) is known as *marginal likelihood* and it is a key quantity for many important tasks like model selection. Advantages of fully Bayesian methods is that model parameters are not directly estimated but the interest is shifted over parameter distributions. This is an extremely suitable property because models are often a simplification of the data reality and training a given model could generate many problems e.g. overfitting and poor generalization. Bayesian methods do not estimate the model but probabilities over all possible models with the big advantage of avoiding or at least alleviating many parameter estimation problems.

In the same fashion as before we could also consider a probability over a given model $m$ denoted by$p(m)$. In this case, data evidence can be obtained by marginalizing over all possible models i.e.

$$p(Y) = \sum_m p(m)p(Y|m). \tag{2.5}$$

where $p(Y|m)$ is defined as in (2.4). It is important to notice that we have started our discussion with a parametric model with explicit parameters and hidden variables and we ended up to quantity (2.5), where all possible terms have been integrated out and only data evidence is considered.

## 2.1 Maximum Likelihood Parameters Learning

In this section we explicitly consider the learning algorithm for generative modeling giving some classical examples. The first very simple example we can consider is the learning in case of a single Gaussian with parameters means $\mu$ and variance $\sigma$ in the scalar case. If the observation set is $Y = \{y_1, \ldots, y_N\}$ and observations are considered independent the maximum likelihood estimation for Gaussian parameters is trivially $\mu = \sum_i y_i/N$ and $\sigma = \sum_i (y_i - \mu)^2/N$. This solution can be found simply deriving the data log-likelihood w.r.t. parameters and solving.

Unfortunately things are not always that simple. Let us now consider the incomplete data case; because of the fact a part of the variables are hidden there is no close form optimization for parameters $\theta$ in quantity (2.3) and an iterative algorithm known as *Expectation-Maximization* (EM) is generally used.

## 2.2 The Expectation-Maximization (EM) algorithm

The Expectation-Maximization (EM) algorithm introduced by [39] is an iterative algorithm used to learn parameters of models that contain hidden variables for which a closed form optimization cannot be possible. Iterations consist in an E-step in which given an initial parameter estimation $\bar{\theta}$, hidden variables probabilities $p(X = x|\bar{\theta})$ are estimated ($x$ represent a particular value of $X$) and an M-step in which the maximum likelihood solution for $\bar{\theta}$ is obtained given hidden variable values estimated in the E-step.

In more details, it is possible to factorize probability of $Z = \{X, Y\}$ as:

$$p(Z|\bar{\theta}) = p(Y, X|\bar{\theta}) = p(X|Y, \bar{\theta})\, p(Y|\bar{\theta}) \tag{2.6}$$

As long as we want to maximize $p(Y|\bar{\theta})$, let us consider expression (2.6) in term of logarithms:

$$log\, p(Y|\bar{\theta}) = log\, p(Y, X|\bar{\theta}) - log\, p(X|Y, \bar{\theta}) \tag{2.7}$$

Taking now the expectation of both sides of equation (2.7) w.r.t. probability of hidden variable X computed with parameters $\theta$ we obtain:

$$log\, p(Y|\bar{\theta}) =< log\, p(X, Y|\bar{\theta}) >_{X|\theta} - < log\, p(X|Y, \bar{\theta}) >_{X|\theta}= Q(\theta, \bar{\theta}) - H(\theta, \bar{\theta}) \tag{2.8}$$

where we denote with $< t >_{X|\theta}$ the expectation of function $t$ over hidden variables $X$ estimated using parameter $\theta$. Let us explicit $Q(\theta, \bar{\theta})$ and $H(\theta, \bar{\theta})$ as:

$$Q(\theta, \bar{\theta}) =< log\, p(X, Y|\bar{\theta}) >_{X|\theta}= \sum_X p(X|Y, \theta)\, log\, p(X, Y|\bar{\theta}) \tag{2.9}$$

$$H(\theta, \bar{\theta}) =< log\, p(X|Y, \bar{\theta}) >_{X|\theta}= \sum_X p(X|Y, \theta)\, logp(X|Y, \bar{\theta}) \tag{2.10}$$

The key point of the EM algorithm is to choose a $\bar{\theta}$ in order to not decrease Q(.) because

$$Q(\theta, \bar{\theta}) \geq Q(\theta, \theta) \Rightarrow log\, p(Y|\bar{\theta}) \geq log\, p(Y|\theta). \tag{2.11}$$

since from Jensen inequality [66] $H(\theta, \bar{\theta}) \leq H(\theta, \theta)$. Function $Q(\theta, \bar{\theta})$ is also known as Q-function or auxiliary function. In order to verify condition (2.11), it is enough to maximize the Q-function; in this way the log-likelihood will monotonically increase and converge in a local maximum of the Q function.

So given a parameter initialization $\theta$, the EM algorithm consists in iteratively computing the auxiliary function $Q(\theta, \bar{\theta})$ (E-step) and then estimating parameters $\bar{\theta}$ that maximize Q (M-step). The M-step is actually a maximum likelihood estimation with complete data because hidden variables were estimated in the E step. Then $\theta$ is replaced by $\bar{\theta}$ and the EM steps are iterated until convergence of the objective function.

EM algorithm can be applied to a huge set of different models that use hidden variables like Gaussian Mixture models or Hidden Markov Models. Anyway the possibility of applying the EM algorithm is sometimes limited by the possibility of computing the Q-function i.e. the possibility of an estimation of the hidden variable set; we will see later that in some cases of EM inapplicability, an approximated objective function can be obtained for bounding the objective function and obtaining a tractable algorithm. In the next two sections we will detail two cases of EM/ML estimation very frequently used in speech processing: GMM and HMM.

### 2.2.1 Maximum Likelihood Gaussian Mixture Models

Gaussian mixture model can be seen as a generalization of Vector Quantization ([88]). GMM makes the hypothesis that data $Y$ can be modeled by a mixture of Gaussian density function of the form:

$$p(Y) = \sum_{i=1}^{M} c_i\, N(Y_i|\mu_i, \Sigma_i) \tag{2.12}$$

where model parameter set $\theta$ is constituted by weights $c_i$, means $\mu_i$ and covariance $\Sigma_i$ with $0 \leq c_i \leq 1$ and $\sum_{i=0}^{M} c_i = 1$ where $M$ is the mixture number. Mixture density is a typical case of EM estimation [58]. Observed data are constituted by $Y = \{y_1, \dots, y_t\}$ and missing variables $X = \{x_1, \dots, x_t\}$ assume a value in $\{1, \dots, M\}$ that designates which Gaussian component out of the $M$ has generated the observed data. It is useful to observe that the number of missing variables is equal to the number of data $t$. Supposing that elements of $Y$ are independent sample, it is straightforward to write the auxiliary Q-function:

$$Q(\theta, \bar{\theta}) = \sum_{t=1}^{T} \sum_{x_t=1} p(x_t | y_t, \theta) \, log \, p(y_t, x_t | \bar{\theta}) = \sum_{t=1}^{T} \sum_{x_t=1} \frac{p(y_t, x_t | \theta)}{p(y_t | \theta)} \, log \, p(y_t, x_t | \bar{\theta}) \tag{2.13}$$

Using 2.12 and manipulating 2.13, it is possible to obtain:

$$Q(\theta, \bar{\theta}) = \sum_{i=1}^{M} \gamma_i \, log \, \bar{c}_i + \sum_{i=1}^{M} Q_i(\theta, \bar{\theta}) \tag{2.14}$$

where

$$\gamma_i^t = \frac{c_i p(y_t | \mu_i, \Sigma_i)}{p(y_t | \theta)} \tag{2.15}$$

$$\gamma_i = \sum_{t}^{T} \gamma_i^t \tag{2.16}$$

$$Q_i(\theta, \bar{\theta}) = \sum_{t=1}^{T} \gamma_i^t \, log \, p(y_t | \mu_i, \Sigma_i) \tag{2.17}$$

Now that the auxiliary function is computed, it is enough to maximize it w.r.t. parameters $\theta$ in order to find a parameter estimate. Deriving w.r.t. $c_i, \mu_i, \Sigma_i$ and solving we obtain (for details see [21]):

$$\bar{c}_i = \frac{\gamma_i}{\sum_i \gamma_i} = \frac{\gamma_i}{T} \tag{2.18}$$

$$\bar{\mu}_i = \frac{\sum_{t=1}^{T} \gamma_i^t \, y_t}{\sum_{t=1}^{T} \gamma_i^t} \tag{2.19}$$

$$\bar{\Sigma}_i = \frac{\sum_{t=1}^{T} \gamma_i^t \, (y_t - \bar{\mu}_i)(y_t - \bar{\mu}_i)^T}{\sum_{t=1}^{T} \gamma_i^t} \tag{2.20}$$

It must pointed out that parameter optimization is a constrained optimization that must satisfy the condition $\sum_{i=0}^{M} c_i = 1$; constrained optimization can be achieved using Lagrange multipliers method. Iteratively applying equations (2.15) (for the E-step) and (2.18), (2.19) and (2.20) (for the M-step) parameters converge into a local maxima of the objective function.

Two issues must be pointed out. First, all EM methods converge to a local maxima that depends on the initialization i.e. given the same training data set and the same model topology two different initializations may yield two different estimations.

On the other side GMM/ML learning may yield a singular solution. In fact let us consider the case in which a single vector is "assigned" to a given Gaussian component, the covariance matrix becomes singular. This is a very unsuitable problem that take place when the initial model is not well chosen or not well initialized (e.g. very few data and too many components). Generally during EM learning the covariance matrix determinant is checked at each iteration and different strategies are possible when a singularity occurs (see [98]). Again a way to avoid this kind of problem is to carefully select the model. We will see later that singular solutions are not a problem in Bayesian learning.

### 2.2.2 Maximum Likelihood for Hidden Markov Models

In this section we consider the most popular model used in many speech processing applications: Hidden Markov Model (HMM). HMM was first introduced in speech processing independently by [64] and [65]. It consists in a probabilistic model for a collection of observed data $Y = \{y_1, \dots, y_t, \dots, y_T\}$ and hidden variables $S = \{s_1, \dots, s_t, \dots, s_T\}$ that are discrete variables that designate the "state" of the HMM and can take $N$ values where $N$ is the state space dimension. Two hypothesis define an HMM:

- The hidden variable $s_t$ is dependent only on $s_{t-1}$ i.e. $p(s_t|s_{t-1}, \dots, s_1, Y) = p(s_t|s_{t-1})$

- The observation $y_t$ dependent only on the hidden variable $s_t$ i.e. $p(y_t|y_{t-1}, \dots, y_1, s_{t-1}, \dots, s_t) = p(y_t|s_t)$

Another assumption typically used is that the transition probability $p(s_t|s_{t-1})$ is time-homogeneous i.e. independent of time; in this way transition probability can be represented as a time-independent transition matrix $A = a_{ij} = p(s_t = j|s_{t-1} = i)$. The case of $t = 1$ is represented in the initial state distribution $\pi_i = p(s_1 = i)$. Emission probability of a vector $p(y_t|s_t = i) = b_i(y_t)$ can be discrete or continuous (generally a Gaussian distribution or a GMM). To summarize the HMM parameter set $\theta$ is represented by $\pi, A, B$. The classical approach to HMM consider three main problems (see [109]):

- Given an observation sequence $Y$ and a parameter set $\theta$ find $p(Y|\theta)$.

- Given an observation sequence $Y$ and a parameter set $\theta$ find the best state sequence $S$ that generated observation sequence.

- Given an observation sequence $Y$ find the maximum likelihood estimate for $\theta$.

In this section we will review in a synthetic way those three problems in order to compare them with the variational solution to the same questions.

Concerning the *first problem* the brute force solution is marginalizing over all possible path $S$ i.e.

$$P(Y|\theta) = \sum_S P(Y|\theta, S)P(S|\theta) \tag{2.21}$$

but when the state space is large, marginalization may be impracticable. A practical solution is using the *forward algorithm* that benefits from the Markov hypothesis i.e. computation of $p(s_t|s_{t-1})p(y_t|s_t, \theta)$ involves only $s_t, s_{t-1}, y_t$ so that it is possible to write the likelihood as a recursion over $t$. Let us define the forward probability as:

$$\alpha_t(i) = p(Y_1^t, s_t = i|\theta) \tag{2.22}$$

The forward algorithm can be represented as a 3 step algorithm with an initialization:

$$\alpha_1(i) = \pi_i b_i(y_1) \; with \, 1 \leq i \leq N \tag{2.23}$$

Then, an induction step in which the recursion is computed:

$$\alpha_t(j) = [\sum_{i=1}^{N} \alpha_{t-1}(i)a_{ij}]b_j(y_t) \; with \; 2 \leq t \leq T; \; 1 \leq j \leq N \tag{2.24}$$

and finally the termination step gives the likelihood:

$$P(Y|\theta) = \sum_{i=1}^{N} \alpha_T(i) \tag{2.25}$$

It is easy to check that the complexity of the forward algorithm is simply $O(N^2T)$ instead of exponential with $N$.

Solution to *second problem* i.e. finding the best path or state sequence can be solved using the Viterbi algorithm [136]. The Viterbi algorithm is a greedy solution to an exhaustive search on the state space $S$ and is based on the same principle of the forward algorithm. Instead of summing all possible contributions as in (2.24), the algorithm chooses the highest contribution to $\alpha_t$ and "remembers" the best path. At the end of the recursion the best path is computed backtracking the optimal local choices. The Viterbi algorithm complexity is similar to the forward algorithm complexity i.e. $O(N^2T)$.

The problem of parameter learning (i.e. the *third problem*) is another application of the EM algorithm that in the case of HMM take the name of Baum-Welch algorithm ([16] and [15]). It is based on the use of a forward-backward recursion as a trick for estimating the HMM sufficient statistics. It is possible to define a backward algorithm analogous to the forward, that starts the recursion from the end of the sequence. Let us define the backward probability:

$$\beta_t(i) = p(Y_{t+1}^T | s_t = i, \theta) \tag{2.26}$$

the backward algorithm compute recursively $\beta_t$ in this way:

$$\begin{aligned} \beta_T(i) &= 1/N \; with \; 1 \leq i \leq N & (2.27) \\ \beta_t(i) &= [\sum_{j=1}^{N} a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)] \; t = T-1, ..., 1; \; 1 \leq i \leq N & (2.28) \end{aligned}$$

The interest of the backward-forward recursion is in the possibility of estimating HMM key quantity $\gamma_t(i,j)$ (actually the HMM sufficient statistics) defined as:

$$\begin{aligned} \gamma_t(i,j) &= p(s_{t-1} = i, s_t = j | Y_1^T, \theta) = \frac{p(s_{t-1} = i, s_t = j, Y_1^T \theta)}{p(Y_1^T | \theta)} \\ &= \frac{\alpha_{t-1} a_{ij} b_j(y_t) \beta_t(j)}{\sum_{k=1}^{N} \alpha_T(k)} & (2.29) \end{aligned}$$

The importance of quantities $\gamma_t(i,j)$ can be understood considering the EM in terms of auxiliary function $Q$:

$$Q(\theta, \bar{\theta}) = \sum_s \frac{p(Y, s|\theta)}{p(Y|\theta)} log \, p(Y, s|\bar{\theta}) \tag{2.30}$$

Let us explicit elements in 2.30, it is possible to write:

$$\begin{aligned} Q(\theta, \bar{\theta}) &= Q_{\pi_i}(\theta, \bar{\pi}_i) + Q_{a_i}(\theta, \bar{a}_i) + Q_{b_j}(\theta, \bar{b}_j) & (2.31) \\ Q_{\pi_i}(\theta, \bar{\pi}_i) &= \sum_i \frac{P(Y, s_0 = j|\theta)}{P(Y|\theta)} \, log \, \bar{\pi}_i & (2.32) \\ Q_{a_i}(\theta, \bar{a}_i) &= \sum_i \sum_j \sum_t \frac{p(Y, s_{t-1} = i, s_t = j|\theta)}{p(Y|\theta)} \, log \, \bar{a}_{ij} & (2.33) \\ Q_{b_j}(\theta, \bar{b}_j) &= \sum_j \sum_t \sum_{t \epsilon y_t = o_k} \frac{P(Y, s_t = j|\theta)}{P(Y|\theta)} \, log \, \bar{b}_j(k) & (2.34) \end{aligned}$$

Optimization of $Q$ can now be done separately for the two independent term according to constraints on $a_{ij}$ and $b_j(k)$. Constrained optimization is generally achieved by using the Lagrange multipliers:

$$\begin{aligned}
\bar{\pi}_i &= \sum_j^N \gamma_0(i,j) && (2.35)\\
\bar{a}_{ij} &= \frac{\sum_{t=1}^T \gamma_t(i,j)}{\sum_{t=1}^T \sum_{k=1}^N \gamma_t(i,k)} && (2.36)\\
\bar{b}_j(k) &= \frac{\sum_{t \epsilon y_t = o_k} \sum_i \gamma_t(i,j)}{\sum_{t=1}^T \sum_i \gamma_t(i,j)} && (2.37)
\end{aligned}$$

Instead of a discrete distribution a continuous distribution for the emission probability can be considered.

A Gaussian mixture model can be also used. Assuming that the emission probability of state $j$ is modeled with a $M$ component GMM with parameters $c_{jk}, \mu_{jk}, \sigma_{jk}$ estimation formula becomes:

$$\begin{aligned}
\bar{c}_{jk} &= \frac{\sum_{t=1}^T \xi_t(j,k)}{\sum_{t=1}^T \sum_{k=1}^M \xi_t(j,k)} && (2.38)\\
\bar{\mu}_{jk} &= \frac{\sum_{t=1}^T \xi_t(i,j) y_t}{\sum_{t=1}^T \sum_{k=1}^M \xi_t(j,k)} && (2.39)\\
\bar{\Sigma}_{jk} &= \frac{\sum_{t=1}^T \xi_t(i,j)(y_t - \bar{\mu}_{jk})(y_t - \bar{\mu}_{jk})^T}{\sum_{t=1}^T \sum_{k=1}^M \xi_t(j,k)} && (2.40)
\end{aligned}$$

where

$$\xi_t(i,k) = \frac{P(Y, s_t = j, k_t = k|\theta)}{P(Y|\theta)} = \frac{\sum_{i=1}^M \alpha_{t-1}(i) a_{ij} \beta_{jk} b(jk)(y_t) \beta_t(j)}{\sum_{i=1}^N \alpha_T(i)} \qquad (2.41)$$

We have considered here the case of learning with just one observation sequence but the algorithm can be easily extended to the case of multiple observations.

We will see later problems that come from the use of a fully Bayesian framework.

## 2.3 Bayesian Learning

One of the most popular probability law is Bayes law that given two stochastic variables $a, b$ states:

$$p(a|b) = \frac{p(b|a)p(a)}{p(b)} \qquad (2.42)$$

A possible interpretation of equation (2.42) is that variable $p(a)$ can be seen as the *prior information about variable* $a$, $p(a|b)$ can be seen as the *posterior information on* $a$ *once* $b$ *is known* and $p(b|a)$ can be seen the “amount” of information that the knowledge of $b$ gives to $a$. In Bayesian learning each parameter is considered as a stochastic variable with its own distribution. So considering again notation of previous sections, we designate with $Y$ observations and with $\theta$ parameters, in this case the Bayes rule gives:

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)} \tag{2.43}$$

so that $p(\theta)$ is parameter prior, $p(\theta|Y)$ is parameter posterior after considering data $Y$ and $p(Y|\theta)$ is additional knowledge brought to $\theta$ by $Y$. In the same fashion, considering a probability over the model $m$ gives:

$$p(m|Y) = \frac{p(Y|m)p(m)}{p(Y)} \tag{2.44}$$

where $p(Y|m)$ can be computed marginalizing over all parameters i.e.

$$p(Y|m) = \int d\theta\, p(Y|\theta, m)p(\theta|m) \tag{2.45}$$

Expression (2.45) is again a marginal likelihood as in expression (2.4) (without considering any hidden variable this time).

It is clear that the key of any Bayesian approach is the use of the prior distribution i.e. for instance $p(\theta|m)$ in expression (2.45). Actually many kind of solutions for setting prior distributions are possible. We will consider them in the same four main class as in [18],[147]: objective, subjective, hierarchical, and empirical.

### 2.3.1 Objective priors

Objective priors try to incorporate as few a priori information as possible into the model. This is a useful property when there is no knowledge at all concerning the current task. Anyway it turns out that this modeling is extremely difficult and often objective priors result into extremely complicated form or unnormalized function as improper priors. Those kinds of priors are also known as *non-informative* priors in contrast with the *informative* priors that add some knowledge about the models.

For example translation invariant prior can be written as $p(\theta) = const$ and scale invariant prior $p(\theta) = \frac{const}{\theta}$. Both priors are improper i.e.

$$\int p(\theta)d\theta = +\infty \tag{2.46}$$

As long as improper priors are not used in this work and we will limit our discussion to proper prior into the conjugate-exponential family and those kinds of priors will not be consider more here.

### 2.3.2 Subjective priors

Subjective priors contain previous knowledge or previous hypothesis on a certain model. Generally subjective priors are chosen of a form that gives a tractable analytical form for the problem. A serious drawback is that generally subjective priors are far away from the real prior distribution of data. The most popular class of subjective priors is the *conjugate* prior in the *exponential family* (e.g. see [40]). The main advantage of this class of distribution is the tractability; in fact the posterior distribution will have the same analytical form as the prior distribution but with "updated" parameters. Mathematically speaking, if we consider a prior distribution over parameters $p(\theta|\lambda)$ where $\theta$ are model parameters and $\lambda$ are distribution parameters (a.k.a. *hyperparameters*), then we will have:

$$p(\theta|\bar{\lambda}) = p(\theta|Y) \propto p(Y|\theta) \times p(\theta|\lambda) \tag{2.47}$$

where $\bar{\lambda}$ are posterior distribution *hyperparameters*. The general form for a function belonging to the *exponential family* is :

$$p(y|\theta) = g(\theta)f(y)\, e^{\phi(\theta)^T u(y)} \tag{2.48}$$

where $g(\theta)$ is a normalizing constant. $\phi(\theta)$ is a vector of *natural parameters* and $u(y)$ and $f(y)$ are functions that define the family. The conjugate prior has the following form:

$$p(\theta|\eta,\nu) = h(\eta,\nu)\, g(\theta)^{\eta}\, e^{\phi(\theta)^T \nu} \tag{2.49}$$

where $\nu$ and $\eta$ are prior parameters (hyperparameters as before) and $h(\eta,\nu)$ is a normalization constant. The trick is considering the same functions $g(\theta)$ and $\phi(\theta)$ in (2.48) and (2.49) in this way it is possible for a data set $Y = \{y_1, \dots, y_n\}$ to write the posterior distribution as:

$$p(\theta|Y) \propto p(\theta|\bar{\eta},\bar{\nu}) = p(\theta|\eta,\nu)p(Y|\theta) \tag{2.50}$$

where $\bar{\eta} = \eta + n$ and $\bar{\nu} = \nu + \sum_{i=1}^{n} u(y_i)$ are "updated" hyperparameters. This is a very suitable property in term of tractability: to compute parameter posterior distributions, only hyperparameters must be modified and the distribution will keep the same form. A very intuitive explanation can be found to this phenomenon: prior hyperparameters $\nu, \eta$ are the information that comes from the initial knowledge and they are updated using contribution from data. If the data contribution is void, then the posterior will coincide with the prior.

Another extremely interesting point is that posterior distribution can be used again as prior distributions for another model still keeping the conjugate form.

Unfortunately when hidden variables are used, it is impossible to find a conjugate form for prior distributions, in those cases approximated methods are used.

### 2.3.3 Empirical Bayes priors

In previous sections we have seen that prior distributions have their own parameters generally known as *hyperparameters*. The empirical Bayesian approach considers hyperparameters as another kind of parameters to be optimized from the data. Mathematically speaking model parameters are marginalized obtaining:

$$p(Y|\lambda) = \int d\theta \, p(Y|\theta)p(\theta|\lambda) \tag{2.51}$$

and now evidence $p(Y|\lambda)$ can be used to give a maximum likelihood estimation for hyperparameters $\lambda$. The main advantage of this approach is that it overcomes problem given by the mis-setting of $\lambda$ but on the other hand is subject to the same overfitting problem of maximum likelihood estimation or other frequentist approaches. For example this is the approach followed in [93].

### 2.3.4 Hierarchical priors

In hierarchical priors, hyperparameters themselves have their own prior distributions regulated by their hyperparameters and so on. So prior distributions on parameters $\theta$ are $p(\theta|\theta_1)$, prior distribution on hyperparameters are $p(\theta_1|\theta_2)$ and more generally $p(\theta_{n-1}|\theta_n)$. It can be shown that it is equivalent to using a proper distribution $p(\theta)$ given by:

$$p(\theta) = \int p(\theta|\theta_1)p(\theta_1|\theta_2)\ldots\, p(\theta_{n-1}|\theta_n)\, d\theta_1 \,\ldots\, d\theta_n \tag{2.52}$$

Even if it seems that in theory nothing is gained using such a type of posterior in practice, it is a useful tool for building robust priors. Theorically the hierarchy can continue up to an infinite number of levels: this kind of models have been studied in [19] and [110].

## 2.4 Tractable approximation and MAP

In previous sections we have shown that fully Bayesian inference must consider all contributions to the Bayesian integral i.e. we have to consider quantities (2.45) or (2.4) in the case of hidden variables. Anyway in real data problems with hidden variables, fully Bayesian inference is an extremely difficult problem. Generally a closed form solution is not possible and two possibilities are open: computing the integral with some numerical methods as Monte Carlo methods or using some approximation of the integral.

In this section we will consider the simplest approximation: ignoring the integral and reducing the estimation to a point estimation i.e. find parameters $\bar{\theta}$ such that:

$$\bar{\theta} = argmax_{\bar{\theta}} p(\theta)p(Y|\theta) \tag{2.53}$$

This criterion is also known as the *Maximum a Posteriori* criterion. Using a metaphor coming from physics, MAP considers just probability density instead of the mass. In other words all the mass of the distribution is concentrated in a point (the MAP solution).

The advantage of using a criterion like (2.53) is in the tractability; in fact when optimization involves hidden variables the EM algorithm ([39]) can also be applied to the MAP parameter estimation with a simple modification of the auxiliary function (see section 2.4.1). On the other hand a serious problem is given by the non-unicity of the solution. In fact MAP approach is not parameterization invariant; two models with identical priors and likelihood functions will give different MAP estimates depending on their parameterization. In the next section we will discuss EM for MAP while in section 2.5 we will consider into detail the reparameterization problem.

### 2.4.1 EM for MAP estimation

A simple modification of the $Q$ function can be used to give the MAP parameter estimation. In fact adding the contribution from prior distributions to the auxiliary function still gives an auxiliary function (see [46]):

$$Q_{MAP}(\theta, \bar{\theta}) = log(p(\bar{\theta})) + Q(\theta, \bar{\theta}) \tag{2.54}$$

EM algorithm can be directly applied to the auxiliary function (2.54) for obtaining the MAP parameter estimate. The Expectation step will have a form similar to the one for the ML estimate because $log(p(\bar{\theta}))$ in (2.54) does not contain hidden variables; the M-step will have to deal with prior distributions. In next section we will consider the case of the GMM and HMM with MAP learning.

### 2.4.2 MAP estimate for GMM

In this section we will derive the MAP estimation formulae for a Gaussian mixture model with same parameters as in section 2.2.1.

The very first important point is defining prior distribution for the model parameters. According to the discussion hold in section 2.3.2 a strong tractability gain can be obtained using prior belonging to the exponential conjugate family. In GMM the choice is a very classical one and consists in using a Dirichlet distribution as prior distribution over weights, a Normal-Wishart distribution as joint distribution over means and covariance matrix assumed independent of $\{c_i\}$ i.e.:

$$\begin{aligned} p(\{c_i\}) &= Dir(\lambda) && (2.55)\\ p(\mu_i|\Sigma_i) &= N(\rho_i, \xi_i\Sigma_i) && (2.56)\\ p(\Sigma_i) &= W(a_i, B_i) && (2.57)\\ p(\theta) &= p(\{c_i\})\prod_i p(\mu_i|\Sigma_i)p(\Sigma_i) && (2.58) \end{aligned}$$

where $Dir()$, $N()$ and $W()$ designate a Dirichlet, Normal and Wishart distribution and $\lambda_i, \rho_i, \xi_i, a_i, B_i$ are distribution hyperparameters. Imposing such prior distributions we are sure that posterior distributions have the same form as prior with augmented hyperparameters.

The E-step will have the same form as the E-step for the ML estimate i.e. equations 2.15.

$$\begin{aligned} \gamma_i^t &= \frac{c_i p(y_t|\mu_i, \Sigma_i)}{p(y_t|\theta)} && (2.59)\\ \gamma_i &= \sum_t^T \gamma_i^t && (2.60)\\ \bar{\omega}_i &= \frac{\sum_{t=1}^T \gamma_i^t\, y_t}{\sum_{t=1}^T \gamma_i^t} && (2.61)\\ \bar{r}_i &= \sum_{t=1}^T \gamma_i^t\,(y_t - \bar{\mu}_i)(y_t - \bar{\mu}_i)^T && (2.62) \end{aligned}$$

On the other side for the M-step optimal posterior distributions must be derived. Posterior distributions will have the same form as priors i.e.

$$\begin{aligned} p(\{\bar{c}_i\}) &= Dir(\{\bar{\lambda}_i\}) && (2.63)\\ p(\bar{\mu}_i|\bar{\Sigma}_i) &= N(\bar{\rho}_i, \bar{\xi}_i\bar{\Sigma}_i) && (2.64)\\ p(\bar{\Sigma}_i) &= W(\bar{a}_i, \bar{B}_i) && (2.65) \end{aligned}$$

with augmented hyperparameters i.e.

$$\begin{aligned} \bar{\lambda}_i &= \lambda_i + \gamma_i && (2.66)\\ \bar{\xi}_i &= \xi_i + \gamma_i && (2.67)\\ \bar{a}_i &= a_i + \gamma_i && (2.68)\\ \bar{\rho}_i &= \frac{\rho_i\xi_i + \bar{\omega}_i\gamma_i}{\xi_i + \gamma_i} && (2.69)\\ \bar{B}_i &= B_i + \gamma_i\,\bar{r}_i + \frac{\xi_i\,\gamma_i}{\xi_i + \gamma_i}(\rho_i - \bar{\omega}_i)(\rho_i - \bar{\omega}_i)^T && (2.70) \end{aligned}$$

MAP parameter estimation directly follows maximizing posterior distributions:

$$\bar{c}_i = \frac{\bar{\lambda}_i - 1}{\sum_i^M \bar{\lambda}_i - 1} \tag{2.71}$$

$$\bar{\mu}_i = \bar{\rho}_i \tag{2.72}$$

$$\bar{\Sigma}_i = \frac{\bar{B}_i}{\bar{a}_i - d} \tag{2.73}$$

where $d$ is the observation vector dimension. The idea of the MAP parameter estimation is evident in those formulae: parameter are estimated starting from prior that get modified by data.

### 2.4.3 MAP estimate for HMM

In this section we will focus on the MAP parameter estimation for HMM. Let us designate as before HMM parameters with $\theta = \{\pi, A, B\}$ where we consider here the discrete emission probability case. Let use define the following prior probabilities over the parameters as Dirichlet distributions i.e.:

$$p(\theta) = p(\pi) \prod_{ij} p(a_{ij}) \prod_{ik} p(b_{ik}) \tag{2.74}$$

$$p(\pi) = Dir(\lambda_\pi) \tag{2.75}$$

$$p(a_{ij}) = Dir(\lambda_{a_{ij}}) \tag{2.76}$$

$$p(b_{ik}) = Dir(\lambda_{b_{ik}}) \tag{2.77}$$

EM algorithm can be used again optimizing expression 2.54. The E-step will have a form similar to the ML E-step. The forward-backward algorithm can be used in order to compute $\gamma_t(i, j)$ defined as in 2.29.

Concerning the M-step, the maximization will consider also the prior data term in 2.54 i.e. $log(p(\bar{\theta}))$. Assuming a prior distribution over parameters as in 2.74, we can rewrite $log(p(\bar{\theta}))$ as:

$$log(p(\bar{\theta})) = log(p(\pi)) + log(p(a_{ij})) + log(p(b_{ik})) \tag{2.78}$$

It means that prior over parameters factories as other terms in the auxiliary function 2.31 and can be optimized independently. Again taking advantage from the fact that prior are chosen in the conjugate exponential family, posterior distributions have the same form of priors with augmented hyperparameters.

$$p(\bar{\pi}) = Dir(\bar{\lambda}_{\pi_i}), \ \ \bar{\lambda}_{\pi_i} = \lambda_{\pi_0} + \sum_{j}^{N} \gamma_0(i,j) \quad (2.79)$$

$$p(\bar{a}_{ij}) = Dir(\bar{\lambda}_{a_{ij}}), \ \ \bar{\lambda}_{a_{ij}} = \lambda_{a_{ij}} + \sum_{t=1}^{T} \gamma_t(i,j) \quad (2.80)$$

$$p(\bar{b}_{ik}) = Dir(\bar{\lambda}_{b_{ik}}), \ \ \bar{\lambda}_{b_{ik}} = \lambda_{b_{ik}} + \sum_{t}^{T} \sum_{i} \gamma_t(i,j) \quad (2.81)$$

MAP parameter estimation directly follows from parameter posterior.

$$\bar{\pi}_i = \frac{\bar{\lambda}_{\pi_i} - 1}{\sum \bar{\lambda}_{\pi_i} - 1} \quad (2.82)$$

$$\bar{a}_{ij} = \frac{\bar{\lambda}_{a_{ij}} - 1}{\sum \bar{\lambda}_{a_{ij}} - 1} \quad (2.83)$$

$$\bar{b}_{ik} = \frac{\bar{\lambda}_{b_{ik}} - 1}{\sum \bar{\lambda}_{b_{ik}} - 1} \quad (2.84)$$

As before, probability emission in a HMM can be modeled using a GMM, the extension is straightforward combining results from this section and previous section. Using GMM parameters as in section 2.4.2 we obtain:

$$\gamma_{ik} = \sum_{t=1}^{T} \xi_t(i,k) \quad (2.85)$$

$$\bar{\omega}_{ik} = \frac{\sum_{t=1}^{T} \xi_t(i,k) y_t}{\sum_{t=1}^{T} \sum_{k=1}^{M} \xi_t(j,k)} \quad (2.86)$$

$$\bar{r}_{ik} = \frac{\sum_{t=1}^{T} \xi_t(i,j)(y_t - \bar{\omega}_{jk})(y_t - \bar{\omega}_{jk})^T}{\sum_{t=1}^{T} \sum_{k=1}^{M} \xi_t(j,k)} \quad (2.87)$$

That results in hyperparameters for posterior distributions:

$$\bar{\lambda}_{c_{ik}} = \lambda_{c_{ik}} + \gamma_{ik} \quad (2.88)$$

$$\bar{\rho}_{ik} = \frac{\rho_{ik} \xi_i + \bar{\omega}_{ik} \gamma_{ik}}{\xi_i + \gamma_{ik}} \quad (2.89)$$

$$\bar{B}_{ik} = B_{ik} + \gamma_{ik} \, \bar{r}_{ik} + \frac{\xi_i \, \gamma_{ik}}{\xi_i + \gamma_{ik}} (\rho_{ik} - \bar{\omega}_{ik})(\rho_{ik} - \bar{\omega}_{ik})^T \quad (2.90)$$

Finally MAP parameters can be estimated as:

$$\bar{c}_{ik} = \frac{\bar{\lambda}_{c_{ik}} - 1}{\sum_{ik}^{M} \bar{\lambda}_{c_{ik}} - 1} \quad (2.91)$$

$$\bar{\mu}_{ik} = \bar{\rho}_{ik} \quad (2.92)$$

$$\bar{\Sigma}_{ik} = \frac{\bar{B}_{ik}}{a_{ik} - p} \quad (2.93)$$

## 2.5 MAP is not invariant to reparameterization

A serious problem with Maximum a Posteriori parameter estimation is that it is not invariant to different parameterization. Let us consider a simple example as in [91] given some data $Y$ and a parameter $\theta$ the MAP estimation is given by $argmax_{\theta} p(Y|\theta)p(\theta)$ while the ML estimation is given by $argmax_{\theta} p(Y|\theta)$. Let us consider another parameterization now as $u = log(\theta)$ and in order to make the prior over $u$ invariant to translation let us fix it as an improper prior $p(u) = c$ with $c = constant$. Considering that $\frac{du}{d\theta} = \frac{1}{\theta}$, prior distribution for $\theta$ becomes $p(\theta) = \frac{du}{d\theta} \times c = \frac{c}{\theta}$.

The function $log()$ is a monotonal function that means that if $\bar{u} = argmax_u p(Y|u)$ and $\bar{\theta} = argmax_{\theta} p(Y|\theta)$ then $\bar{u} = log(\bar{\theta})$ i.e. maximum likelihood parameter estimation is invariant to different parameterization.

On the other side let us consider the MAP estimator $\bar{\theta} = argmax_{\theta}[p(Y|\theta)\,\frac{c}{\theta}]$ and $\bar{u} = argmax_u[p(Y|u)\,c]$. It is evident that in general we will have $\bar{u} \neq log(\bar{\theta})$.

This is a very effective example that shows that changing parameterization can bring a completely different MAP parameter estimation. In general given a parameter setting it is always possible to find a basis in which it corresponds to mode of posterior distributions. This is of course a very unsuitable property because we would like an estimation independent from the choice of the basis that represent probability function.

An effect of parameterization problem for MAP can also be seen in the previously considered MAP estimator for GMM (as in section 2.4.2) and (as in section HMM) 2.4.3. Let us consider parameter estimation with a Dirichlet distribution for $c_i$ with parameters $\lambda_i$ as prior distribution:

$$P(c|\lambda) = \frac{1}{Z(\lambda)} \prod_i^I c_i^{\lambda_i - 1} \delta(\sum_i c_i - 1) \tag{2.94}$$

Posterior hyperparameters are $\bar{\lambda}_i = \lambda_i + \gamma_i$ and the MAP estimate for $c_i$ is as in equation (2.71):

$$\bar{c}_i = \frac{\bar{\lambda}_i - 1}{\sum_i^M \bar{\lambda}_i - 1} \tag{2.95}$$

Actually the maximum of the Dirichlet distribution does not coincides with the mean i.e. $\bar{\lambda}_i / \sum \bar{\lambda}_i$. If $\bar{\lambda}_i < 1$ (i.e. $\lambda_i < 1$ and for example $\gamma_i = 0$), the MAP estimation gives non-zero probability to a parameter value that's actually outside the considered domain i.e. $c_i \leq 0$; this can be simply avoided using $\bar{\lambda}_i \geq 1$ but also the case $\bar{\lambda}_i \leq 1$ has been studied in [97], [95] and [63]. The '-1' in the formula is a direct effect of the basis in which the Dirichlet distribution is represented; if the Dirichlet distribution is reparameterized into a soft-max basis as in [95] it would give a parameter estimate without the '-1'.

Let us consider the following basis transform where $\{c_i\}$ depends on a new set $\{c_i\}$:

$$c_i(a) = \frac{exp(a_i)}{\sum_j exp(a_j)} \tag{2.96}$$

Instead of the traditional basis, a soft-max basis is used. In this case $c_i(a)$ always normalize to 1 and there is a redundant degree of freedom because $a + n$ with $n = [1, \ldots, 1]^T$ leaves $p(a)$ unchanged. This extra degree of freedom can be frozen by imposing an arbitrary constraint. Let us designate with $g(a \times n)$ an arbitrary density that constrains the redundant degree of freedom. It is possible to write the Dirichlet distribution as :

$$P(a|\lambda) = \frac{1}{T(\lambda)} \prod_i b_i(a)^{\lambda_i} g(a \times n) \tag{2.97}$$

where $T_\lambda$ is normalization constant. At a first sight it can be noticed that the distribution has no more the '-1' in the distribution that means that we have no divergence problems. Then the maximum of the Dirichlet distribution in this basis is equal to the mean of the distribution.

The question that naturally arises at this point is what kind of Bayesian process are invariant w.r.t. choice of basis. In order to obtain a parameterization invariant estimation it is enough to marginalize w.r.t posterior distributions obtaining the marginal likelihood as objective function. We repeat again that for complex models that involve hidden variables, estimation cannot be done in closed form and approximations must be considered.

## 2.6 Model selection

Each modeling task starts with an hypothesis over the model; for example in a GMM, an hypothesis on the number of components is done while in an HMM the hypothesis is on the number of states and on the allowed transitions. A wrong model choice can bring to a completely ineffective learning resulting poor generalization and predictivity of the trained model. For this reason one of the most important task in machine learning is choosing the model that better fit to the data i.e. the *model selection*.

Let us define the problem in a mathematical point of view in probabilistic terms: let us consider some data $Y$ and a set of models $m = \{1, \ldots, M\}$ trained on $Y$. Let us suppose that our goal is to make prediction on some unseen data $U$ i.e. $p(U|m)$. Actually data $U$ are not known at the moment of the learning and best model must just be obtained using the information in the training data $Y$.

Let us consider a GMM example in which models just differ for the component number. The likelihood of the training set $p(Y|m)$ will reasonably increase as long as the component number increase and the model will fit always better the training data. Anyway when the model is too specialized, it will 'overfit' the training data giving a very good score $p(Y|m)$ but a very poor score $p(U|m)$ where $U$ is different set from $Y$. The goal of the model selection is to find the 'right' model $m$ that trained on the data set $Y$ can generalize as much as possible the properties of the data.

In next section we will consider some of the most popular model selection techniques.

### 2.6.1 Cross Validation

Cross validation is probably the simplest way to obtain a very general model from the training data. In order to simulate an unseen data set the training set is splitted into $k$ folders. Then the given model $m$ is trained $k$ times using only $k-1$ folders out of the $k$ and leaving the unseen folder for computing the score function (e.g. the classification error or simply the maximum likelihood). Finally an average over the $k$ trials is done. The best model is the model that holds the highest average score but this time average score is computed on data that were not available during the training. Pushing the idea of the cross validation to the extreme the number of folders can be chosen equal to the data size $N$ resulting in the so called 'leave-one-out' method.

A big advantage of cross validation is that no hypothesis on the model is done and it can be applied to any kind of model selection problem (topology, size, hidden variables etc..). On the other hand is evident that this method is extremely computationally expensive. In fact $k$ different models must be learned (one for each different folder) instead of one. Furthermore cross validation present some serious problems when the score function presents some discontinuities (like in Neural Network) or when the number of elements in the folder is too small.

To overcome those problems many methods that directly work on the model with the all training data set have been proposed.

### 2.6.2 Bayesian model selection

Model selection problem can also be considered from the Bayesian point of view. In fact following Bayes rule it is possible to write $p(m|Y) = p(Y|m)p(m)/p(Y)$ where $p(m|Y)$ is the posterior probability of the model given the data, $p(Y|m)$ is the data evidence given the model, $p(m)$ is the prior over model and $p(Y)$ is the data probability (that does not depend on $m$). Comparing two models $m_1$ and $m_2$ means comparing their posterior probabilities $p(m_1|Y)$ and $p(m_2|Y)$ i.e.

$$\frac{p(m_1|Y)}{p(m_2|Y)} = \frac{p(Y|m_1)p(m_1)}{p(Y|m_2)p(m_2)} \tag{2.98}$$

If no prior informations on the models are available i.e. $p(m_1) = p(m_2)$ comparing model posterior results in comparing data evidences $p(Y|m_1)$ and $p(Y|m_2)$. If models are parametric, data evidences can be obtained integrating over parameters $\theta$ i.e.

$$p(Y|m) = \int p(Y, \theta|m)d\theta = \int p(Y|\theta, m)p(\theta|m)d\theta \tag{2.99}$$

The data evidence (2.99) or marginal likelihood is the key quantity in the model selection framework. In fact the Bayesian integral has the remarkable property of embedding the Occam razor property [93]. Occam razor is a philosophical principle of the 14th century stating that one should not make more assumptions than needed and when multiple explanations are available the most simple must be preferred. It is a common (and wrong) opinion that Bayesian models just differ from non-Bayesian models because of the use of some prior information; Bayesian models automatically embed the Occam factor that allows comfortable model selection.

In order to show this interesting property let us consider the same example of [93] with a mono-dimensional distribution. Assuming that the parameter posterior distribution $p(\theta|Y, m) \propto p(Y|\theta, m)p(\theta|m)$ has a strong peak at the most probable parameters $\theta_{MP}$, the data evidence can be approximated by the value of the peak times its width $\sigma_{\theta|Y}$ i.e.

$$p(Y|m) \simeq p(Y|\theta_{MP}, m)\, p(\theta_{MP}|m)\sigma_{\theta|Y} \tag{2.100}$$

First factor in 2.100 $p(Y|\theta_{MP}, m)$ is the data likelihood computed in the most probable parameter fit while factor $p(\theta_{MP}|m)\sigma_{\theta|D}$ is the well known *Occam factor* (see [56]). Occam factor acts like a penalty term that penalizes more complex models and it is embedded in the Bayesian integral.

Let us make a further simplification considering $p(\theta_{MP}|m)$ uniform on a large interval $\sigma_\theta$ resulting in a prior distribution $p(\theta_{MP}|m) = 1/\sigma_\theta$. The Occam factor becomes $\frac{\sigma_{\theta|Y}}{\sigma_\theta}$. Interpretation of the Occam factor given in [93] consists in the posterior volume of $m$ parameter space divided by the prior accessible volume. A complex model is a model with a prior accessible volume $\sigma_\theta$ larger compared with a simpler model; in this case the Occam factor will penalize the more complex model. It is anyway important to notice that Occam factor does not simply depend on parameter number but prior and posterior probabilities are considered (i.e. the prior and posterior parameter accessible volume).

In the case of multidimensional parameters, same conclusion can be recovered assuming that the posterior distribution is well approximated by a Gaussian. In this case the data evidence can be written as:

$$p(Y|m) \simeq p(Y|\theta_{MP}, m) \times p(\theta_{MP}|m)\, |H/2\pi|^{-1/2} \tag{2.101}$$

where $H = -\nabla\nabla\, log\, p(\theta|Y,m)$ is the Hessian of the parameter posterior distribution. In this case the Occam factor is $p(\theta_{MP}|m)\,|H/2\pi|^{-1/2}$. This result is also known as the Laplace approximation that we will consider in the next section.

Figure 2.1 gives a visual explication of the Occam razor as described in [18] and [93]. Three models $m1$, $m2$ and $m3$ with three different marginal likelihood function of the space of possible data set. Model $m1$ is extremely simple and provides an extremely high value on a limited set of possible data; on the other side model $m2$ is more complex than $m1$, because marginal likelihood must integrate at 1, it will cover a larger set of possible data sets with a smaller value than $m2$. Finally the model $m3$ is the most complex that cover the larger data set and obviously have the smallest marginal likelihood. To select the best model given a particular instance of the data set, it is enough to compare the marginal likelihood for the three models.

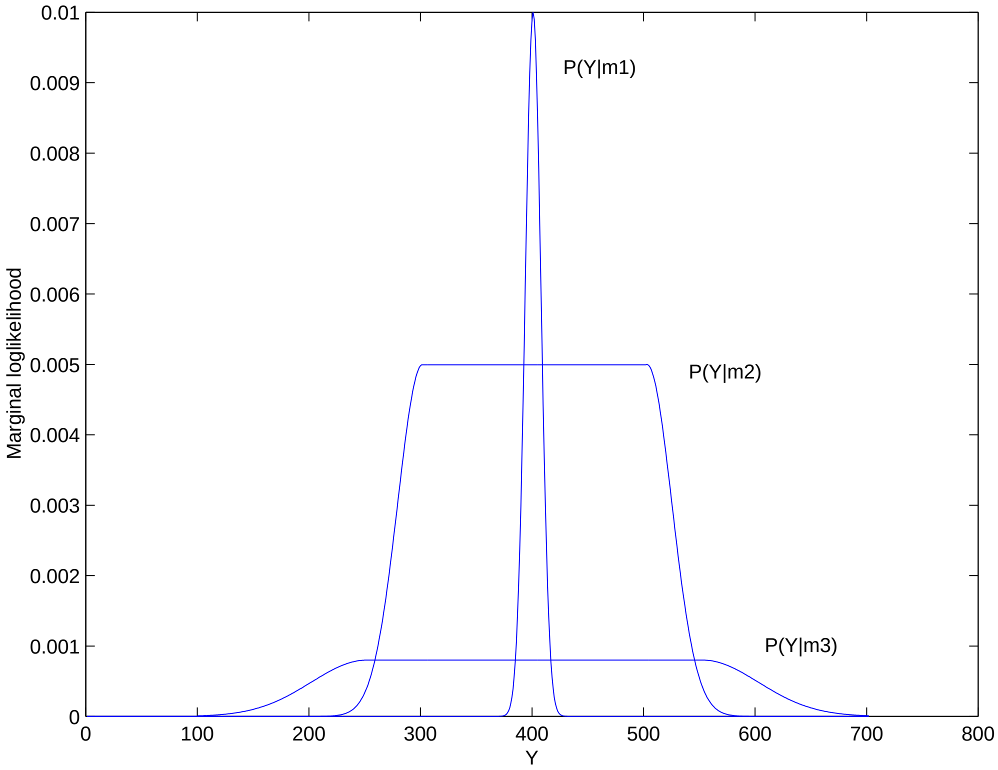


Figure 2.1: Visual explication of the Occam razor as described in [18] and [93]. Y denotes all possible data set.

### 2.6.3 Laplace approximation

The Laplace approximation is a local Gaussian approximation of the marginal likelihood (or data evidence) based on a MAP parameter estimation $\bar{\theta}$ (see [77],[92]).

Let us write the logarithm of the posterior parameter distribution as:

$$l(\theta) = log(p(\theta, Y|m)) = log(p(\theta|m)p(Y|\theta, m)) = log\, p(\theta|m) + \sum_{i=1}^{n} log\, p(y_i|\theta, m) \tag{2.102}$$

under the hypothesis of independence in the data set $\{Y_i\}$. Let us consider now the Taylor series of $l(\theta)$ around the MAP parameters $\bar{\theta}$:

$$l(\theta) = l(\bar{\theta}) + (\theta - \bar{\theta})^T \frac{\partial l(\theta)}{\partial \theta}|_{\theta=\bar{\theta}} + \frac{1}{2}(\theta - \bar{\theta})^T \frac{\partial^2 l(\theta)}{\partial \theta \partial \theta^T}|_{\theta=\bar{\theta}}(\theta - \bar{\theta}) + \dots \tag{2.103}$$

Let us stop the expansion to the second order; the linear term disappear because the development is done in a maximum of the function (MAP estimation). The second ordered term is actually the Hessian of the log posterior i.e.

$$H(\bar{\theta}) = \frac{\partial^2 l(\theta)}{\partial \theta \partial \theta^T}|_{\theta=\bar{\theta}} = -\nabla\nabla log(p(\theta|Y, m)) \tag{2.104}$$

It is now possible to rewrite the log marginal likelihood as:

$$\begin{aligned} log\, p(Y|m) = log \int d\theta exp[l(\theta)] \simeq l(\bar{\theta}) + \frac{1}{2} log|2\pi H^{-1}| = \\ = log(p(\bar{\theta}|m)) + log(p(Y|\bar{\theta}, m)) + \frac{d}{2} log(2\,\pi) - \frac{1}{2} log|H| \end{aligned} \tag{2.105}$$

with $d$ space dimension. It is finally possible to write the Laplace approximation for marginal likelihood as:

$$p(Y|m) \simeq p(\bar{\theta}|m)p(Y|\bar{\theta}, m)|2\pi H^{-1}|^{1/2} \tag{2.106}$$

Laplace approximation suffers of many drawbacks. First of all it is based on large data limit, and when this condition is not met the approximation can be very rough. On the other side the approximation is based on a Taylor series expansion that has as hypothesis some regularity conditions. When the model uses hidden variables the log posterior may not meet this regularity conditions resulting in a false approximation.

Furthermore Laplace approximation suffers from the main problem of all methods based on MAP estimation: it is a basis dependent solution as explained in section 2.5. For example when Laplace approximation of a Dirichlet distribution must be used it is worthy to express it in a soft-max basis instead of a classical basis in order to avoid singularity problems.

### 2.6.4 The Bayesian Information Criterion

The state-of-art model selection criterion in many speech system is the Bayesian Information Criterion (BIC) proposed in [120]. BIC can be directly derived by the Laplace approximation. Let us consider separately the logarithm of terms in 2.106: $log[p(\bar{\theta}|m)]$ does not depend on the data set size $n$, $log[p(Y|\bar{\theta}, m)]$ linearly increase with $n$ while $log|H|$ increases as $d\,log\,(n)$. BIC approximation just considers terms that grows with $n$. Furthermore asymptotically the Hessian can be simplified as follows:

$$lim_{n\to\infty}\frac{1}{2}log|H| = \frac{1}{2}log|nH_0| = \frac{d}{2}log(n) + \frac{1}{2}log|H_0| \tag{2.107}$$

where $d$ is the number of free parameters contained in the model. Keeping again just the term that grows with $n$ the well known BIC criterion can be derived:

$$logp(Y|m)_{BIC} = logp(Y|\bar{\theta}, m) - \frac{p}{2}log(n) \tag{2.108}$$

Expression 2.108 has a very intuitive explanation: models with a large number of parameters (i.e. $p$) are penalized more than models with fewer parameters. The only point that matter in the BIC penalty term is the model size and the logarithm of training set size. It does not depend at all on the prior parameter probability. This is of course a very rough approximation compared to Laplace approximation but it has the interesting properties of being bases independent.

The BIC approximation is equivalent to other non Bayesian model selection criteria. For instance in [122] the equivalence in linear system between the BIC and the cross validation with size of leave-out set equal to $n[1 - \frac{1}{logn-1}]$ is demonstrated . BIC is exactly equivalent to another model selection criterion coming from coding theory: the Minimum Description Length criterion (see [115]).

Because of its simplicity the BIC is the most used technique in many practical engineering model selection problems and in the following of this thesis we will compare it with the Variational Bayesian model selection.

### 2.6.5 Other methods

Laplace approximation and BIC are two of the most popular model selection criterion because of their simplicity and their low computational costs. Anyway other approximations have been studied in literature. For example another popular approximation is the Cheeseman-Stutz approximation (see [30]) that uses multiple Laplace approximation in order to try to compensate errors.

On the other side when available data is extremely poor, those approximations are ineffective and inaccurate. In those cases numerical integration methods can give a more precise answer even if they are extremely expensive in terms of computational time and almost inapplicable when the parameter space is too huge. Those algorithms belong to the family of the Monte Carlo methods and probably the most popular methods is the Markov Chain Monte Carlo (MCMC) methods as in [104]. Bayesian variants of Monte Carlo methods integration have been proposed as well (see [111]).

## 2.7 Conclusion

In this chapter we have revised the fundamental concepts of generative learning with hidden variables. We have shown how Maximum likelihood learning can be performed using the Expectation-Maximization algorithm. On the other side fully Bayesian learning with hidden variable cannot be performed in the same way. Bayesian inference i.e. inference in models where all quantities are integrated out is impossible in close form or using iterative algorithm like the EM. On the other side Bayesian inference has the appealing property of embedding the Occam razor property that is extremely useful in the model selection framework.

Bayesian inference is generally done in an approximated way for example using the well known Maximum a Posteriori criterion that is *not* fully Bayesian and suffers from reparameterization problems but is tractable in the case of hidden variables with an algorithm similar to the EM.

The Bayesian integral useful in the model selection task can be approximated in various way or computed with numerical methods (that are generally resource consuming and unsuitable when the parameter space becomes large). For instance Laplace method makes a local Gaussian approximation around the MAP parameter estimation asymptotically true. These approximations can be very inaccurate because of poor available data or because parameter posterior are poorly approximated by a Gaussian. Further simplified criteria are the BIC and the MDL that approximate asymptotically the Bayesian integral.

In the rest of this work we will consider another type of approximation that allows a posterior distribution estimation and define a lower bound on the Bayesian integral. This approximation turns out to be extremely effective and an iterative algorithm similar to the EM (and with the same computational cost) can be derived in order to learn posterior distributions and Bayesian integral.

# Chapter 3

# Variational Bayesian Learning

Variational methods are approximated methods that simplify the original problem by adding extra parameters (the variational parameters) that are trained in order to provide an approximated solution instead of original ones. From an historical point of view the term *variational* comes from the roots of the techniques in the calculus of variations. Readers interested in a general perspective can refer to [73], anyway in this work we will generally focus on the variational Bayesian methods.

The core of variational approximation is in the capacity of bounding a given function of interest $f(\theta)$. Convex analysis theory is a useful tool for bounding convex function (see [116]) and convex duality principle that states a concave function $f(x)$ can be represented as a dual function :

$$f(\theta) = min_{\lambda}\{\lambda^T \theta - f^*(\lambda)\} \tag{3.1}$$

$$f^*(\theta) = min_{\theta}\{\lambda^T \theta - f(\theta)\} \tag{3.2}$$

Let us now consider a probabilistic problem in a general graphical model with observed variables $Y$ and hidden variables $X$ ([73]). Data evidence $p(Y|\theta) = p(Y,X|\theta)/p(X|Y,\theta)$ is strictly related with the joint hidden variable and observed variable probability. Variational approximations generally address the problem of intractable form of joint $p(Y,X|\theta)$ probability. For example it is possible to consider an upper bound $p_U(Y,X|\theta_U)$ to the joint distribution ($\theta_U$ are bound parameters i.e. the variational parameters and $p_U$ is the variational approximation bound). An extremely important point is to choose the upper bound *tractable* in order to allow efficient computation e.g. likelihood $\sum_X p_U(Y,X|\theta_U)$. The variational transformation from $p(Y,X|\theta)$ to $p_U(Y,X|\theta_U)$ is generally achieved transforming dependency between variables and considering variational distribution parameters $\theta_U$ instead of $\theta$. The transformation must result in a tractable form and at the same time must be as close as possible to the original probability in order to obtain an accurate approximation.

In literature variational algorithm can be classified in two main groups according to [73]: sequential and block algorithms. In sequential algorithm probability dependence is transformed during the learning process allowing the evidence to determine the best path (corresponding to the optimal set $\{X\}$; see for example [61],[60]). On the other hand in the block algorithms, transformation is defined off-line, according to some knowledge of tractable structure that can be used. Block methods were first introduced in [119] and developed in [49],[48] and [74]. Both sequential and block approaches can be unified on the bases of the convex duality (see [59]).

Mathematically speaking in block algorithm, the set of variables $Y^{'}$ and $X^{'}$ that gives some kind of intractability are selected and their distribution $p(X^{'}|Y^{'})$ is approximated with $q(X^{'}|Y^{'},\theta)$ where $q()$ is the variational distribution with parameters $\theta$. Variational distribution $q()$ is generally chosen in order to minimize the distance between the true and the approximated distribution i.e.

$$\bar{\theta} = argmin_{\theta} KL(q(X^{'}|Y^{'},\theta)||p(X^{'}|Y^{'})) \tag{3.3}$$

where $KL$ designates the Kullback-Leibner divergence ([35]). The best variational approximation becomes so $q(X^{'}|Y^{'},\bar{\theta})$. The use of the KL divergence can be justified using the bounding theory for convex functions. The data evidence can in fact be bounded by:

$$\begin{aligned} log\, p(Y^{'}) &= log \sum_{X^{'}} p(X^{'},Y^{'}) = log \sum_{X_{\prime}} q(X^{'}|Y^{'},\theta)\, \frac{p(X^{'},Y^{'})}{q(X^{'}|Y^{'},\theta)} \\ &\geq \sum_{X^{'}} q(X^{'}|Y^{'},\theta)\, log\, \frac{p(X^{'},Y^{'})}{q(X^{'}|Y^{'},\theta)} = KL(q(X^{'}|Y^{'},\theta)||p(X^{'}|Y^{'})) \end{aligned} \tag{3.4}$$

where Jensen's inequality has been applied ([66]). So the tightest bound to the evidence is the one with $\theta = \bar{\theta}$.

In this work we will focus on two main applications for variational methods: *parameter training* and *Bayesian model selection*. In fact when parameters cannot be estimated, they can be substituted by variational parameters. If the variational bound is chosen in an appropriate way, the lower bound can be optimized using an algorithm that generalizes the well known EM; this result was proposed for the first time in [103] but Bayesian application of variational learning in generic graphical models was considered in [36]. Variational methods are also known as *ensemble learning* from a well known paper on neural network ([57]) where "ensemble" of neural network are fitted to data. In those cases the variational approximation concerns the posterior distribution that cannot be computed in a tractable way. Variational Bayesian methods have been applied to a variety of different models like Gaussian mixture models [6], hidden Markov models [94] and mixture of experts [142].

In this chapter we will detail variational Bayesian methods with specific applications to HMM and GMM for model selection and model learning.

## 3.1 Variational algorithms for Bayesian learning

In this section we mathematically detail variational methods for Bayesian learning. Let us consider again some data $Y$, hidden variable set $X$ and a parameter set $\theta$ for a given model $m$. The key quantity for fully Bayesian learning (as described in previous chapter) is the log-marginal likelihood:

$$log\, p(Y|m) = log \int d\theta dX p(Y, X, \theta|m) \tag{3.5}$$

When hidden variables are used it is not possible to derive the posterior parameter and hidden variables distribution $p(\theta, X|Y, m)$ and to compute 3.5 in an exact way without using numerical methods. Here comes the need for the variational approximation. So let us introduce a variational distribution $q(X, \theta)$ in order to approximate the true (and unknown) $p(X, \theta|Y, m)$. Applying the Jensen inequality is possible to define an upper bound on the marginal log-likelihood as:

$$\begin{aligned} log\, p(Y|m) &= log \int d\theta dX p(Y, X, \theta|m) = log \int d\theta dX q(X, \theta) \frac{p(Y, X, \theta|m)}{q(X, \theta)} \\ &\geq \int dX d\theta q(X, \theta) log\, \frac{p(Y, X, \theta|m)}{q(X, \theta)} \end{aligned} \tag{3.6}$$

The bound represented as in 3.6 is still an untractable bound. The key assumption is to assume that the variational distribution can factorize over parameters $\theta$ and hidden variables $X$ i.e. $q(X, \theta) = q(X)q(\theta)$. Now an iterative procedure that optimize $q(X)$ and $q(\theta)$ can be derived. In fact 3.6 can be further simplified as:

$$\begin{aligned} log\, p(Y|m) &\geq \int d\theta dX log\, q(X)q(\theta) \frac{p(Y, X, \theta|m)}{q(X)q(\theta)} = \\ &= \int d\theta q(\theta) [\int dX q(X) log \frac{p(Y, X|\theta, m)}{q(X)} + log\, \frac{p(\theta|m)}{q(\theta)}] = \\ & \int d\theta q(\theta) \int dX q(X) log\, p(Y, X|\theta, m) - \int dX q(X)\, log\, q(X) + \\ &- log\, \frac{q(\theta)}{p(\theta|m)} = F_m(q(X), q(\theta)) \end{aligned} \tag{3.7}$$

Expression 3.7 is also known as variational energy or *free energy*. The goal of variational learning is to find optimal $q(X)$ and $q(\theta)$ that maximize the free energy. 3.7 is composed by the expected energy given $q(X)$ minus the entropy of

$q(X)$ (see [43],[103]), minus a penalty term that becomes larger with the number of parameters. Because of the assumed factorization an algorithm similar to the EM algorithm generally known as *Variational Bayesian Expectation Maximization* (VBEM) ([6]) can be derived.

### 3.1.1 Variational Bayesian EM

In this section we develop the VBEM algorithm that allows an iterative optimization of the free energy that represent a lower bound on the log marginal likelihood. Let us suppose that data $Y = \{y_1, \dots, y_n\}$ are i.i.d. and that model $m$ contains parameters $\theta$ and hidden variables $X = \{x_1, \dots, x_n\}$. Optimization of free energy 3.7 can be achieved simply deriving w.r.t $q(\theta)$ and $q(X)$ and equating to zero. This is a case of constrained optimization because $q(X)$ and $q(\theta)$ must be probability density functions i.e. $\int dx_i \, q(x_i) = 1$ and $\int q(\theta) d\theta = 1$. We will denote with $a^{(t)}$ the estimation of variable $a$ at iteration $t$. In the iteration process posteriors $q(\theta)$ and $q(X)$ are successively reestimated.

Let us consider the derivative of 3.7 w.r.t $q(X)$ using Lagrange multipliers method to enforce the constraint $\int dx_i \, q(x_i) = 1$:

$$\begin{aligned} &\frac{\partial}{\partial q(X)} \{F_m(q(X), q(\theta)) + \lambda[\int dX q(X) - 1]\} = \\ &\int d\theta \, q(\theta) \frac{\partial}{\partial q(X)} \int dX \, q(X) log \frac{p(X, Y|\theta, m)}{q(X)} + \lambda = \\ &\int d\theta q(\theta)[log \, p(X, Y|\theta, m) - log \, q(X) - 1] + \lambda = 0 \qquad (3.8) \\ & \qquad (3.9) \end{aligned}$$

Solving w.r.t q(X) we obtain:

$$log \, q(X)^{(t+1)} = \int d\theta \, q(\theta)^{(t)} \, log \, p(X, Y|\theta, m) + [\lambda - 1] \tag{3.10}$$

Enforcing the Lagrange multipliers $\lambda$, the normalizing constant $Z(X)^{(t+1)}$ is obtained i.e.:

$$Z(X)^{(t+1)} = \int dX \, exp(\int d\theta \, q(\theta)^{(t)} \, log \, p(X, Y|\theta, m)) \tag{3.11}$$

We finally obtain for the update at time $(t + 1)$:

$$log \, q(X)^{(t+1)} = \int d\theta \, log \, p(X, Y|\theta, m) - log \, Z(X)^{(t+1)} \tag{3.12}$$

Let us consider now the i.i.d. hypothesis. It is now possible to factorize $q(X)^{(t+1)}$ as $q(X)^{(t+1)} = \prod_i^n q(x_i)^{(t+1)}$ (and in consequence the normalization constant

$Z(X)^{(t+1)} = \prod_i (Z(x_i)^{(t+1)})$:

$$log\, q(x_i)^{(t+1)} = \int d\theta\, log\, p(x_i, y_i|\theta, m) - log\, Z(x_i)^{(t+1)} \quad \forall\, i \Rightarrow$$
$$q(x_i)^{(t+1)} = \frac{1}{Z(x_i)^{(t+1)}}\, exp[\int d\theta\, q(\theta)^{(t)}\, log\, p(x_i, y_i|\theta, m)] \quad \forall\, i \qquad (3.13)$$

Let us now derive w.r.t $q(\theta)$ enforcing the constraint $\int d\theta\, q(\theta) = 1$ using Lagrange multipliers.

$$\frac{\partial}{\partial q(\theta)}\{F_m(q(X), q(\theta)) + \lambda[\int d\theta q(\theta) - 1]\} =$$
$$\frac{\partial}{\partial q(\theta)} \int d\theta\, q(\theta)[\int dX q(X) log\, p(X, Y|\theta, m) \; + \; log\, \frac{p(\theta|m)}{q(\theta)}] + \lambda =$$
$$\int dX\, q(X)\, log\, p(Y, X|\theta, m) + log\, p(\theta|m) - log\, q(\theta) + \lambda = 0 \qquad (3.14)$$

Enforcing the Lagrange multiplier, it is possible to obtain the normalization constant $Z(\theta)$ as before. It is finally possible to write the update for parameter distributions.

$$log\, q(\theta)^{(t+1)} = log\, p(\theta|m) \; + \int dX\, q(X)^{(t+1)}\, log\, p(Y, X|\theta, m) - log\, Z(\theta)^{(t+1)} \Rightarrow$$
$$q(\theta)^{(t+1)} = \frac{1}{Z(\theta)}\, p(\theta|m)\, exp[\int dX\, q(X)^{(t+1)}\, log\, p(Y, X|\theta, m)] \qquad (3.15)$$

We show below that an analogy with the EM algorithm the update of hidden variables is called E-like step while the update of parameters is called M-like step. The VBEM will converge into a local maxima of the objective function (as in the classical EM). In order to summarize the algorithm let us write the E-like step and the M-step like:

$$\text{VBE step: } q(x_i)^{(t+1)} = \frac{1}{Z_{x_i}}\, exp[\int d\theta\, q(\theta)^{(t)}\, log\, p(x_i, y_i|\theta, m)] \;\forall i \qquad (3.16)$$
$$\text{with } q(X)^{(t+1)} = \prod_{i=1}^{N} q(x_{i+1})^{(t+1)} \qquad (3.17)$$
$$\text{VBM step: } q(\theta)^{(t+1)} = \frac{1}{Z(\theta)}\, p(\theta|m) exp[\int dX\, q(X)^{(t+1)}\, log\, p(Y, X|\theta, m)] \qquad (3.18)$$

A first important consideration is that this is a free form optimization in the sense that no hypothesis are done on the form of the distribution $q(X)$ and

$q(\theta)$; the only assumption is that they are factorizable; for the rest the VBEM optimization is a very general procedure.

Furthermore VBEM is a fully Bayesian framework in the sense that it produces parameter posterior distribution (i.e. the variational posterior distribution $q(\theta)$) and not any explicit parameter estimation (like the MAP). In this sense VB estimation has the useful property of being invariant to reparameterization. Furthermore because of the fact free energy is an approximation of the Bayesian integral, it directly benefits from the Occam razor property (see section 2.6.2).

Concerning the choice of variational distribution $q(\theta)$ an important hint can come from expression 3.18. In fact we will recognize that posterior $q(\theta)$ is the product of a factor depending on data and a prior distribution $p(\theta|m)$: this suggests that choosing the prior distribution belonging to the exponential conjugate family helps in the tractability of the problem resulting in posterior distributions with the same form priors.

As already noted into the introduction, optimizing the free energy means finding the variational distributions that minimize the KL divergence with the truth joint posterior over parameters and hidden variables i.e.:

$$\begin{aligned} log\, p(Y|m) - F_m = \int d\theta\, dX\, q(\theta)q(X)\, log\, \frac{q(\theta)q(X)}{p(Y,X|\theta,m)} = \\ D(q(\theta)q(X)||p(Y,X|\theta,m)) \end{aligned} \tag{3.19}$$

This is generally visually depicted as in figure 3.1.

### 3.1.2 VBEM: another point of view

We have previously considered the case of two different variational distributions for hidden variables and for model parameters. In another point of view parameters and hidden variables are considered at the same level as stochastic variables defined by a probability distribution ( see [25],[23]). Looking at the VBE-step 3.16 and to the VBM-step 3.18, it is easy to notice a certain symmetry; There is apparently no prior term in VBM-step 3.18, but actually there is a prior term embedded in the joint probability $p(X,Y|\theta,m)$.

Let us designate the set of hidden variables and parameters with $R = \{X, \theta\}$: the factorization hypothesis becomes:

$$q(R|Y) = \prod_i q_i(R_i|Y) \quad \forall i \tag{3.20}$$

In this case the free energy will show no distinction between hidden variables and parameters:

$$F_m = \sum_R q(R|Y) log\, \frac{p(R,Y)}{q(R|Y)} \tag{3.21}$$

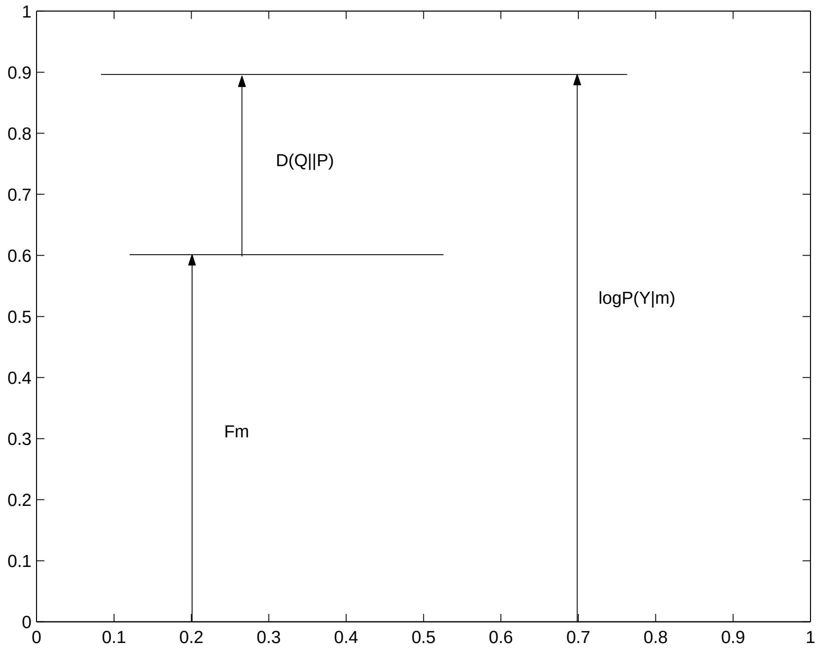


Figure 3.1: The marginal log-likelihood $log\, P(Y|m)$ differs from the free energy $F_m$ from the KL divergence between variational posterior distributions $Q$ and prior distributions $P$.

Variational free energy 3.21 can now be optimized fixing all variables $R_i$ but one $R_j$ deriving and solving w.r.t $R_j$ that gives:

$$q(R_j) = \frac{< P(R,Y) >_{i \neq j}}{\sum_R < P(R,Y) >_{i \neq j}} \tag{3.22}$$

where $< \, . \, >_{i \neq j}$ designates the expected value w.r.t. all $R_i$ but $R_j$. Iteration over all distribution $R_i$ will give a complete step over all distributions. The normalization constraint $\int q(R|Y)dR = 1$ must be of course imposed.

There is anyway a significant difference between hidden variables and model parameters in term of complexity: in fact model parameters depends just on the topology of the model while hidden variable number increases with the volume of the training data set.

### 3.1.3 Empirical prior for VBEM

Up to this moment we have not mentioned any problem concerning the prior distribution $p(\theta|m)$ that appears in the free energy. As long as the prior distribution has a parametric form, it is defined by its own parameters called *hyperparameters*. Hyperparameters (denoted with $\lambda$) can be considered as quantities to be optimized themselves; this is the idea of the empirical prior. In this case the VBEM algorithm will start with an initial guess over $\lambda$, and once the convergence of the free energy is achieved, hyperparameters are optimized i.e.

$$\begin{aligned} \text{initialize } \lambda \ & \rightarrow F_m(q(X), q(\theta), Y, \lambda) && (3.23) \\ \text{VBEM: } \{\bar{q}(X), \bar{q}(\theta)\} &= argmax_{\{q(X),q(\theta)\}} F_m(q(X), q(\theta), Y, \lambda) && (3.24) \\ \text{optimize } \lambda\text{: } \bar{\lambda} &= argmax_{\lambda} F_m(\bar{q}(X), \bar{q}(\theta), Y, \lambda) && (3.25) \end{aligned}$$

Generally 3.25 will result in a non-linear system of equations that must be solved with numerical methods.

Anyway other possibilities for handling prior distributions can be considered as in section 2.3.4.

## 3.2 Approximated learning for ML and MAP

Variational methods are extremely useful for intractable form like marginal likelihood; anyway they can also be applied to tractable forms like the Maximum Likelihood or Maximum a Posteriori criteria. In this section we will show the link between EM for ML and MAP and VBEM showing that the VBEM is a generalization of the Expectation-Maximization algorithm. This is intuitively reasonable by considering that MAP and ML criteria are special case of marginal likelihood maximization (see [18]).

### 3.2.1 VBEM for Maximum Likelihood

The objective function for maximum likelihood is obtained marginalizing w.r.t. hidden variable set $X$:

$$log\, p(Y|\theta) = \sum_{i=1}^{n} log\, p(y_i|\theta) = \sum_i log \int dx_i p(x_i, y_i|\theta) \qquad (3.26)$$

As stated earlier, the ML estimation considers $\theta_{ML} = argmax_\theta log\, p(Y|\theta)$.

A variational bound can also be considered on the log-likelihood. Given the convexity property of 3.26, a bound can be derived using any approximated distribution $q(x)$ over hidden variables. The bound can be simply derived using Jensen inequality as follows:

$$\begin{aligned} log\, p(Y|\theta) &= \sum_{i=1}^{n} log \int dx_i\, q(x_i) \frac{p(x_i, y_i|\theta)}{q(x_i)} \\ &\geq \sum_i \int dx_i\, q(x_i)\, log\, \frac{p(x_i, y_i|\theta)}{q(x_i)} = F_m \qquad (3.27) \end{aligned}$$

Expression (3.27) is a free energy as well as in the VB case i.e. expression (3.7) and it represents a lower bound on the exact log-likelihood. The same structure can be recovered rewriting the expression as:

$$F_m = \sum_{i=1}^{n} \int dx_i\, q(x_i)\, log\, p(x_i, y_i|\theta) - \int dx_i q(x_i)\, log\, q(x_i) \tag{3.28}$$

We can notice the same structure of an energy form conditional on $q(x_i)$ and the entropy of the conditional distribution $q(x_i)$ as in (3.7).

It is easy to verify that if the approximated variational distribution $q(x_i)$ is chosen as the exact posterior distribution for hidden variables $p(x_i|y_i, \theta)$ then the bound reduces to the exact log-likelihood:

$$\begin{aligned} F_m &= \sum_i \int dx_i\, q(x_i)\, log\, \frac{p(x_i, y_i|\theta)}{q(x_i)} = \sum_i \int dx_i p(x_i|y_i, \theta)\, log\, \frac{p(x_i, y_i|\theta)}{p(x_i|y_i, \theta)} \\ &= \sum_i \int dx_i p(x_i|y_i, \theta)\, log\, p(y_i|\theta) = \sum_i log\, p(y_i|\theta) = log\, p(Y|\theta). \end{aligned} \tag{3.29}$$

This result can be inferred applying the VBEM algorithm to the free energy (3.7) enforcing the constraint on the hidden variable distribution $\int dx_i q(x_i) = 1$. Indeed deriving w.r.t. $q(x_i)$ and solving:

$$\begin{aligned} &\frac{\partial}{\partial q(x_i)}[\sum_i \int dx_i\, q(x_i)\, log\, \frac{p(x_i, y_i|\theta)}{q(x_i)} + \sum_i \lambda_i \{\int dx_i q(x_i) - 1\}] \\ &\qquad = log\, p(x_i, y_i|\theta) - log\, q(x_i) - 1 + \lambda_i = 0 \end{aligned} \tag{3.30}$$

Enforcing the Lagrange multiplier constraint we obtain:

$$\lambda_i = 1 - log \int dx_i p(x_i, y_i|\theta) \quad \forall i \tag{3.31}$$

and so:

$$q(x_i) = p(x_i|y_i, \theta) \quad \forall i \tag{3.32}$$

In summary optimizing the free energy w.r.t. $q(x_i)$ results in finding exact posterior distributions. In consequence, the M-step becomes:

$$\bar{\theta} = argmax_\theta\, F(q(x_i)) = argmax_\theta \sum_i \int dx_i\, p(x_i|y_i, \theta)\, log\, p(x_i, y_i|\theta) \tag{3.33}$$

Solution is exactly the auxiliary function used in the EM algorithm even if we started our discussion with an approximated free energy.

### 3.2.2 Approximated training

In extremely complicated models where interactions between a huge number of variables is considered, posterior distribution (i.e. $p(x_i|y_i,\theta)$) cannot be computed. In those cases the EM algorithm cannot be applied as long as in the E step posterior distributions over hidden variables must be considered.

In those cases an approximated posterior distribution must be considered instead of the exact one. As before a mandatory hypothesis it to consider independence of hidden variables $x_i$ for obtaining a tractable approximation. In this case, the free energy does not simply reduce to the log-likelihood but we have:

$$\begin{aligned} F_m &= \sum_i \int dx_i q(x_i) log\, \frac{p(x_i,y_i|\theta)}{q(x_i)} = \\ &= \sum_i \int dx_i q(x_i) log\, p(y_i|\theta) + \sum_i \int dx_i\, q(x_i)\, log\, \frac{p(x_i|y_i,\theta)}{q(x_i)} \\ &= \sum_i log\, p(y_i|\theta) - \sum_i \int dx_i\, q(x_i)\, log\, \frac{q(x_i)}{p(x_i|y_i,\theta)} \end{aligned} \tag{3.34}$$

Because of the fact $log\, p(y_i|\theta)$ does not depend on hidden variables optimizing 3.34 means minimizing the KL divergence between real joint posterior distribution of hidden variables i.e.

$$\bar{q}(x_i) = argmin_{q(x_i)} \int dx_i q(x_i) log\, \frac{q(x_i)}{p(x_i|y_i,\theta)} = argmin_{q(x_i)}\, D(q(x_i)||p(x_i|y_i,\theta)) \tag{3.35}$$

In general the bound will not coincide with the log-likelihood because $q(x_i)$ will just approximate with the posterior $p(x_i|y_i,\theta)$.

In the M-step the expectation will be taken w.r.t. $q(x_i)$ instead of $p(x_i|y_i,\theta)$:

$$\bar{\theta} = argmax_\theta \sum_i dx_i\, q(x_i) log\, p(x_i,y_i|\theta) \tag{3.36}$$

### 3.2.3 VBEM for MAP

In the case of MAP estimation, the optimization algorithm must handle prior distributions on parameters together with hidden variables.

$$\theta_{MAP} = argmax_\theta\, [log\, p(\theta) + \sum_i^n\, log \int dx_i p(x_i,y_i|\theta)] \tag{3.37}$$

The E-step is not concerned by prior over parameters and for this reason it can be treated as in previous section; if a close form for posterior over hidden variables is available the variational bound will reduce to the exact log-likelihood, otherwise the approximation must be used in order to reduce the distance between the variational distribution and the real one.

The M-step will simply consider prior distributions into the optimization task i.e. in the case of exact posteriors:

$$\theta_{MAP} = argmax_{\theta}[log\, p(\theta) + \sum_{i}^{n} \int dx_i p(x_i|y_i, \bar{\theta})\, log\, p(x_i, y_i|\theta)] \tag{3.38}$$

In this case the M step coincides with the M-step for classical MAP described in section 2.4.1. On the other hand in case of variational optimization:

$$\theta_{MAP} = argmax_{\theta}[log\, p(\theta) + \sum_{i}^{n} \int dx_i\, q(x_i)\, log\, p(x_i, y_i|\theta)] \tag{3.39}$$

### 3.2.4 ML and MAP as special cases of VB

Let us consider the extreme case in which the variational distribution over parameters is Dirac delta i.e. $q(\theta) = \delta(\theta - \theta^*)$. Let us consider the case of variational distribution over parameters (without hidden variables for simplicity), the free energy can be decomposed as:

$$\begin{aligned} F_m = \int d\theta q(\theta)\, log\, p(Y|\theta)p(\theta) - \int d\theta q(\theta)\, log\, q(\theta) = \\ = \int d\theta\delta(\theta - \theta^*)\, log\, p(Y|\theta)p(\theta) - \int d\theta\delta(\theta - \theta^*)\, log\, \delta(\theta - \theta^*) = \\ log\, p(Y|\theta^*)p(\theta^*) - \int d\theta\delta(\theta - \theta^*)\, log\, \delta(\theta - \theta^*) \end{aligned} \tag{3.40}$$

The term $\delta(\theta - \theta^*) log\, \delta(\theta - \theta^*)$ is constant and does not take part into the optimization. Maximizing free energy gives in this case:

$$\theta^*_{MAP} = argmax_{\theta^*}[log\, p(Y|\theta^*)p(\theta^*)] = argmax_{\theta^*}[log\, p(Y|\theta^*)p(\theta^*)] \tag{3.41}$$

Expression 3.41 is actually the MAP estimation for parameter $\theta^*$. This enforce our statement that MAP is a point estimation that just considers 'density' instead of 'mass'. If the variational distribution is constrained to be a delta i.e. all its mass is concentrated in a point then the VB estimation reduces to the MAP estimation. Anyway it must be pointed out again that the MAP solution is always dependent on the choice of the representation of $p(\theta)$; that means that while the VB will converge to the same solution independently from the chosen basis, once the distribution is constrained to be a delta, the solution will depend on the representation chosen for $\theta$.

On the other side, in the case of Maximum Likelihood learning, if the variational distribution is constrained to be a Dirac delta, the result will be basis independent i.e.

$$
\begin{aligned}
F_m = \int d\theta q(\theta)\, log\, p(Y|\theta) - \int d\theta q(\theta)\, log\, q(\theta) = \\
= \int d\theta \delta(\theta - \theta^*)\, log\, p(Y|\theta) - \int d\theta \delta(\theta - \theta^*)\, log\, \delta(\theta - \theta^*) = \\
log\, p(Y|\theta^*) - \int d\theta \delta(\theta - \theta^*)\, log\, \delta(\theta - \theta^*)
\end{aligned}
\quad (3.42)
$$

that results simply into the classical ML criterion because $\delta(\theta - \theta^*) log\, \delta(\theta - \theta^*)$ is constant:

$$
\theta^*_{ML} = argmax_{\theta^*} log\, p(Y|\theta^*) = argmax_{\theta^*} log\, p(Y|\theta^*) \quad (3.43)
$$

To summarize MAP and ML criterion can be recovered from variational learning when variational distributions are assumed over parameters and are constrained to be a Dirac delta. Anyway while VB and ML will converge to same results as long as they are representation independent, the MAP solution will always depend on the choice of the representation.

## 3.3 VBEM for Conjugate-Exponential models

In this section we discuss the application of variational Bayesian methods to a general model. The following problem has been studied in [52] and [18] coming to conclusion that convenient update in the VBEM algorithm can obtained for all models belonging to the Conjugate-Exponential (CE) model. Let us first consider the definition of CE model.

### 3.3.1 Conjugate-exponential models

Conjugate-exponential models are well known models in statistical inference; for an exhaustive review see [14] and [29]. Let us consider a given model that represents data $Y$ with hidden variables $X$, parameters $\theta$ and prior over parameters $p(\theta|\lambda)$; $p(X, Y|\theta)$ belongs to the conjugate-exponential family if it satisfy two conditions:

Condition 1 : The complete data likelihood belongs to the exponential family i.e.

$$
p(X, Y|\theta) = g(\theta) f(X, Y) exp(\phi(\theta)^T u(X, Y)) \quad (3.44)
$$

with $u(X, Y)$ and $f(X, Y)$ are functions that characterize the exponential family, $\phi(\theta)$ is a vector of natural parameters and $g(\theta)$ is simply the normalization constant resulting from integration of 3.44 w.r.t. $X$ and $Y$.

Condition 2 : The parameter prior distribution is conjugate to the complete-data likelihood i.e.:

$$p(\theta|\eta,\nu) = h(\eta,\nu)g(\theta)^{\eta} exp(\phi(\theta)^T \nu) \tag{3.45}$$

with $\lambda = \{\eta, \nu\}$ hyperparameters and $h(\eta, \nu)$ normalization constant coming from integrating out 3.45 w.r.t. $\theta$.

Once conditions (1) and (2) are satisfied, posterior distributions have the same analytical form as prior distributions with augmented hyperparameters $\bar{\lambda}$ that can be interpreted as a modification of prior hyperparameters $\lambda$ by data.

### 3.3.2 VBEM for CE

Once conditions (1) and (2) are met, the VBEM algorithm can be derived with tractable updates for the E step and for the M step (see [52] and [18]). Let us consider variational distributions over hidden variables $q(X)$ and parameters $q(\theta)$ and let us assume the factorization $q(\theta, X) = q(\theta)q(X)$.

The VB-E step is:

$$\begin{aligned} q(x_i) &= g_{x_i} f(x_i, y_i) \, exp[\bar{\phi}^T u(x_i, y_i)] = p(x_i|y_i, \bar{\phi}) & (3.46) \\ q(X) &= \prod_{i=1}^{N} q(x_i) & (3.47) \end{aligned}$$

where $g_{x_i}$ is a normalization constant and $\bar{\phi}$ is:

$$\bar{\phi} = \int d\theta \, q(\theta)\phi(\theta) =< \phi(\theta) >_{q(\theta)} \tag{3.48}$$

where $< . >_{q(\theta)}$ designates the expectation w.r.t. $q(\theta)$.

This result can be simply obtained substituting the CE model into the VBE-step 3.16 i.e. :

$$q(X) = \frac{1}{Z_X} exp(< log \, p(X, Y|\theta, m) >_{q(\theta)}) =$$

$$= \frac{1}{Z_X} exp(\sum_{i=1}^{N} < log \, g(\theta) + log \, f(x_i, y_i) + \phi(\theta)^T u(x_i, y_i) >_{q(\theta)}) \tag{3.49}$$

Because of the fact that we consider the logarithm of an exponential model, it is possible to average parameters w.r.t. $q(\theta)$ obtaining:

$$q(X) = \frac{1}{Z_X} [\prod_{i=1}^{n} f(x_i, y_i)] \, exp(\sum_{i=1}^{n} \bar{\phi}(\theta)^T u(x_i, y_i)) \tag{3.50}$$

In [18], the so called parameter inversions is considered i.e. if $\phi(\theta)$ is invertible, it is possible to find a parameter set $\tilde{\theta}$ such as $\bar{\phi} = \phi(\tilde{\theta})$. This is actually an averaging effect of the "ensemble" learning; in other words a parameter set is equivalent to the parameter set of ensemble of models defined by the probability distribution $q(\theta)$. The VBE step can be rewritten in this case as:

$$q(X) = \frac{1}{Z_X} \prod_{i=1}^{n} p(x_i, y_i | \tilde{\theta}, m) \tag{3.51}$$

$\tilde{\theta}$ is known as the *variational point estimate* ([52],[18]). Variational distribution $q(x_i)$ still belong to the CE family; it means that multiple VBEM iterations are possible keeping the same structure.

The VBM step can be equivalently obtained by direct substitution and taking advantage of the fact that prior and posterior distributions have the same form:

$$\begin{aligned} q(\theta) &= \frac{1}{Z_(\theta)} exp(< log\, p(X, Y|\theta, m) >_{q(X)})\, p(\theta|m) = \\ &= h(\tilde{\eta}, \tilde{\nu}) g(\theta)^{\tilde{\eta}} exp(\phi(\theta)^T \tilde{\nu}) \end{aligned} \tag{3.52}$$

with updated hyperparameters:

$$\begin{aligned} \tilde{\eta} &= \eta + n && (3.53) \\ \tilde{\nu} &= \nu + \sum_{i=1}^{n} \bar{u}(y_i) && (3.54) \\ \bar{u}(y_i) &= < u(x_i, y_i) >_{q(x_i)} && (3.55) \\ h(\tilde{\eta}, \tilde{\nu}) &= \frac{1}{Z_\theta} exp(\sum_{i=1}^{n} < log\, f(x_i, y_i) >_{q(x)}) && (3.56) \end{aligned}$$

It is important to notice that after each iteration the new $q(\theta)$ and $q(X)$ still belong to the exponential family so that the updated formulas are actually valid at a generic iteration of the VBEM algorithm.

The variational point estimation allows a straightforward comparison between the MAP and the VB. In fact the MAP uses current parameter estimation in the E step , while VB uses parameters obtained by averaging over current possible models with parameter distribution $q(\theta)$. MAP and VB have actually the same complexity but the variational point estimation cannot always be obtained in a convenient way. In fact it relies on the invertibility of the function $\phi(.)$ and on the unicity of the inversion solution. This can be easily understood if we consider $\phi^{-1}[< \phi >_{q(\theta)}]$ where $\phi$ and $\theta$ have different dimension; in this case the inversion may not be well determined. For models like HMM natural parameter inversion is a straightforward procedure but for other models like Linear Dynamical Systems ([18]) the problem is more flagrant. Currently methods for automated algorithm derivation have been proposed in the VB framework in order to overcome those difficulties (see [27],[25]).

## 3.4 Variational Bayesian Model Selection

Variational Bayesian methods operate on a lower bound of the log marginal likelihood. In section 2.6.2 we have described the model selection properties of the marginal likelihood that embeds the Occam factor. Occam factor penalizes more complex models that overfit the training data and offer poor generalization.

As long as Variational free energy $F_m$ is a bound over the log marginal likelihood, it is reasonable to expect that it has some model selection properties. In this section we will detail the mathematical background that motivates this property and in the next chapter of this work we will make intensive use of Variational Bayesian model selection.

### 3.4.1 Ensemble learning

In the original formulation of the ensemble learning ([5]) using the notation in [103], a variational bound over an ensemble of models can be obtained i.e.

$$L = log\, p(Y) = \sum_m \int d\theta\, dX\, p(Y, X, \theta, m) \geq$$
$$\sum_m \int d\theta\, dX\, q(X, \theta, m)\, log\, \frac{p(Y, X, \theta, m)}{q(X, \theta, m)} = F \tag{3.57}$$

where with $m$ we designate the model index and with $q(X, \theta, m)$ we designate the joint variational distribution over hidden variables, parameters and model. Expression 3.57 comes again from Jensen inequality and it is true for a generic $q(X, \theta, m)$ distribution. In order to find the best $q(X, \theta, m)$ we make the hypothesis of factorization over the three quantities $X$,$\theta$ and $m$.

$$q(X, \theta, m) = q(m)\, q(X|m)\, q(\theta|m) \tag{3.58}$$

Rewriting expression 3.57 we obtain:

$$\begin{aligned} F &= \sum_m q(m)[\int dX\, d\theta\, q(X|m)\, q(\theta|m)\, log\, \frac{p(Y, X, \theta|m)}{q(X|m)\, q(\theta|m)} + log\, \frac{p(m)}{q(m)}] = \\ &= \sum_m q(m)[F_m + log\, \frac{p(m)}{q(m)}] \end{aligned} \tag{3.59}$$

where with $F_m$ and $p(m)$ we designate the free energy of model and the prior distribution over model $m$. Optimal variational distributions $q(\theta|m)$ and $q(X|m)$ can be found using the VBEM algorithm. In order to find the optimal variational distributions $q(m)$ over models it is enough to solve $\partial F / \partial\, q(m) = 0$ under the constraint $\sum_m q(m) = 1$ enforced by the use of a Lagrange multiplier.

$$\begin{aligned} & \frac{\partial}{\partial q(m)} \sum_m q(m)[F_m + log\,\frac{p(m)}{q(m)}] + \lambda[\sum_m q(m) - 1] = \\ & F_m + log\, p(m) - log\, q(m) - 1 + \lambda = 0 \;\; \forall m \end{aligned} \tag{3.60}$$

Enforcing the Lagrange multiplier condition and solving w.r.t. $q(m)$ we obtain:

$$q(m) = \frac{1}{Z_m} exp(F_m)\, p(m) \tag{3.61}$$

where $Z_m$ is the normalization constant:

$$Z_m = \sum_m exp(F_m)\, p(m) \tag{3.62}$$

In other words the best variational posterior for a model $m$ is proportional to the exponential of the model free energy times the prior over the model. If no prior information is available over the model, the best model is the model with the highest free energy. This is a very appealing results because it shows that in the variational approximated framework the free energy has the same model selection properties as the log marginal likelihood. Furthermore in VB learning the free energy has the double property of objective function and model scoring function.

### 3.4.2 Free energy and Occam factor

If the free energy can be used to make model selection, it must have a penalty term embedded somewhere, a sort of Occam factor as defined in section 2.6.2. The penalty term is evident if we rewrite the free energy as follows:

$$F_m = F_{likelihood} - D_{penalty} \tag{3.63}$$

$$F_{likelihood} = \int dX\, d\theta q(\theta) q(X)\, log\, \frac{p(Y, X, |\theta)}{q(X)} \tag{3.64}$$

$$D_{penalty} = D(q(\theta)||p(\theta)) \tag{3.65}$$

The first term in 3.63 ($F_{likelihood}$) is the likelihood of complete data likelihood computed using variational distributions $q(X)$ and $q(\theta)$, while the second term $D_{penalty}$ is the KL divergence between the variational distributions $q(\theta)$ and prior distributions over parameters $p(\theta)$. By definition $D(a||b) \geq 0$ with equality when $a = b$; so $D(q(\theta)||p(\theta))$ is always positive and acts like a penalty term. In fact it will be larger for model with a larger number of parameters. Contrarily to the BIC it does not simply considers the parameter number but it consider the divergence between prior distribution and the posterior (approximated) distributions. This is exactly the same principle of the Occam razor that considers the prior and posterior accessible parameters volume i.e. $\sigma_{w|Y}/\sigma_w$ (see section 2.6.2).

Unsurprisingly in large data limit, the BIC is recovered also from the free energy penalty term (see [5]). Let us give an intuitive explanation as in [18] considering separately the likelihood term and the penalty term.

Concerning the likelihood term $F_{likelihood}$, we can imagine that when $n \rightarrow \infty$ ($n$ is the size of the data set), the mass of the variational distribution is concentrated around the MAP estimation i.e. $q(\theta) \rightarrow \delta(\theta - \theta_{MAP})$. Let us consider then the E-step that becomes:

$$log\, q(X) \propto \int \delta(\theta - \theta_{MAP})\, log\, p(X, Y|\theta) = log\, p(X, Y|\theta_{MAP}) \tag{3.66}$$

As consequence the limit of the likelihood term will give the likelihood computed on the MAP parameter estimation i.e.

$$lim_{n\rightarrow\infty}\, F_{likelihood} = log\, p(Y|\theta_{MAP}) \tag{3.67}$$

Let us consider now the penalty term $D_{penalty}$ (i.e. expression (3.65))limiting our investigation to the case of $q(\theta)$ of a conjugate-exponential form. In this case it can be shown that exponential family distribution results in asymptotic normality (see [18]). Assuming in the large data limit a Gaussian form for $q(\theta)$, it is possible to write the following limit:

$$\begin{aligned} lim_{n\rightarrow\infty} D_{penalty} &= lim_{n\rightarrow\infty}[\int q(\theta)\, log\, p(\theta) + \frac{d}{2}\, log\, 2\pi - \frac{1}{2} log\, |H|] = \\ &= -\frac{d}{2}\, log\, n + O(1) \end{aligned} \tag{3.68}$$

Assumption made here are similar to the assumption made in the BIC section (see 2.6.4): just terms that grow with $n$ are kept. Finally we can write the asymptotic limit of the complete free energy as:

$$lim_{n\rightarrow\infty} F_m = log\, p(Y|\theta_{MAP}) - \frac{d}{2}\, log\, n = BIC(Y, n) \tag{3.69}$$

that is the Bayesian information criterion for model $m$. The case in which $q(\theta)$ is not in the conjugate exponential family is not considered here.

### 3.4.3 A bias problem

Free energy has a mathematical foundation as model selection criterion even if it is just an approximation for the real (often uncomputable) log marginal likelihood. This approximation can introduce a bias in the decision about the best model. Let us reconsider the difference between the free energy and the marginal likelihood for a model $m$:

$$log\, p(Y|m) = log \int dX\, d\theta\, p(Y, X, \theta|m) = F_m + D(q(X, \theta)_m || p(X, \theta|Y, m)) \tag{3.70}$$

Comparing two models $m_1$ and $m_2$ exactly means comparing $log\, p(Y|m_1)$ and $log\, p(Y|m_2)$ (under the hypothesis of uniform prior distributions otherwise priors must be considered too) i.e.

$$\begin{aligned} log\, p(Y|m_1) - log\, p(Y|m_2) = F_{m_1} - F_{m_2} + \\ D(q(X, \theta)_{m_1} || p(X, \theta|Y, m_1)) - D(q(X, \theta)_{m_2} || p(X, \theta|Y, m_2) \neq F_{m_1} - F_{m_2} \end{aligned} \tag{3.71}$$

The difference between comparing marginal likelihood and comparing free energy is $D(q(X, \theta)_{m_1} || p(X, \theta|Y, m_1)) - D(q(X, \theta)_{m_2} || p(X, \theta|Y, m_2)$. As long as $q(X, \theta)_{m_1}$ and $q(X, \theta)_{m_2}$ are extremely different the bound may became more or less accurate. Predicting the accurateness of the bound is not a trivial task.

Let us consider a simple example with two GMM with $M$ and $M + 1$ components. In order to simplify the problem, let us assume that each component contributes to the term $D(q(X, \theta)_M || p(X, \theta|Y, M))$ with of a fixed amount $D_M$ (this is again an approximation) in order to have $D(q(X, \theta)_M || p(X, \theta|Y, M)) = M\, D_M$ and $D(q(X, \theta)_{M+1} || p(X, \theta|Y, M + 1)) = (M + 1)\, D_M$. Comparing the two models will result in:

$$\begin{aligned} log\, p(Y|M + 1) - log\, p(Y|M) &= F_{M+1} - F_M + (M + 1)\, D_M - M\, D_M = \\ &= F_{M+1} - F_M + D_M \neq F_{M+1} - F_M \end{aligned} \tag{3.72}$$

Looking at 3.72 it is easy to understand that when $F_{M+1} - F_M$ is used $D_M$ is neglected introducing a bias towards simpler models. This phenomena has been experimentally verified in [18] comparing the VB model selection with Bayesian model selection obtained using Monte-Carlo methods.

## 3.5 Online learning

In real data problems when the amount of data is extremely important, an increase in hidden variable number is generally consequent (we repeat that hidden variable number increases with the number of available observations). In those cases it is extremely useful to derive online algorithms that contrarily to the batch algorithms update parameters (or distributions) for each instance of the observations at a time.

Expectation-Maximization can be implemented in an online (incremental) fashion as proposed in [103]. In Bayesian learning, when the M-step passes through augmented hyperparameters, the incremental version of the learning algorithm is even more appealing because it can be easily interpreted.

In the context of Variational Bayesian learning online VBEM has been considered in [50],[118]. Here we will revise the basic concept of online learning as in [18].

Let us limit the discussion again to conjugate-exponential models with hyperparameters $\lambda = \nu, \eta$ and let us divide the data set in batches of size $N^b$ where $b$ designates the batch. The idea is to pass the augmented hyperparameters of batch $b$ as prior hyperparameters of batch $b+1$. Mathematically speaking in the E-step the algorithm will compute $N^b$ hidden variables $X^b$ for data in the batch $b$ and hyperparameters are updated:

$$\begin{aligned} \bar{\eta} &= \eta^{b-1} + N^b & (3.73) \\ \bar{\nu} &= \nu^{b-1} + \sum_{y_i \,\epsilon\, b} \bar{u}(y_i) & (3.74) \end{aligned}$$

where with $y_i \,\epsilon\, b$ we designate data $y_i$ that belongs to batch $b$. Once convergence is achieved in one batch $b$, it is enough for the next batch to initialize hyperparameters with results from the previous batch i.e. $\eta^b = \bar{\eta}$ and $\nu^b = \bar{\nu}$.

Intuitively all information about batches processed up to a certain moment is contained in the current estimation of $\bar{\eta}, \bar{\nu}$ and does not depend on previous hidden variables values or previous hyperparameters estimation.

Because of the huge quantity of data in experimental part of this work we will make an intensive use of online algorithms.

## 3.6 Variational Bayesian prediction

One of the goal of machine learning is building a model for making inference on some unseen data. For example in speech recognition the inference consists in making recognition on a test set different from the training set used for learning the model. Even if in the experimental part of this thesis (chapters 6 and 7) we focus on clustering and no prediction is used, for sake of completeness we address the problem of prediction in the Variational Bayesian framework. The prediction task consists in computing the probability of an unseen data set $Y_U$ given some training data $Y$ and a model $m$ i.e.

$$p(Y_U|Y, m) = \int d\theta \, p(Y_U|\theta, m) \, p(\theta|Y, m) \qquad (3.75)$$

or eventually averaging over all possible models:

$$p(Y_U|Y) = \sum_m \int d\theta \, p(Y_U|\theta, m) \, p(\theta, m|Y) \qquad (3.76)$$

Eventually expressions 3.75 and 3.76 can be easily extended to the case of hidden variables. As long as posterior distributions are not always exactly computable a first approximation consists in substituting them with variational posterior distributions i.e. $p(\theta|m,Y) \approx q(\theta|m)$ and $p(\theta,m|Y) \approx q(\theta,m)$ that results into:

$$p(Y_U|Y,m) \approx \int d\theta\, p(Y_U|\theta,m)\, q(\theta|m) \tag{3.77}$$

$$p(Y_U|Y,m) \approx \sum_m \int d\theta\, p(Y_U|\theta,m)\, q(\theta,m) \tag{3.78}$$

For some models expressions 3.77 and 3.78 are tractable and can produce straightforward Bayesian prediction (e.g. GMM). Anyway in some other models like HMM they still produce intractable forms. In those cases a further convex approximation based on Jensen inequality must be considered (for further details see section 4.2.4).

Another rough and simple method to approximate 3.75 and 3.76 is simply to consider parameter first modes (a.k.a. means) that we will denote with $\theta_{MVB}$ as in [94] where likelihood computed on $\theta_{MVB}$ is obtained as:

$$p(Y_U|Y,m) \approx p(Y_U|\theta_{MVB},m) \tag{3.79}$$

$$p(Y_U|Y,m) \approx \sum_m p(Y_U|\theta_{MVB},m) \tag{3.80}$$

Even if this is a rough approximation, it can still give interesting results (see [18],[94]).

## 3.7 VB in machine learning

In recent years the use of Variational methods (Bayesian and not) has been applied to huge number of models. From an historical point of view it was first applied in [57] in the case of a one-hidden layer neural network (with hidden variables) with a variational distribution constrained to be a Gaussian with diagonal covariance matrix. In [13] the case of full covariance matrix is considered.

The term *ensemble learning* appears in [103] in the sense of fitting an ensemble of models.

Since then VB methods have found their way in many different machine learning problems. For example in [26] a VB algorithm for Principal Component Analysis is described; in [142] the case of VB learning of mixture of experts is considered; in [51] the case of a Bayesian mixtures of factor analyzers is studied in the light of VB framework. In [53] approximated inference on switching state-space models is proposed. In [94] free form optimization with HMM is proposed for the first time and in [6] generic inference in graphical models with hidden variables is considered.

## 3.8 VB in speech processing

Even if Variational Bayesian methods are relatively recent, they have already been applied to a large number of problems related to audio and speech processing.

A first type of application of VB algorithms is solving problems that cannot be solved with classical exact techniques like the EM and require approximated methods.

For example the denoising and dereverberation problem can be formulated in a probabilistic framework of Bayes-optimal signal estimation (see [10]). This framework can be efficiently treated using a VB approach to solve the problem in an approximated way. In this case the intractability comes from the definition of a model that considers clean speech as a hidden variable to be estimated (because corrupted by noise). As previously discussed, the use of hidden variables in Bayesian approaches leads to an impossibility of closed form solution. This approach has been applied both to single microphone and microphone array (see [8],[9],[7]).

In the same family of variational algorithm we can consider the ALGOQUIN algorithm (see [79],[80],[44]). ALGOQUIN is a probabilistic method for considering noisy speech as the result of clean speech, noise and channel effect. The joint posterior distribution between those terms cannot be considered in an exact form and variational algorithms are used to determine their approximated posterior distributions. The spirit of the ALGOQUIN algorithm is very close to the one in the work of [10] because both of them introduce intractable probabilistic framework on the speech production taking into consideration different factors (noise, channel , reverberation) and solve them with approximated algorithms.

VB framework has been used also into models like the switching state space models (SSM) as a method for efficient inference in an extremely computational expensive task (see [87],[86]). In this case the main issue is the computational complexity; in fact, extending the SSM to a switching method significantly increases the computational overload. The use of variational methods simplifies the inference in this kind of models.

Many other works are devoted to the comparison of classical learning methods like MAP and ML with the VB criterion in order to avoid many common problems like overfitting of the data or mismatches between models and data. [125] and [128] consider the VB-GMM for speech data outlying the highest capacity of VB to match the data to model under different conditions like an extremely huge number of Gaussians and shows that the recognition results can improve when VB is used instead of ML.

Application of VB to continuous speech recognition systems has been treated in [137]. In this work and in successive publications (see [138],[139]) authors introduce the Variational Bayesian Estimation and Clustering (VBEC) algorithm. In VBEC framework a shared state triphone clustering in a speech recognition system is built using VB for estimation and for model selection. Comparison with a ML/MDL based system is done. VB outperforms the ML/MDL system when the amount of training data is poor; on the other side when the amount of data increases performances of two systems are similar.

The Bayesian method is often used in order to determine the optimal Gaussian number in large vocabulary speech system [141] and the robustness of the obtained model w.r.t. different factors [140]. In [69] and [68] VB is applied in order to obtain sofisticated Successive-State-Splitting (SSS) algorithms: the topology and the optimal SSS are determined applying the Bayesian property of the VB learning.

A comparison of ML, MAP and VB on large vocabulary speech recognition has been studied in [126]. In this work it seems that VB sometimes oversmooths posterior distribution estimation resulting in a small loss of accuracy.

Application to speech recognition is relatively satisfactory because this is by definition a discriminative task and VB (like MAP and ML) is a generative procedure thus results cannot in any case compete with those coming from discriminative learning methods like MCE or MMI.

On the other side, when the problem is merely formulated as a generative task, VB approach shows interesting improvements on the classical techniques. In [133] and [129] we studied the application of VB learning and model selection to a generative task as the speaker clustering. In [130] we studied and compared the VB with a BIC/ML system while in [131] we studied the use of a background model as prior information in the clustering task comparing with a BIC/MAP system. In this kind of generative tasks that need efficient model selection the VB interestingly outperforms the ML and the MAP.

Other interesting application of VB in the context of feature transformation and feature selection can be found in [82] where Variational Bayesian Principal Component Analysis is applied to space feature reduction for speech recognition inferring the optimal subspace dimension for each phoneme. We have derived a VB formulation of the feature saliency problem for speech recognition (see [132]) and for audio type classification ([134]) and compared it with the original method proposed in [85] i.e. the MML (Minimum Message Length).

As final remark Variational Bayesian modeling has also been applied to speech synthesis problems (see [146]).

# Chapter 4

# Variational Bayesian Learning for GMM and HMM

In this chapter we consider the two most important models in speech processing i.e. Gaussian Mixture Models and Hidden Markov Models under the light of the Variational Bayesian framework. In chapter 2 we have discussed in details the Maximum Likelihood and the Maximum a Posteriori learning for GMM and HMM. After discussing general VB learning in chapter 3, we apply here the VBEM to the two most important models in audio processing methods. The state of the art systems in speech recognition uses HMM with GMM as density function for emission probability (see [144]). HMM/GMM training and decoding have been achieved using both generative and discriminative methods. ML learning has been studied in [109], MAP learning in [46]. Discriminative methods can be grouped into two groups: Minimum Classification Error (MCE) methods (see [75]) and Maximum Mutual information (MMI) methods (see [106]). As long as in this work we consider just generative models we will not give details about discriminative methods for HMM/GMM.

The variational Bayesian solution to the HMM problem was first proposed by D.J.C. MacKay in [94] while VB GMM was studied by H. Attias in [6],[5] and in [34]. We reconsider here both solutions because they will be the core of VB applications of chapters 6 and 7.

## 4.1 VB for GMM

Let us consider again a Gaussian Mixture model with $M$ components and parameters $\theta = \{c_i, \mu_i, \Sigma_i\}$:

$$p(y_n) = \sum_{i=1}^{M} p(y_n|x_n = i, \theta)p(x_n = i|\theta) = \sum_{i=1}^{M} c_i \, N(y_n|\mu_i, \Sigma_i) \tag{4.1}$$

with $Y = \{y_1, \dots y_N\}$ i.i.d. samples and $X = \{x_1, \dots, x_N\}$ hidden variables. The natural choice for parameter prior distributions is the same as in MAP learning i.e.:

$$\begin{aligned} p(\theta) &= p(\{c_i\}, \{\mu_i\}, \{\Sigma_i\}) = p(\{c_i\}) \prod_i p(\mu_i|\Sigma_i) p(\Sigma_i) && (4.2) \\ p(\{c_i\}) &= Dir(\lambda_0) && (4.3) \\ p(\mu_i|\Sigma_i) &= N(\rho_0, \xi_0 \Sigma_i) && (4.4) \\ p(\Sigma_i) &= W(a_0, B_0) && (4.5) \end{aligned}$$

in order to obtain a conjugate-exponential model where $Dir()$, $N()$ and $W()$ designate a Dirichlet, Normal and Wishart distribution and $\lambda_0, \rho_0, \xi_0, a_0, B_0$ are distribution hyperparameters.

According to the previous discussions on conjugate-exponential models (see sections 2.3.2 and 3.3 variational posterior distributions have the same form of prior distributions with new hyperparameters i.e.:

$$\begin{aligned} q(\theta) &= q(\{c_i\}, \{\mu_i\}, \{\Sigma_i\}) = q(\{c_i\}) \prod_i q(\mu_i|\Sigma_i) q(\Sigma_i) && (4.6) \\ q(\{c_i\}) &= Dir(\lambda_i) && (4.7) \\ q(\mu_i|\Sigma_i) &= N(\rho_i, \xi_i \Sigma_i) && (4.8) \\ q(\Sigma_i) &= W(a_i, B_i) && (4.9) \end{aligned}$$

where $\lambda_i, \rho_i, \xi_i, a_i, B_i$ are updated hyperparameters. It is important to notice that factorization (4.6) *is not* an hypothesis but it is a consequence of prior distribution factorization (4.2) and of the M-step of the VBEM algorithm (i.e. (3.18)).

### 4.1.1 VBEM for GMM

The application of VBEM algorithm in this case is straightforward. Let us designate as usually with $q(\theta)$ and $q(X)$ variational distributions that we suppose independent for tractability issues. Concerning parameter distribution $q(\theta)$ according to priors (4.3 - 4.5) we can factorize as $q(\theta) = q(\{c_i\}) \prod_i q(\mu_i|\Sigma_i) q(\Sigma_i)$.

First hyperparameters $\lambda_i, \rho_i, \xi_i, a_i, B_i$ must be initialized. The VB E-step can be obtained simply applying 3.16 resulting in $\gamma_i^n = q(x_n = i|y_n)$ i.e.

$$\gamma_i^n = q(x_n = i|y_n) = \frac{1}{Z(x_n)} \tilde{c}_i \, \tilde{\Sigma}_i^{1/2} \, exp(-(y_n - \rho_i)^T \bar{\Sigma}_i (y_n - \rho_i)/2) \, exp(-\frac{d}{2\xi_i}) \qquad (4.10)$$

$$\begin{aligned} Z(x_n) &= \sum_i q(x_n = i|y_n) && (4.11) \\ q(X) &= \prod_n q(x_n = i|y_n) && (4.12) \end{aligned}$$

where the ensemble parameters are obtained integrating w.r.t. $q(\theta)$:

$$\begin{aligned}
log\, \tilde{c}_i &= < log\, c_i >= \Psi(\lambda_i) - \Psi(\sum_i \lambda_i) && (4.13)\\
log\, \tilde{\Sigma}_i &= < log|\Sigma_i| >= \sum_{z=1}^{d} \Psi((a_i + 1 - z)/2) - log\,|B_s| + d\, log\, 2 && (4.14)\\
\bar{\Sigma}_i &= < \Sigma >_i = a_i B_i^{-1} && (4.15)
\end{aligned}$$

where $\Psi = d\, log\, \Gamma(x)/dx$ is the digamma function ([1]) and $d$ is the dimension of the observation vector.

The VBM step is similar to the MAP M-step because of the fact the model is actually a conjugate-exponential model. First the following quantities are computed:

$$\begin{aligned}
\gamma_i &= \sum_t^T \gamma_i^t && (4.16)\\
\bar{\omega}_i &= \frac{\sum_{t=1}^T \gamma_i^t\, y_t}{\sum_{t=1}^T \gamma_i^t} && (4.17)\\
\bar{r}_i &= \sum_{t=1}^T \gamma_i^t\, (y_t - \bar{\mu}_i)(y_t - \bar{\mu}_i)^T && (4.18)
\end{aligned}$$

then variational posterior distribution hyperparameters are updated as follows:

$$\begin{aligned}
\lambda_i &= \gamma_i + \lambda_0 && (4.19)\\
a_i &= \gamma_i + a_0 && (4.20)\\
\xi_i &= \gamma_i + \xi_0 && (4.21)\\
\rho_i &= \frac{\gamma_i\, \bar{\omega}_i + \xi_0\, \rho_0}{\gamma_i + \xi_0} && (4.22)\\
B_i &= B_0 + \bar{r}_i + \gamma_i \xi_0 (\bar{\omega}_i - \rho_0)(\bar{\omega}_i - \rho_0)^T/(\gamma_i + \xi_0) && (4.23)
\end{aligned}$$

### 4.1.2 Free energy computation for GMM

Free energy is a key quantity for model selection purposes. In the case of Variational Bayesian GMM it can be easily computed as follows:

$$\begin{aligned}
F &= \sum_{i=1}^{M} \sum_{n=1}^{N} q(x_n) \int \int q(\Sigma_i)\, q(\mu_i|\Sigma_i)\, log\, p(y_n|\Sigma_n, \mu_n, x_n) d\Sigma_i\, d\mu_i + \\
&+ \sum_{i=1}^{M} \sum_{n=1}^{N} q(x_n) \int q(c)\, log\, \frac{p(x_n|c)}{q(x_n)} dc + \sum_{i=1}^{M} \int q(\mu_i|\Sigma_i)\, log\, \frac{p(\mu_i|\Sigma_i)}{q(\mu_i|\Sigma_i)} d\mu_i + \\
&+ \sum_{i=1}^{M} \int q(\Sigma_i)\, log\, \frac{p(\Sigma_i)}{q(\Sigma_i)} d\Sigma_i + \int q(c) log\, \frac{p(c)}{q(c)} dc && (4.24)
\end{aligned}$$

Rewriting terms related to distance between prior and posterior distributions (i.e. last three terms in (4.24)) it is possible to obtain:

$$\begin{aligned} F &= -D_{Dir}(\lambda||\lambda_0) - \sum_{i=1}^{M} D_{Wishart}(a_i, B_i||a_0, B_0) \\ &- \sum_{i=1}^{M} D_{Normal}(\rho_i, B_i/(\xi_i a_i)||\rho_0, B_i/(\xi_0 a_i)) + \sum_{i=1}^{M} F(i) \qquad (4.25) \\ F(i) &= \gamma_i \, log \, \tilde{c}_i - \sum_{n=1}^{N} \gamma_i^n \, log \, \gamma_i^n + \frac{\gamma_i}{2}(-d \, log \, 2\pi + log\tilde{\Sigma}_i - Tr(\bar{\Sigma}_i \bar{r}_i/\gamma_i) - \frac{d}{\xi_i}) \\ & \qquad (4.26) \end{aligned}$$

For the KL divergence of Dirichlet, Normal and Wishart distributions see appendix A.

### 4.1.3 VB prediction for GMM

When prediction using VB posterior distributions is used, it turns out to be of a tractable form for GMM. It is a well known result that integrating a Gaussian distribution under Normal-Wishart distribution a t-student distributions.

In fact given an unseen data set $Y_U$:

$$p(Y_U|Y) = \int d\theta p(Y_U|\theta)p(\theta|Y) \approx \int d\theta p(Y_U|\theta) \, q(\theta) =$$

$$\int [\sum_{i=1}^{M} c_i N(Y_U|\mu_i, \Sigma_i) q(\{c_i\}) q(\mu_i|\Sigma_i) q(\Sigma_i)] = \sum_{i=1}^{M} \pi_i \, t_{\omega_i}(Y_U|\rho_i, \Lambda_i) \qquad (4.27)$$

where $t_{\omega_i}$ designate a t-student distribution with $\omega_i = a_i + 1 - d$ degree of freedom, mean $\rho_i$, covariance $\Lambda_i = ((\xi_i + 1)/\xi_i \omega_i) B_i$ and weights $\pi_i = \lambda_i / \sum_i \lambda_i$. The t-student probability density function is defined as:

$$\begin{aligned} t_{\omega_i}(Y_U|\rho_i, \Lambda_i) &= \frac{\Gamma(\frac{\omega_i}{2} + \frac{d}{2})|\Lambda_i|^{d/2}}{\Gamma(\omega_i/2)(\omega_i \pi)^{d/2}}(1 + \frac{\Delta^2}{\omega_i})^{-\frac{(\omega_i + d)}{2}} \qquad (4.28) \\ \Delta^2 &= (y - \rho_i)^T \Lambda_i (y - \rho_i) \qquad (4.29) \end{aligned}$$

In large data limit i.e. $N \rightarrow \infty$ the t-student mixture reduces to a GMM.

On the other side as described in [24], using a t-stud distribution instead of a Gaussian distribution have some advantage in case of poor available data because t-student distribution is more heavy tailed respect to Gaussian distribution. Eventually VB learning can be done directly on the t-stud distribution parameters instead of passing by a GMM and then inferring prediction parameters (see [24]).

Unfortunately many models do not admit the same simple solution for Bayesian prediction (and for VB prediction). In those cases variational approximation must be considered again.

### 4.1.4 Empirical priors for GMM

In this section we address the problem of determining empirical prior in the case of a GMM. In section 3.1.3 we described the general problem of hyperparameters estimation. In the GMM case, step 3.23 consists in the initialization and step 3.24 is the VBEM algorithm described in section 4.1. Let us concentrate on the optimization step i.e. (3.25) when the free energy is expressed as in (4.24). Once optimal posterior distribution are estimated (i.e. the variational distributions), the only term dependent on prior distributions is the KL divergence between priors and posteriors. Thus in order to maximize the free energy w.r.t. hyperparameters it is enough to minimize the distance between prior distributions and variational posteriors i.e.

$$argmax_{\lambda} F_m(\bar{q}(X), \bar{q}(\theta), Y, \lambda) = argmin_{\lambda}\, KL(q(\theta, \bar{\lambda})||p(\theta, \lambda)) \tag{4.30}$$

Because the KL divergence can be written as sum of KL divergences for each independent parameter distributions the optimization can separately work on:

1

$$D_{Dir}(\lambda||\lambda_0) \tag{4.31}$$

2

$$\sum_{i=1}^{M} D_{Normal}(\rho_i, B_i/(\xi_i a_i)||\rho_0, B_i/(\xi_0 a_i)) \tag{4.32}$$

3

$$\sum_{i=1}^{M} D_{Wishart}(a_i, B_i||a_0, B_0) \tag{4.33}$$

Let us consider separately the three divergences.

**First term**: $D_{Dir}(\lambda||\lambda_0)$. Let us derive the (4.31) w.r.t. $\lambda_0$ and equate to zero:

$$\begin{aligned} \frac{\partial}{\partial \lambda_0} D_{Dir}(\lambda||\lambda_0) &= \frac{\partial}{\partial \lambda_0} log\,\Gamma(M\,\lambda_0) - M log\,\Gamma(\lambda_0) + (\lambda_0 - 1)\sum_{i=1}^{M}[\Psi(\lambda_i) - \Psi(\sum \lambda_i)] = \\ &= M\Psi(M\,\lambda_0) - M\Psi(\lambda_0) - \sum_{i=1}^{M}[\Psi(\lambda_i) - \Psi(\sum \lambda_i)] = 0 \end{aligned} \tag{4.34}$$

To solve this equation a gradient based method as the Newton-Raphson method can be used. Furthermore the solution is a maximum because the second derivative w.r.t. $\lambda_0$ is negative for $\lambda_0 > 0$.

**Second term**: $\sum_{i=1}^{M} D_{Normal}(\rho_i, B_i/(\xi_i a_i)||\rho_0, B_i/(\xi_0 a_i))$. As before let us derivate (4.33) w.r.t. $\xi_0$ and $\rho_0$ and equate to zero.

$$\begin{aligned} D &= \sum_{i=1}^{M} D_{Normal}(\rho_i, B_i/(\xi_i a_i)||\rho_0, B_i/(\xi_0 a_i)) = \\ &= \sum_{i=1}^{M} \int N(\rho_i, B_i/\xi_i a_i)[log(B_i/\xi_0 a_i) - (\rho_i - \rho_0)^2 \cdot B_i/\xi_0 a_i] \end{aligned} \tag{4.35}$$

$$\frac{\partial D}{\partial \rho_i} = 0 \Rightarrow \rho_i = \frac{1}{M}\sum_{i=1}^{M} < \mu_i >_{q(\mu_i)} = \frac{1}{M}\sum_{i=1}^{M} \rho_i \tag{4.36}$$

$$\frac{\partial D}{\partial \xi_0} = 0 \Rightarrow \xi_0^{-1} = \frac{1}{M}\sum_{i} < (\rho_i - \rho_0) \cdot (\rho_i - \rho_0) > a_i/B_i \tag{4.37}$$

**Third term**: $\sum_{i=1}^{M} D_{Wishart}(a_i, B_i||a_0, B_0)$. As before let us derivate (4.33) w.r.t. $a_0$ and $B_0$ and equate to zero.

$$D = \sum_{i=1}^{M} D_{Wishart}(a_i, B_i||a_0, B_0) \tag{4.38}$$

$$\frac{\partial D}{\partial a_0} = \sum_{i=1}^{M}[\sum_{d=1}^{D} \Psi(a_0 + 1 - l) - 2ln2 + lnB_0 - < log\Sigma_i >] = 0 \tag{4.39}$$

$$\frac{\partial D}{\partial B_0} = \sum_{i=1}^{M}[a_0 B_0^{-1} - < \Sigma_i >] = 0 \tag{4.40}$$

The system composed by equations (4.39) and (4.40) fixes the first moment and the first log-moment of the prior distribution to the average of the same quantities obtained averaging the variational posteriors. To solve this system numerical methods must be used.

## 4.2 Variational Bayesian Hidden Markov Models

The very first work on VBHMM was proposed by MacKay in [94] where a procedure equivalent to the Baum-Welch was derived in order to optimize the free energy of an HMM. In this section we consider those results and finally we try to answer to the three fundamental questions on HMM exposed in section 2.2.2.

### 4.2.1 HMM Variational Free Energy

Considering the same notation as in section 2.2.2, we designate with $Y = \{y_1, \dots, y_t, \dots, y_T\}$ the observation sequence, with $S = \{s_1, \dots, s_t, \dots, s_T\}$ the state sequence and with $\theta = \{A, B, \pi\}$ the HMM parameters (we consider here the discrete probability state emission). Prior distributions are set to independent Dirichlet distributions:

$$\begin{aligned} p(\theta) &= p(\pi)\prod_{ij} p(a_{ij}) \prod_{ik} p(b_{ik}) && (4.41) \\ p(\pi) &= Dir(\lambda_\pi) && (4.42) \\ p(a_{ij}) &= Dir(\lambda_{a_{ij}}) && (4.43) \\ p(b_{ik}) &= Dir(\lambda_{b_{ik}}) && (4.44) \end{aligned}$$

Let us introduce the variational distributions over parameters and hidden variables i.e. state sequence $S$: $q(A, B, \pi, S)$. The variational distribution approximate the intractable exact posterior $p(A, B, \pi, S|Y)$. In this form the problem is not tractable so the independence hypothesis is considered i.e. $q(A, B, \pi, S) = q(A, B, \pi)q(S)$. The following form for the free energy is obtained applying Jensen inequality.

$$\begin{gathered} log\, p(Y) = log \int d\pi \int dA \int dB \sum_S p(Y, S|\pi, A, B) p(\pi, A, B) \geq \\ \geq \int d\pi \int dA \int dB \sum_S q(\pi, A, B, S) log \frac{p(Y, S|\pi, A, B)p(\pi, A, B)}{q(\pi, A, B, S)} = \\ \geq \int d\pi \int dA \int dB \sum_S q(\pi, A, B) q(S) log \frac{p(Y, S|\pi, A, B)p(\pi, A, B)}{q(\pi, A, B)q(S)} = \\ \int d\pi \int dA \int dB q(\pi, A, B) [\sum_S q(S) log \frac{p(Y, S|\pi, A, B)}{q(S)} + log \frac{p(\pi, A, B)}{q(\pi, A, B)}] = F_{HMM} \end{gathered} \tag{4.45}$$

We find again in 4.45 two terms: the first one is the likelihood of the data and the second one is the penalty term that penalizes more complex models.

Let us rewrite $p(Y, S|\theta, A, B)$ as:

$$log\, p(Y, S|\pi, A, B) = \pi_{s_1} [\prod_{t=1}^{T} a_{s_t s_{t+1}}][\prod_{t=1}^{T} b_{s_t y_t}] \tag{4.46}$$

We can now rewrite the free energy as:

$$\begin{aligned} F_{HMM} &= \int d\pi \sum_S q(S)\pi_{s_1} + \int dA \sum_S q(S) \sum_{t=1}^{T} a_{s_t s_{t+1}} + \int dB \sum_S q(S) \sum_{t=1}^{T} b_{s_t y_t} \\ &- \sum_S q(S) log\, Q(S) + \int d\pi dA dB q(\pi, A, B) log \frac{p(\pi, A, B)}{q(\pi, A, B)} \qquad (4.47) \\ & \qquad (4.48) \end{aligned}$$

### 4.2.2 VBEM for HMM

Because the HMM with prior distributions defined before is a model that belongs to the Conjugate-Exponential family the Variational Bayesian EM algorithm can be applied in order to find optimal distributions $q(\theta)$ and $q(S)$. An optimization algorithm was derived by MacKay [94] inspiring successively the natural parameter inversion proposed in [18].

Let us consider the E-step and derive the free energy (4.45) w.r.t. the distribution over state sequence $q(S)$:

$$\frac{\partial F_{HMM}}{\partial q(S)} = \int d\pi \int dA \int dB q(\pi, A, B)\, p(Y, S|\theta, A, B) - log\, q(S) + const = 0 \qquad (4.49)$$

Substituting now the sequence (4.46) into the partial derivate, we can find again the independence between parameters in the first derivate.

$$\begin{aligned} log\, q(S) = \int d\pi q(\pi)\pi_{s_1} + \int dA q(A) log\, a_{s_t s_{t+1}} + \int dB q(B) b_{s_t y_t} - log\, Z(S) = \\ < log\, \pi_{s1} >_{q(\pi)} + < log\, a_{s_t s_{t+1}} >_{q(A)} + < log\, b_{s_t y_t} >_{q(B)} - log\, Z(S) \end{aligned} \qquad (4.50)$$

We have added here a normalization constant $Z(S)$ that should results from a constrained optimization problem enforcing the constraint $\sum_S Q(S) = 1$; in section 4.2.3, we show how $Z(S)$ can be obtained directly from the backward-forward algorithm.

The intuition in [94] consists in observing that the sequence $q(S)$ can be rewritten in terms of a modified parameter set:

$$(\tilde{\pi}, \tilde{A}, \tilde{B}) = (exp < log\, \pi >_{q(\pi)}, exp < log\, A >_{q(A)}, exp < log\, B >_{q(B)}) \qquad (4.51)$$

$$q(S) = \frac{1}{Z(S)} \tilde{\pi}_{s_1} [\prod_{t=1}^{T} \tilde{a}_{s_t s_{t+1}}][\prod_{t=1}^{T} \tilde{b}_{s_t y_t}] \qquad (4.52)$$

Under a Dirichlet prior expression 4.51 can be computed using the expectation of the log Dirichlet distribution i.e.

$$\int d\pi \, Dir(\pi|\lambda) = \Psi_{\lambda_i} - \Psi(\sum_i \lambda_i) \tag{4.53}$$

In order to compute sufficient statistic for the HMM the forward-backward recursion can be again applied using parameters 4.51. The only difference is that now those quantities are sub-normalized distributions and a normalization constant must be set for normalization purposes. Let us define $\tilde{\alpha}$ and $\tilde{\beta}$ recursion terms analogous to $\alpha$ and $\beta$ in section 2.2.2:

$$\begin{aligned} \tilde{\alpha}_t(i) &= \tilde{p}(Y_1^t, s_t = i|\tilde{\theta}) = [\sum_{i=1}^{N} \tilde{\alpha}_{t-1}(i)\tilde{a}_{s_i s_j}]\tilde{b}_{jy_t} && (4.54) \\ \tilde{\beta}_t(i) &= p(Y_{t+1}^T|q_t = i, \tilde{\theta}) = [\sum_{j=1}^{N} \tilde{a}_{s_i s_j}\tilde{b}_{jy_{t+1}}\tilde{\beta}_{t+1}(j)] && (4.55) \end{aligned}$$

and equivalent sufficient statistics $\tilde{\gamma}_t(i,j)$

$$\tilde{\gamma}_t(i,j) = \tilde{p}(s_{t-1} = i, s_t = j|Y_1^T, \theta) = \frac{\tilde{\alpha}_{t-1}\tilde{a}_{ij}\tilde{b}_j(y_t)\tilde{\beta}_t(j)}{\sum_{k=1}^{N}\tilde{\alpha}_T(k)} \tag{4.56}$$

Let us consider now the VBM step. Deriving free energy (4.47) w.r.t. $q(\pi, A, B)$ and solving, we obtain:

$$\begin{aligned} log\, q(\pi, A, B) &= [\int q(S) log\, p(Y, S|\pi, A, B)] + log\, p(\pi, A, B) = \\ &= \int q(S) log\, p(s_1|\pi) + \int q(S) log\, p(s_{2:t}|s_1, A) + \int q(S) log\, p(Y|S, B) + \\ &+ \ log\, p(\pi) + log\, p(A) + log\, p(B) && (4.57) \end{aligned}$$

From 4.57, it is easy to understand that posterior variational distributions over parameters $q(A, B, \pi)$ results so automatically factorized into terms whose optimization is independent from each other. In other words simply using the Markovian hypothesis and the prior independence hypothesis we found out that variational distributions over parameters are independent in between them; this is not an ulterior hypothesis but it is a consequence of the model. To summarize for the VB approach, it is possible to write without any assumption:

$$q(\theta) = q(A, B, \pi) = q(A)q(B)q(\pi) \tag{4.58}$$

Because of the fact HMM with Dirichlet priors belongs to the conjugate exponential model we are sure that posterior distributions have the same form of prior with updated hyperparameters.

$$
\begin{aligned}
q(\bar{\pi}) &= Dir(\bar{\lambda}_{\pi_i}),\ \ \bar{\lambda}_{\pi_i} = \lambda_{\pi_0} + \sum_{j}^{N} \tilde{\gamma}_0(i,j) && (4.59)\\
q(\bar{a}_{ij}) &= Dir(\bar{\lambda}_{a_{ij}}),\ \ \bar{\lambda}_{a_{ij}} = \lambda_{a_{ij}} + \sum_{t=1}^{T} \tilde{\gamma}_t(i,j) && (4.60)\\
q(\bar{b}_{ik}) &= Dir(\bar{\lambda}_{b_{ik}}),\ \ \bar{\lambda}_{b_{ik}} = \lambda_{b_{ik}} + \sum_{t}^{T} \sum_{i} \tilde{\gamma}_t(i,j) && (4.61)
\end{aligned}
$$

Contrarily to the MAP algorithm the VB does not explicitly estimate any parameter but just evaluate the posterior distributions over parameters. In this case there are no reparameterization problems as in formula (2.84).

### Natural parameter inversion for HMM

Results of the previous section can be obtained applying the natural parameter inversion ([18]). In fact expression (4.51) can be interpreted in terms of natural parameters $\phi(\theta)$ and expected natural parameters i.e.

$$
\begin{aligned}
\theta &= \{\pi, A, B\} && (4.62)\\
\phi(\theta) &= \{log\,\pi, log\,A, log\,B\} && (4.63)\\
\bar{\phi} &= < \phi(\theta) >_{q(\theta)} = \{< log\,\pi >_{q(\pi)}, < log\,A >_{q(A)}, < log\,B >_{q(B)}\} && (4.64)
\end{aligned}
$$

In order to obtain the Variational point estimation $\tilde{\theta}$ such as $\bar{\phi} = \phi(\tilde{\theta})$ it is enough to use the inverse of $\phi$. In HMM the situation is very convenient because $\phi = log(.)$ so the inverse is simply the exponential function.

$$
\tilde{\theta} = \phi^{-1}(< \phi_\theta >_{q(\theta)}) = (exp < log\,\pi >_{q(\pi)}, exp < log\,A >_{q(A)}, exp < log\,B >_{q(B)}) \qquad (4.65)
$$

It is evident that results (4.65) and (4.51) are the same. In this case the inversion is straightforward and the Variational point estimation can be easily obtained. This comes from the fact that in HMM parameters are independent. This is not always the case in other models belonging to the CE family where the inversion is not a simple task.

## 4.2.3 Free energy computation for HMM

In order to evaluate expression 4.45 a brute force computation of all terms is not necessary; instead an implementation trick can be used. Let us consider again the free energy in the HMM case and write it as in [43] i.e.:

$$F_{HMM} = \int dAq(A)log\frac{p(A)}{q(A)} + \int dBq(B)log\frac{p(B)}{q(B)} + \int d\pi q(\pi)log\frac{p(\pi)}{q(\pi)} \\ +H(q(S)) + \int dA\,dB\,d\pi dS\,q(A)q(B)q(\pi)\,log\,p(Y,S|A,B,\pi) \quad (4.66)$$

where $H(q(S))$ denotes the entropy of variational distribution over the hidden variables. Let us rewrite entropy using expression (4.50):

$$\begin{aligned} H(q(S)) &= -\sum_S q(S)log\,q(S) = \\ &= -\sum_{q(S)} q(S)[\int d\pi\,dA\,dB\,log\,p(Y,S|\pi,A,B) - log\,Z(S)] \\ &= -\sum_{q(S)} q(S)[\int d\pi\,dA\,dB\,log\,p(Y,S|\pi,A,B)] + log\,Z(S) \quad (4.67) \end{aligned}$$

Substituting expression (4.67) into (4.66), we obtain the following:

$$\begin{aligned} F_{HMM} &= \int dAq(A)log\frac{p(A)}{q(A)} + \int dBq(B)log\frac{p(B)}{q(B)} + \int d\pi q(\pi)log\frac{p(\pi)}{q(\pi)} \\ &+ \; log\,Z(S) \quad (4.68) \end{aligned}$$

The likelihood term simplifies between the two expressions and the free energy reduces to the KL divergences between prior parameter distributions and variational distributions plus the normalization constant for the state sequence. $Z(S)$ can be directly obtained after the forward-backward recursion as:

$$log\,Z(S) = \sum_t Z(\tilde{\alpha}(s_t)) \;\; with \;\; Z(\tilde{\alpha}(s_t)) = \sum_S \tilde{\alpha}(s_t) \quad (4.69)$$

$Z(\tilde{\alpha}(s_t))$ is the normalization constant for the $\tilde{\alpha}$ in the forward algorithm.

It is straightforward to notice the analogy with expression (2.25) for computing the probability of an observation sequence in ML HMM in which the sum of successive $\alpha$ is used.

### 4.2.4 VB prediction for HMM

Let us consider prediction of an unseen sequence $Y_U$ given a training sequence $Y$. As described in section 3.6, posterior distributions can be approximated by variational posterior distributions i.e. $p(\theta|Y) \approx q(\theta)$.

Contrarily to the GMM case this does not result in a tractable integral for a HMM. We are obliged again to approximate the integral with a lower bound applying Jensen inequality as follows:

$$p(Y_U|Y) \approx \int d\theta q(\theta)p(Y_U|\theta) \geq exp \int d\theta\, q(\theta)\, log \sum_S p(Y,S|\theta)$$
$$\geq exp \int d\theta q(\theta) \sum_S q(S) log \frac{p(Y,S|\theta)}{q(S)} \qquad (4.70)$$

In this case, Jensen inequality has been applied two times in order to obtain a bound with variational distributions over parameters and hidden variables (i.e. state sequence). Expression (4.70) can be estimated as proposed in section (4.2.3) with the test sequence instead of the training sequence. So given parameter variational distributions $q(\theta)$, the forward algorithm can be used to determine $log\, Z(S)$ and finally compute the bound.

### 4.2.5 Summary

To conclude this section on HMM we reformulate the three fundamental questions about Hidden Markov Models ([109]) in the light of variational Bayesian learning.

Question 1 "Given an observation sequence Y and a parameter set $\theta$ find $p(Y|\theta)$" must be reformulated in Bayesian terms as "Given an observation sequence Y and a parameter distribution $q(\theta)$ find $\int q(\theta)p(Y|\theta)$". As long as the integral is intractable, it must be bounded as in expression (4.70). To evaluate the bound a modification of the forward algorithm that uses Variational point estimation instead of parameters is derived; the analogy between the VB forward algorithm and the classical forward algorithm is extremely strong anyway: the forward algorithm operates on the exact probability $p(Y|\theta)$ while the VB algorithm operates just on a bound.

Question 2 Concerning the best state sequence given an observation sequence $Y$ and distributions $q(\theta)$, the VB forward algorithm can be easily transformed into a Viterbi like algorithm considering the max instead of the sum all over possible states. Again this algorithm works on a bound and not on the exact probability distributions.

Question 3 The maximum likelihood estimation of parameters $\theta$ in a Bayesian framework is substituted by the estimation of posterior distributions $p(\theta|Y)$. As long as optimal posterior distributions cannot be found unless numerical methods are used the variation bound is used as objective function. Parameter learning can be done with a modification of the Baum-Welch algorithm that consider Variational point estimates as parameters.

### 4.2.6 Empirical priors for HMM

The problem of empirical priors for HMM is analogous to the one described in section 4.1.4 for the GMM case. When emission probabilities are discrete and their prior distribution is a Dirichlet distribution optimal hyperparameters can be found as in section 4.1.4 in the case of mixture coefficients. Again the problem consists in simply making prior distributions as close as possible to the variational posterior distributions. The explicit computation is omitted for brevity.

## 4.3 VB for HMM with GMM emission probabilities

Let us consider in this section the unification of results of section 4.1 and 4.2 in an HMM with emission probabilities represented by GMM.

Parameter set is now simply extended with $\theta = \{\pi, A, G\}$ where $G = \{c_{s_ik}, \mu_{s_ik}, \Sigma_{s_ik}\}$ are weights,means and covariance matrix of the $k$th Gaussian component for the $i$th state.

Two kind of hidden variables must be considered: the state sequence $S = \{s_1, \dots, s_T\}$ and the Gaussian component $X_{s_i} = \{x_{1\,s_i}, \dots, x_{M\,s_i}\}$. Variational distributions factorizes as follows: $q(\pi, X, S) = q(\pi)q(S)q(X|S) = q(\pi)q(S)\prod_i q(X_{s_i}|s_i)$ (obviously the Gaussian component hidden variables are conditioned on the state sequence).

### 4.3.1 HMM/GMM Free energy

The free energy for a HMM/GMM can be simply written as combination of free energies for the HMM and the set of GMM that models the emission probability. Combining notations of section 4.2 and 4.1 it is possible to write:

$$\begin{aligned} F_{HMM/GMM} = \int dA q(A) log\frac{p(A)}{q(A)} + \int d\pi q(\pi) log\frac{p(\pi)}{q(\pi)} + \\ + \sum_{s_i}\sum_{k} [q(c_{s_ik}) log\frac{p(c_{s_ik})}{q(c_{s_ik})} + q(\mu_{s_ik}) log\frac{p(\mu_{s_ik})}{q(\mu_{s_ik})} + q(\Sigma_{s_ik}) log\frac{p(\Sigma_{s_ik})}{q(\Sigma_{s_ik})}] \\ + H(q(S,X)) + \int dA\, dG\, d\pi\, dS\, dX\, q(A)q(\pi)q(G)q(S,X)\, log\, p(Y,S,X|A,G,\pi) \end{aligned} \tag{4.71}$$

In this case free energy (4.71) have to deal with the two sets of hidden variables $X$ and $S$. $H(q(S,X))$ is the entropy of the joint probability over $S$ and $X$.

Extension of $log\, p(Y,S,X|A,G,\pi)$ is straightforward.

$$log\, p(Y,S,X|A,G,\pi) = \pi_{s1} \prod_{t=1}^{T} a_{s_t s_{t+1}} \prod_{t=1}^{T} log\, p(y_t|s_t, x_{s_t}) =$$

$$= \pi_{s1} \prod_{t=1}^{T} a_{s_t s_{t+1}} \prod_{t=1}^{T} log\, N(y_t|\mu_{x_{s_t}}, \Sigma_{x_{s_t}}) \tag{4.72}$$

As before posterior distributions over parameters can be factorized as a consequence of the Markovian hypothesis and prior independence hypothesis:

$$q(A,\pi,G) = q(A)q(\pi)q(G) = q(A)q(\pi) \prod_{s_i} \prod_{k} q(c_{s_i k}) q(\mu_{s_i k}|\Sigma_{s_i k}) q(\Sigma_{s_i k}) \tag{4.73}$$

### 4.3.2 VBEM for HMM/GMM

The EM-like algorithm can be applied to the HMM/GMM again simply combining results from previous sections. In the E-like step derivatives w.r.t hidden variables variational distribution $q(X,S)$ is considered.

$$\frac{\partial F_{HMM/GMM}}{\partial q(X,S)} = \int q(A)q(G)q(\pi)q(S,X)\, log\, p(Y,S,X|A,G,\pi) - log\, q(X,S) + const \tag{4.74}$$

Solving w.r.t. $q(X,S)$, we obtain:

$$log\, q(X,S) = \int d\pi q(\pi) log\, \pi_{s_1} + \int dA q(A) log\, a_{s_t s_{t+1}} +$$

$$\int q(G) log\, N(y_t|\mu_{x_{s_t}}, \Sigma_{x_{s_t}}) - log\, Z(S,X) \tag{4.75}$$

where $Z(S,X)$ is a normalization constant. All expectations in 4.75 have already been computed: expectation of Dirichlet distributions can be computed using formula 4.53 and expectation for Gaussian distributions have the same form of equations (4.13-4.15).

Modification coming from the use of GMM instead of a discrete distribution for output probability results in the same factorized form for the hidden variable distributions as 4.52. This means that a forward-backward algorithm can be used to compute sufficient statistics. The algorithm must consider now hidden variables related with the Gaussian component too; this will simply results in a form similar to 2.41 that uses expected values computed w.r.t. variational distributions:

$$\tilde{\xi}_t(i,k) = \frac{\tilde{p}(Y, s_t = j, k_t = k)|\theta}{\tilde{p}(Y|\theta)} = \frac{\sum_{i=1}^{M} \tilde{\alpha}_{t-1}(i) \tilde{a}_{ij} \tilde{\beta}_{jk} \tilde{b}_{jk}(y_t) \tilde{\beta}_t(j)}{\sum_{i=1}^{N} \tilde{\alpha}_T(i)} \tag{4.76}$$

The M-step is simply obtained combining the M-step of the HMM i.e. 4.59 and 4.60 with the M-step of the GMM i.e. (4.7 - 4.9). Augmented hyperparameters are this time computed using sufficient statistics 4.76.

### 4.3.3 Free energy computation for HMM/GMM

Considerations of section 4.2.3 are still valid for free energy computation in the case of HMM/GMM. The free energy after some simple manipulation can be written as:

$$F_{HMM/GMM} = \int dAq(A)log\frac{p(A)}{q(A)} + \int d\pi q(\pi)log\frac{p(\pi)}{q(\pi)} +$$
$$\sum_{s_i}\sum_{k}[q(c_{s_ik})log\frac{p(c_{s_ik})}{q(c_{s_ik})} + q(\mu_{s_ik})log\frac{p(\mu_{s_ik})}{q(\mu_{s_ik})} + q(\Sigma_{s_ik})log\frac{p(\Sigma_{s_ik})}{q(\Sigma_{s_ik})}] + log\, Z(S,X) \tag{4.77}$$

Again it is basically constituted of two terms: a KL divergence term between prior and variational posterior distributions and the normalization constant of hidden variables sequence (states and Gaussian components).

As before $log\, Z(S,X)$ can be computed with the forward algorithm.

### 4.3.4 Prediction for HMM/GMM

For prediction purposes the real (unknown) posterior distribution over parameters is approximated with variational posterior distributions $p(\theta|Y) \approx q(\theta)$. In the case of GMM, as described in section 4.1.3 the predictive distribution results in a mixture of T-stud distributions. On the other side in HMM, predictive distribution can only be obtained as a lower bound as described in section 4.2.4.

To compute the prediction on unseen data in the HMM/GMM model we are obliged to extend approximation 4.70 in order to be able to handle hidden variables $X$ and $S$ i.e.

$$p(Y_U|Y) \approx \int d\theta q(\theta)p(Y_U|\theta) \geq exp\int d\theta\, q(\theta)\, log\sum_{S}p(Y,S,X|\theta)$$
$$\geq exp\int d\theta q(\theta)\sum_{S}q(S,X)log\frac{p(Y,S,X|\theta)}{q(S,X)} \tag{4.78}$$

As before expression 4.78 corresponds to the variational lower bound of the marginal likelihood of a test sequence given variational posterior distributions. It can be computed again using the forward algorithm using $\tilde{\xi}_t(i,k)$ as defined in 4.76.

The extension of the HMM/GMM in the variational case is basically analogous to the extension of the HMM/GMM in the ML case (see section 2.2.2).

## 4.4 Conclusion and summary

In this chapter we have applied the Variational Bayesian framework to Gaussian Mixture Models , Hidden Markov Models and the combination of HMM/GMM. We have answered to three question: how to write the free energy, how to learn the model and how to make inference.

In both cases the Variational learning results in an inexpensive modification of classical HMM and GMM learning procedures in which expectation over parameters are used. In HMM case, a modified forward-backward algorithm can be applied with variational point estimation for parameters instead of parameters.

# Chapter 5

# Preliminary experiments

In this chapter we analyze different behaviors of Maximum Likelihood (ML), Maximum a Posteriori (MAP) and Variational Bayesian (VB) learning in the framework of Gaussian Mixture Models. We focus our attention on GMM because it is the core of our speaker indexing systems (see chapter 7).

We will consider particularly three impact factors tightly related between them: the model size (that in case of GMM is defined by the component number), the amount of data available for learning and the impact of prior distributions (obviously this factor does not concern the ML framework). Experiments are developed in order to study the three algorithms in different combination of the impact factor.

The model size and the amount of data are strictly related in the sense that an amount of data can be poor or not, depending on the size of model that must be learned. For example 1K observations can be considered a reasonable amount of data for 2 components GMM and a poor amount of data for a 512 components GMM. In different experiments we will fix the model size and change the amount of data or vice versa we will fix the amount of data and progressively increase the components number. We will see that in extreme conditions VB performs definitely better than MAP and ML. Furthermore VB avoids overfitting problems typical of other approaches.

The choice of prior distributions is a more complicated problem. An interesting case study would be to use non-informative priors that bring no information to the learning process. Unfortunately those kinds of prior distributions are often improper distributions or raise intractable forms from a practical point of view (see section 2.3.1). The use of improper prior is possible as long as the posterior distribution is a proper distribution but in the following work we will stick to the case of proper prior distributions. The simple way to “simulate” non-informative distributions is to use very flat prior distributions. Anyway the idea of flat prior still depends on the kind of model and on the quantity of training data. This kind of prior distributions is also known in literature as broad priors, weak priors or small priors together with flat prior; we will refer indifferently to them in the following sections. Let us make a simple example of a Dirichlet prior distribution with prior hyperparameters $\lambda_0$ and with a data term $n_i$; the posterior hyperparameters is $\lambda_i = n_i + \lambda_0$. A "non-informative" prior can be achieved setting $\lambda_0 \ll n_i$ in the case $n_i$ is known or at least estimated. For instance if $n_i$ is of the order of $1M$, $\lambda_0 = 1K$ can be supposed not to bring much prior information while if $n_i$ is of the order of 1, a "non-informative" hyperparameter can be obtained $\lambda_0 \ll 1$. The actual value of $n_i$ is generally determined by the model size and the amount of data. Again in experiments, we will keep fix those two quantities and change the prior value.

Before going on with experiments, some general considerations must be done. It is well known that ML estimation can be considered as a special case of MAP estimation (apart from parameterization problems) when prior over parameters is non-informative; so we expect ML and MAP estimation to give the same results when broad priors are considered. On the other side, we showed that when variational distributions are extremely peaked, (in the limit case a delta distribution), MAP and VB coincide (again apart from parameterization problems). An extremely peaked variational distribution (i.e. $q(\theta)$) can result because of two factors:

- an extremely peaked prior distribution (i.e. $p(\theta)$)

- a huge quantity of data that make the posterior extremely peaked.

In the first case, we expect VB and MAP to have similar performances but very different way from ML. In fact the result is basically given by prior distributions. In the second case VB, MAP and ML will converge towards the same solution, in fact as we showed in chapter 3 when $N \to \infty$ where $N$ is the amount of training data, the three methods converge to the same solution (under some hypothesis on the MAP parameterization form).

To summarize, both MAP and ML are parameter estimation techniques that give the same results when prior distributions on parameters are small w.r.t. the log-likelihood of the data, that happens when priors are weak (in the limit case non-informative) or when an infinite amount of data is available (and the data term dominates the prior term). VB estimator converges to MAP and ML estimator when prior distributions are extremely peaked but when prior distributions are flat, VB estimator significantly differs from others because parameters are integrated out w.r.t. variational distributions.

In the following experiments, we aim at comparing ML,MAP and VB on the same GMM learning task. Anyway an important consideration must be done: MAP and ML are parameter estimation techniques while VB is a parameter distribution estimation technique. It means that using MAP and ML we can learn parameters and use them in the test, while using VB we will obtain parameter distributions. For testing purposes, VB prediction must be used and parameters must be integrated out; we already discussed in section 4.1.3 that integrating out a Normal distributions w.r.t. a conjugate Normal-Wishart prior results in a t-student distribution. For this reason, in the test concerning the VB framework a t-student distribution inferred by VB variational distributions (as defined in section 4.1.3) will be used. As previously outlined using a t-student distributions has many advantages compared to a Normal distribution because it is a heavy tailed distribution less sensitive to outliers (for an extended discussion on the advantages of VB methods for mixture of t-stud distributions see [24]).

Let us now consider a simple measure we will use in the sequel. We experimentally verified that VB makes clustering harder than other techniques (depending on its prior). In other words, other non-fully Bayesian techniques aim at using all parameters available in the model while VB tries to use just the "necessary" number. Let us make a simple example with a GMM; let us consider a case in which only few data are available for large number of components. We experimentally observe that the ML (or the MAP) tries to split the data all over the Gaussian components while VB clusters data in few clusters (consequence of the Occam razor principle). A simple way to quantify this behavior is using accumulator variances (we do not use mixture weights because VB does not estimate explicitly mixture weights). Accumulator mean is $1/M$ where $M$ is the number of Gaussian components and we expect variance to be small when data are split over components while variance to be high when data are concentrated in few clusters. Mathematically speaking if $\gamma_i^t$ designates the accumulator for the $i$th Gaussian estimated on the sample $y_t$, the accumulator variance is computed as:

$$A = \frac{1}{T}\sum_{t=1}^{T} 1/M \sum_{i=1}^{M} [\gamma_i^t - \frac{1}{M}]^2 \qquad (5.1)$$

For this reason, in following experiments we will estimate accumulator variance as a measure of how "hard" clustering is done.

The chapter is organized as follows: in section 5.1 we compare ML, MAP and VB in terms of performances as a function of the data for different prior settings. In a similar way in section 5.2 we study performances as a function of prior for different amount of data. In section 5.3 we study results as a function of the initial Gaussian component number. Prior optimization and model selection experiments on synthetic and real data are described in sections 5.4 and 5.5. Finally in section 5.6 we address the problem of model adaptation comparing MAP and VB.

## 5.1 Performance vs. Data for different priors

In this section we compare the three learning methods (VB,ML,MAP) on simple modeling task progressively increasing the amount of training data with different priors. Once training is done the log-likelihood is computed on a test set using the learned parameters or the t-student distribution (see section 4.1.3). It must be pointed out that those experiments do not aims at deriving any conclusion on a recognition task but only aim at studying the different behaviors of VB,MAP,ML trying to understand their differences.

In order to avoid local maxima problems that exist for all learning framework, we ran 5 times each EM (or VBEM) algorithm with 5 different initializations (the same for the three algorithms), computed 5 different test-log-likelihood and plotted the average.

We are equally interested in changing prior distributions from broad (non-informative) priors to strong peaked priors. Because we are not here interested in providing any prior information but just the strength of the prior distributions (we will consider later the case of empirical Bayes priors), we consider a fully tied prior with a relevance factor $\tau$. Mathematically speaking prior distributions are derived as follows :

$$\lambda_{\beta_{0i}} = \tau \tag{5.2}$$

$$\rho_{0i} = \bar{0} \tag{5.3}$$

$$\xi_{0i} = \tau \tag{5.4}$$

$$a_{0i} = \tau \tag{5.5}$$

$$B_{0i} = \tau\, I \tag{5.6}$$

where $\bar{0}$ is vector of zeros and $I$ is the identity matrix. The double index $0i$ designates prior hyperparameter for the $i$th Gaussian component. We fixed here the component number at 8. The amount of training data varies from 100 frames to about 2000 frames with a step of 20 frames. In this way we should move from a spare training data case to a non-spare training data case. In the same way different values of $\tau$ are considered from a range from weak priors to a highly peaked prior. We consider the following cases: $\tau = \{1E-6, 1, 10, 100, 1E+6\}$.

Figures (5.1-5.5) plot the test-log-likelihood w.r.t. the amount of training data for different values of $\tau$. Figures (5.6-5.10) plot accumulator variances (that give us informations about how "hard" the clustering is) for the different scenarios.

Let us analyze different results. In figure 5.1, a very small prior is used. First of all it can be noticed that ML and MAP give exactly the same result. Furthermore, when training data is spare VB gives a higher test log-likelihood compared to the two other. When the amount of training data increase, curves converges towards the same value. Figures 5.2-5.3 consider the case of informative priors ($\tau = \{1, 10\}$). In those cases, MAP and ML have different performances; MAP that gives better results than ML; on spare data VB still performs better than others, and again the three approaches converges for increasing amount of data. When $\tau = 100$, prior distributions are peaked enough to make VB and MAP have the same performance: in this case they both perform better than ML. Finally when prior is too peaked i.e. $\tau = 1E + 6$ in figure (5.5), MAP and VB still perform the same but data modify prior distribution very slowly resulting in poor performances.

Similar considerations can be done about the accumulator variance as a function of amount of data plotted in figures (5.6-5.10). MAP and ML have the same accumulator variance when priors are non-informative, while MAP and VB coincide when priors are strong. Anyway it can be noticed that when few data are available VB always makes harder clustering than ML and MAP. This can interpreted again as a result of the Occam razor principle in which when few data are available, VB tries to avoid overfitting and gives a more compact solution.

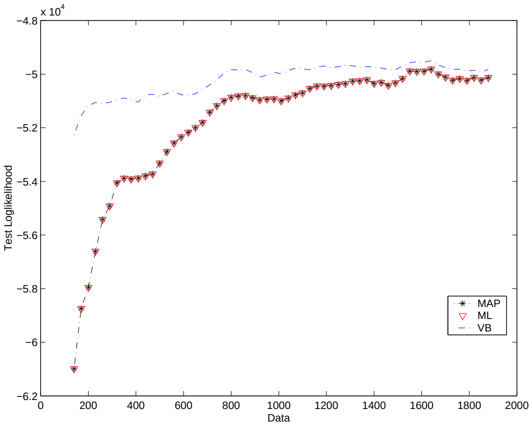


Figure 5.1: Test log likelihood for a 8 components GMM with non informative priors ($\tau = 1E - 6$ )

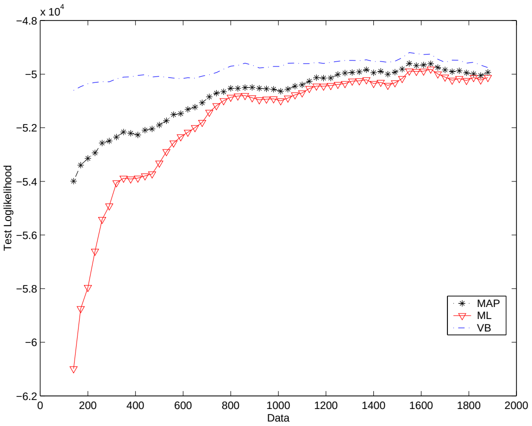


Figure 5.2: Test log likelihood for a 8 components GMM with non informative priors ($\tau = 1$ )

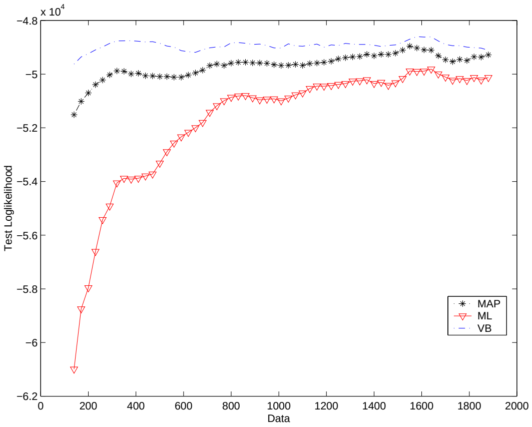


Figure 5.3: Test log likelihood for a 8 components GMM with non informative priors ($\tau = 10$ )

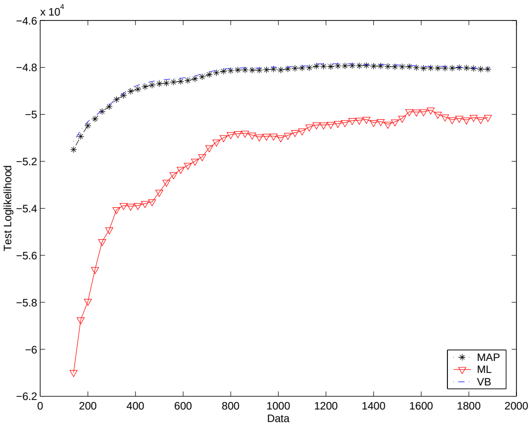


Figure 5.4: Test log likelihood for a 8 components GMM with non informative priors ($\tau = 100$ )

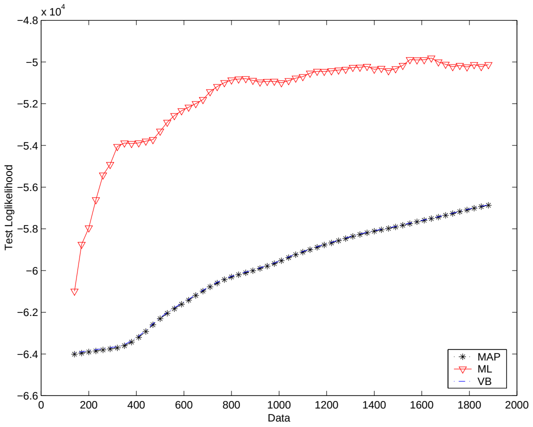


Figure 5.5: Test log likelihood for a 8 components GMM with non informative priors ($\tau = 10000$ )

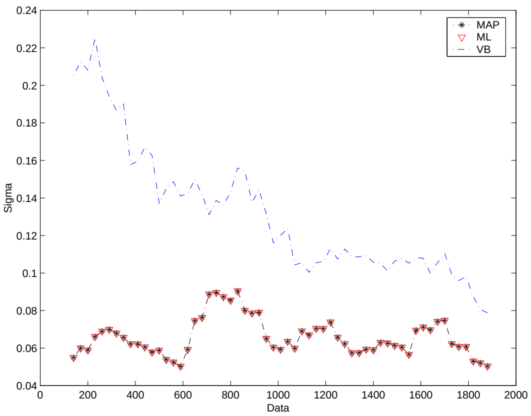


Figure 5.6: Accumulator for a 8 components GMM with non informative priors ( $\tau = 1E - 6$ )

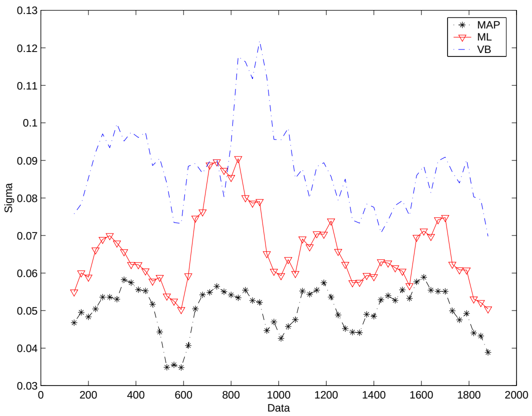


Figure 5.7: Accumulator for a 8 components GMM with non informative priors ( $\tau = 1$ )

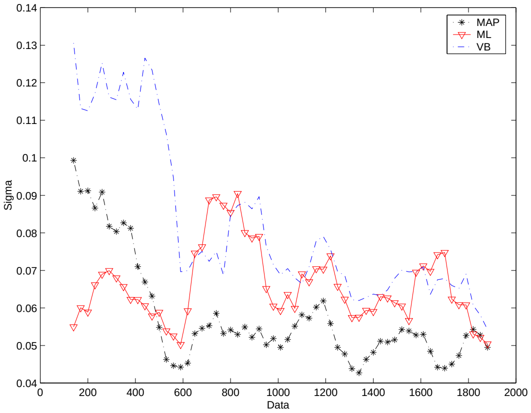


Figure 5.8: Accumulator for a 8 components GMM with non informative priors ( $\tau = 10$ )

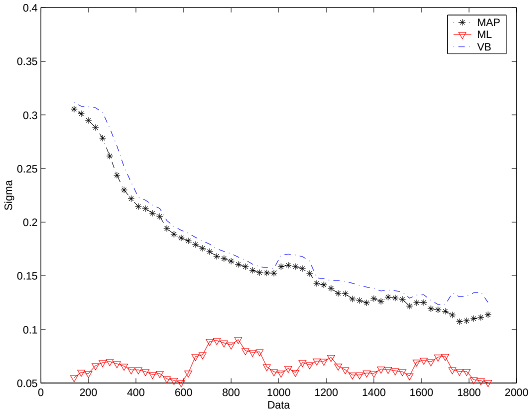


Figure 5.9: Accumulator for a 8 components GMM with non informative priors ( $\tau = 100$ )

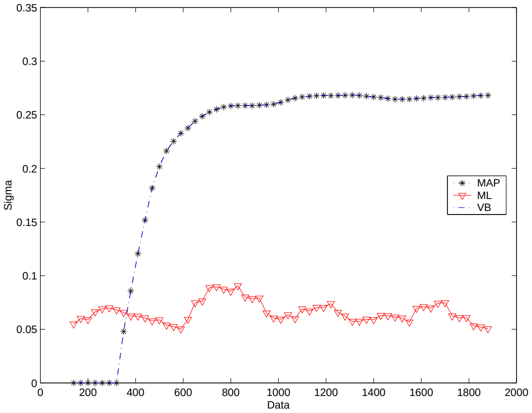


Figure 5.10: Accumulator for a 8 components GMM with non informative priors ( $\tau = 10000$ )

## 5.2 Performance vs. Priors with different data

In this section we compare in a similar way the three algorithms changing the prior distribution sharpness regulated by $\tau$ for different amount of data. The experimental framework is the same as in the previous section: an 8 Gaussian components GMM is used and to avoid local maxima, 5 different initializations are considered and finally the average value for test-log-likelihood is plotted. Values of $\tau$ ranges from 1E-6 to 1E+10. Test log-likelihood is plotted on a semilogarithmic scale w.r.t. $\tau$. We consider different amount of data $=\{20, 200, 2k, 20k, 200k\}$ frames for training. Figures (5.11-5.15) plots test log-likelihood w.r.t. $\tau$ for different amount of data while figures (5.16 - 5.20) plot accumulator variance for the same parameters. In all pictures ML estimation is evidently a constant because it does not depend on any prior distribution.

Let us analyze the different scenarios. An extremely interesting consideration can be done looking at the very extreme case of only 20 frames for the training (figures 5.11 and 5.16). The VB solution is extremely regularized even for small values of $\tau$ and better than the ML solution. On the other hand for small prior values the MAP solution is the worst solution. We would expect that the MAP and ML have the same performance but it does not happen here, this is a direct consequence of the MAP parameterization problems (see section 2.5). In the classical MAP basis representation, the lack of data generates drop of performances: a possible solution to this problem is finding another basis such that the estimation does not degrade that much e.g. a soft-max representation (see [95]).

Looking at accumulator variance in figure 5.16 we can notice that for small priors VB makes clustering less hard than ML. Again this is a consequence of Occam razor principle that does not make harder clustering but tries to make the right clustering i.e. when very few data are available and no prior information is available the clustering does not overfit the data but regularize the solution avoiding a model that strongly fit only the few data (in the same way in which it avoids model that strongly fit lot of data).

When the amount of data increases, we verify again that for small priors MAP and ML have the same performance while when prior becomes more important MAP and VB have the same performance. When prior becomes too strong performances degrade (not enough data to modify the prior). In figures 5.12,5.13,5.14, we can notice a peak in the performance; this peak actually corresponds to the best prior (that is actually unknown but can be somehow estimated). In all those cases VB performs better than MAP and ML for small prior and limited amount of training data. When very large amount of data is used (figure 5.15), the three techniques perform the same for small $\tau$ values and we observe performance degradation when prior becomes too strong.

Same considerations can be done looking at accumulator variance. We can notice a peak in the variance for an optimal value of prior (that's actually not at the same value as the peak in the log-likelihood figures because log-likelihood is computed on the test set and accumulator variances is computed on the train set). When prior becomes too big variance goes to zero. This is easily understandable because with large common priors over Gaussians all accumulators will have the value of $1/M$ where M is the number of components.

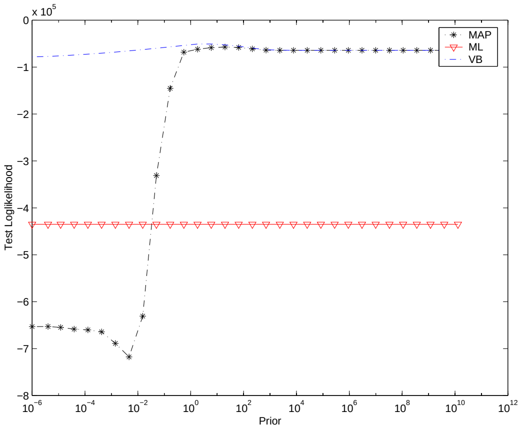


Figure 5.11: Test log likelihood for a 8 components GMM with (data = 20 )

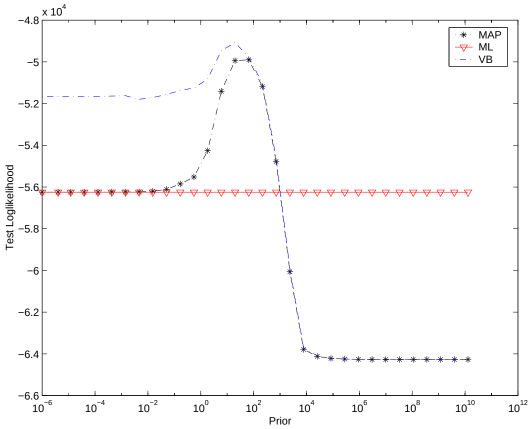


Figure 5.12: Test log likelihood for a 8 components GMM with (data = 200 )

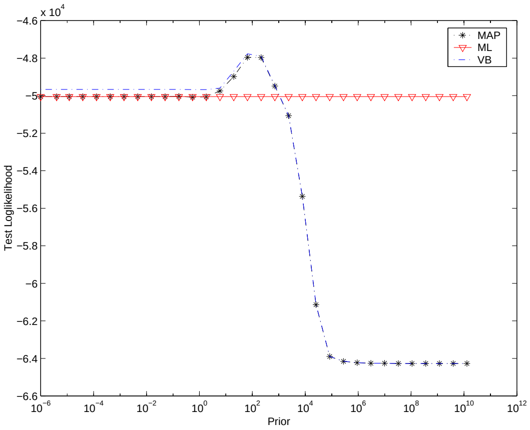


Figure 5.13: Test log likelihood for a 8 components GMM with (data = 2000 )

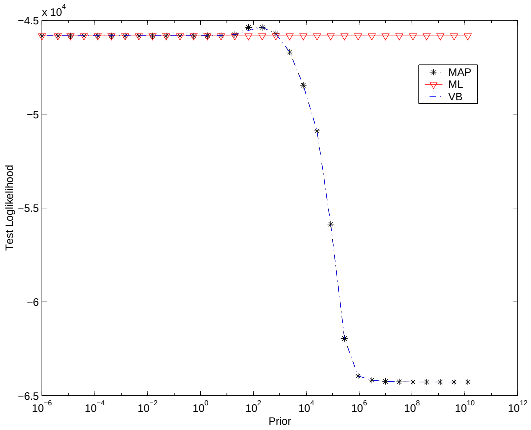


Figure 5.14: Test log likelihood for a 8 components GMM with (data = 20000 )

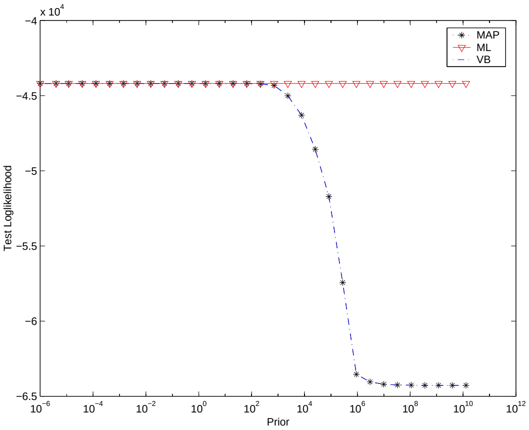
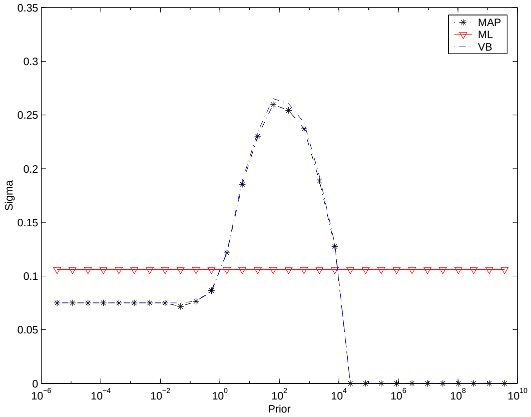


Figure 5.15: Test log likelihood for a 8 components GMM with (data = 50000 )



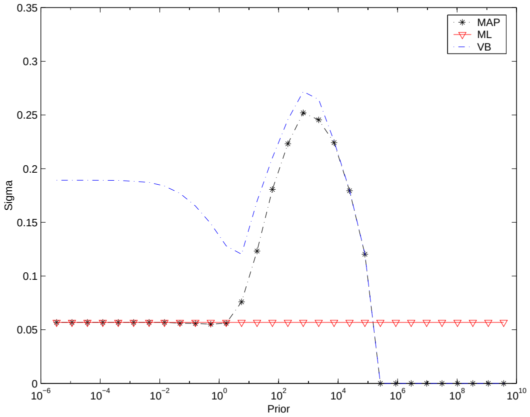
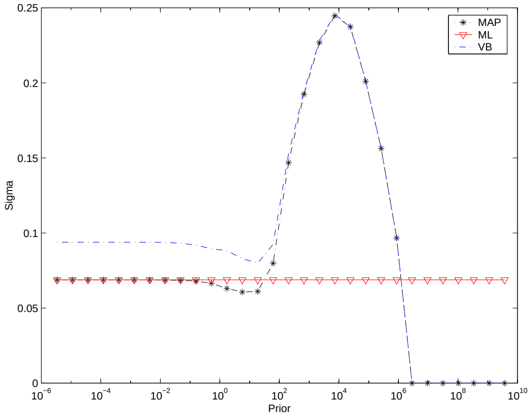


Figure 5.16: Accumulator for a 8 components GMM with (data = 20 )

Figure 5.17: Accumulator for a 8 components GMM with (data = 200 )



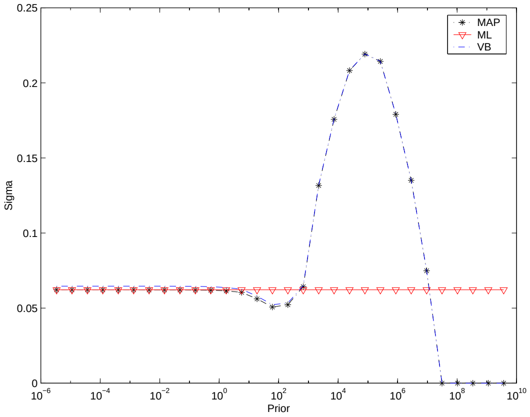


Figure 5.18: Accumulator for a 8 components GMM with (data = 2000 )

Figure 5.19: Accumulator for a 8 components GMM with (data = 20000 )

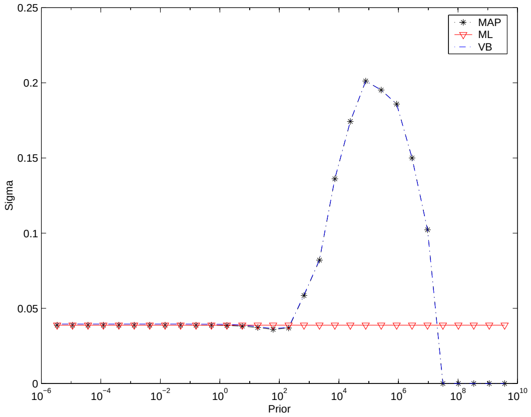


Figure 5.20: Accumulator for a 8 components GMM with (data = 50000 )

## 5.3 Performance vs. Initial number of Gaussian components for different priors

In this section we consider system performance with a variable initial number of Gaussian components with different priors. We fixed the amount of training data at 2k frames. Initial number of Gaussians moves from 10 to 250 by a step of 5. Multiple initialization is used as in previous sections. Three different values of $\tau = \{1E-6, 1, 100\}$ are used. We aim here to investigate the capacity of self pruning of MAP and VB and make a comparison with ML.

Let us consider the case in which a single vector is assigned to a Gaussian, in this case the ML algorithm will produce an infinite covariance; on the other hand VB and MAP will produce simply a null accumulator for the given Gaussian. MAP will then produce parameters equal to maxima of prior distributions, while VB will produce Gaussian parameter distributions equal to prior parameter distribution. In other words, data does not give any contribution to estimation of parameters that relies only on the prior information. When prior is not large, this typically results in the loss of the Gaussian component. Anyway because VB is a fully Bayesian technique, we expect to have a different behavior for different priors.

Increasing the Gaussian component number and keeping constant the amount of training data, we expect many components to be pruned from the MAP or the VB. In figures (5.21,5.23,5.25) the test log-likelihood w.r.t. initial number of Gaussian component is plotted for different values of prior $\tau$. In figures 5.22,5.24,5.26 the final Gaussian components after the training is plotted. When prior is broad, we can again notice that MAP and VB give the same result. Increasing the Gaussian component the test log-likelihood dramatically decreases for the MAP/ML learning while it is almost constant for the VB (figure 5.21). This can be explained looking at the final component number represented in figure 5.22. MAP and ML keep all initial components while VB prunes out components that are too weak resulting in an almost constant number of final components. This is an extremely useful property because, extra degrees of freedom do not give overfit problems because they are pruned out. Anyway there are some cases in which VB pruning can give a wrong pruning. In [96] an example of wrong pruning is described; in fact when the amount of data is extremely limited (in the limit case the train set may contain a single element) the symmetry break that generates the pruning (and the clustering itself) may degenerates in a completely wrong result (interested reader can find details in [96]).

When $\tau$ increases MAP performances diverge from ML and get closer to VB performance. In figure 5.23, $\tau = 1$ is used; priors are not negligible now. An increased performance results for both VB and MAP; now MAP better regularizes the solution. Figure 5.24 shows the final Gaussian component: VB component number is now increasing because of the regularization imposed by the stronger prior. Also MAP prunes some components, but in an softer way than VB.

When finally $\tau$ is 100 i.e. a strong prior MAP and VB have the same performance. Test log-likelihood decreases when the initial Gaussian component number increases (figure 5.25). Initial Gaussian component is preserved at the end of the training (figure 5.26). Anyway VB and MAP performances do not degrades drastically as the ML performances due to the regularization effect done by prior distributions and due to the averaging effect.

An interesting phenomenon is depicted in figure 5.26. Using a value of $\tau = 100$ both MAP and ML have the final component number equal to the initial one up to 250 components. On the other side the VB keep all components up to a value of about 230 and then starts pruning them out. This means that the high value of $\tau$ that prevents from pruning for few components becomes relatively less important when the component number increase.

To summarize, a useful property of the VB learning is pruning extra degrees of freedom that are not used during the training. This process is of course sensitive to prior values that can regularize or not the pruning process.

It is interesting to figure out what happens to the free energy when a degree of freedom is lost. Figure 5.27 plot the free energy value (left Y-axis) and the pruned component number (right X-axis) w.r.t. iteration number. Each time a component is pruned out there is a "jump" in the free energy; in fact each time a component accumulator is zero, its variational posterior distribution becomes identical to the prior distribution and their KL divergence is zero. The penalty term is so decreased and the free energy is increased; that explains the jump in the graph.

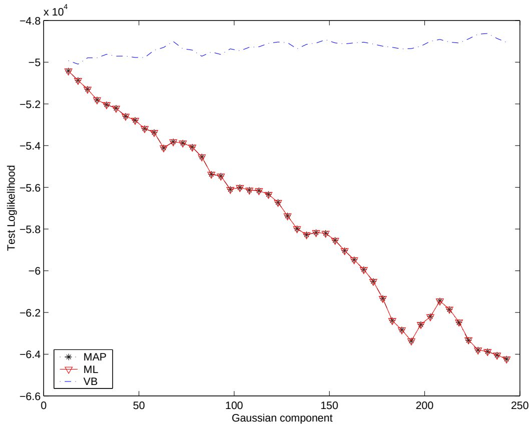


Figure 5.21: Test log likelihood for variable size GMM with (prior = 1E-7 )

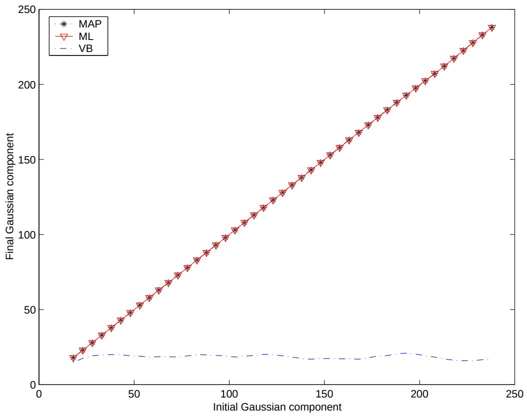


Figure 5.22: Final component (prior = 1E-7 )

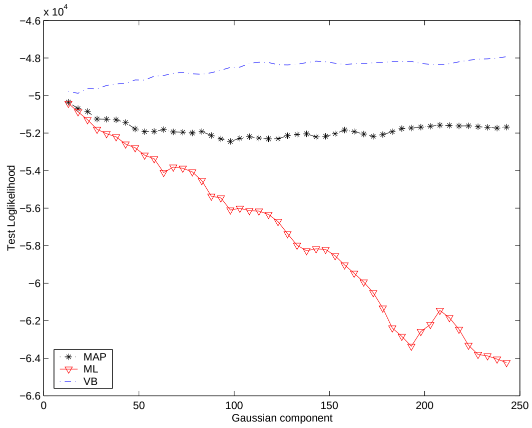


Figure 5.23: Test log likelihood for variable size GMM with (prior = 1 )

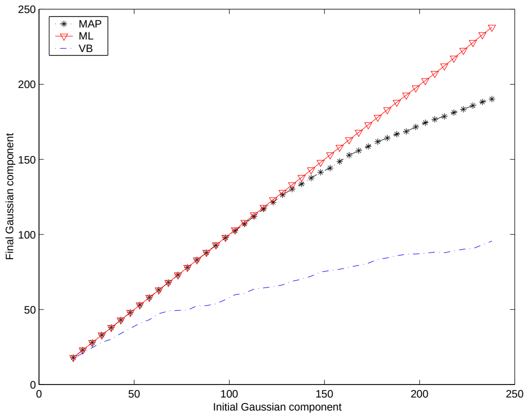


Figure 5.24: Final component (prior = 1 )

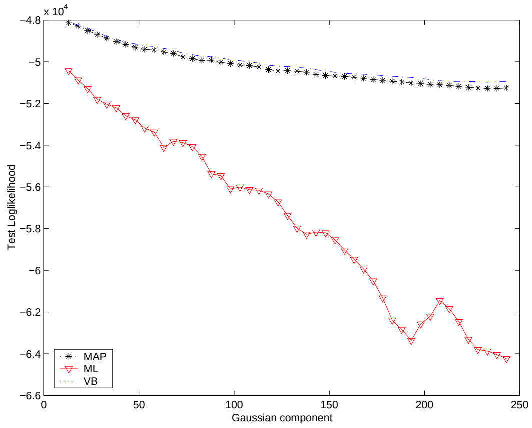


Figure 5.25: Test log likelihood for variable size GMM with (prior = 100 )

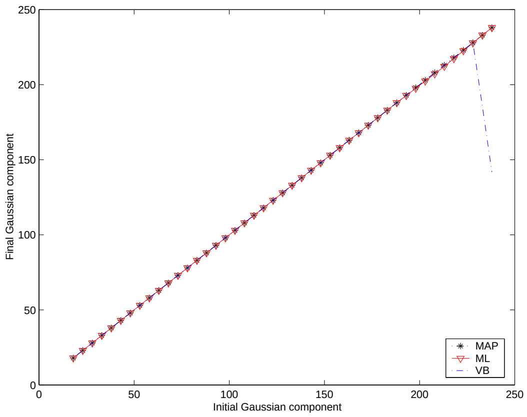


Figure 5.26: Final component (prior = 100 )

## 5.4 Prior optimization

In this section we study the effect of the optimal prior estimation in the VB framework. The procedure for optimal prior estimation is described in section 4.1.3. It consists in iteratively optimizing the bound given some priors and then priors given variational distributions previously obtained. In this way it is possible to further increase the value of the free energy.

In order to study the impact of prior optimization, we run experiments on a 8 Gaussian component GMM with 2000 frames as training data. Broad priors are used (i.e. $\tau = 1E - 6$). Then 30 iterations of the VBEM algorithm described in section 4.1 are done followed from a one-step prior optimization. This procedure is iterated until convergence. Free energy w.r.t. the iteration number is plotted in figure 5.28. During the VBEM iteration, free energy is increasing with iteration. When optimal priors are estimated for the first time, there is a free energy jump, because prior distributions move from non-informative distributions to distributions fitted to data. Then convergence is achieved after few steps. It is important to notice that there is always a jump (positive or negative) in the free energy after a prior optimization: this is because the optimal prior distribution *is not* the optimal posterior distribution.

The problem we want to investigate now is: do optimal prior depend on initial prior choices ? In order to answer this question we run joint variational distribution-prior distribution optimization with different choices for initial priors. Figure 5.29 represents the initial prior choice as a function of the optimization iteration. Figure 5.30 represents free energy value. Unsurprisingly different initial prior choices always converge to the same free energy value, showing that optimal priors are not sensitive to initial prior choice. Anyway this is not valid if the prior distributions are too strong to be modified by data; in this case the optimal prior will simply coincides with the current prior because the data contribution will be negligible.

Figure 5.31 plots in the same way the test log-likelihood obtained after prior optimization. Again the final value of convergence is the same for all initial prior choice but it *is not* the value that maximizes the test set log-likelihood.

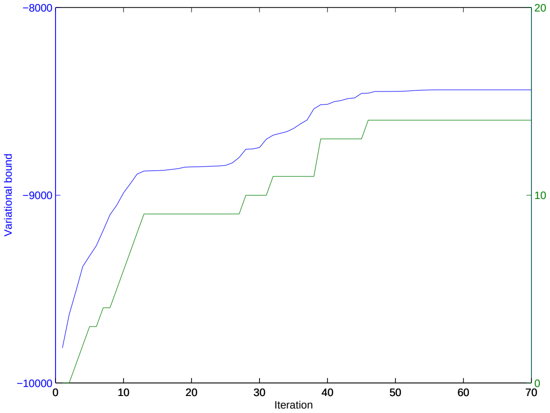


Figure 5.27: Variational Bound and zeros components vs iteration number

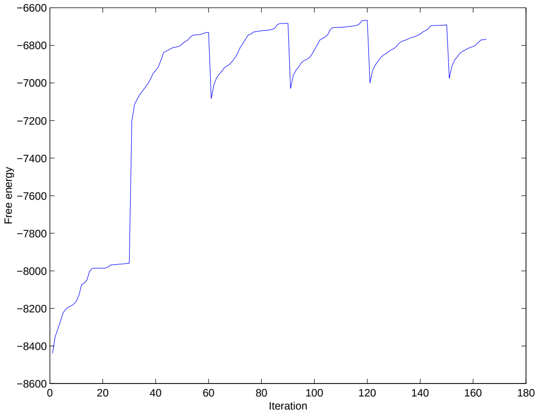


Figure 5.28: Variational Bound with prior optimization w.r.t. iteration (30 iteration for each hyper parameter)

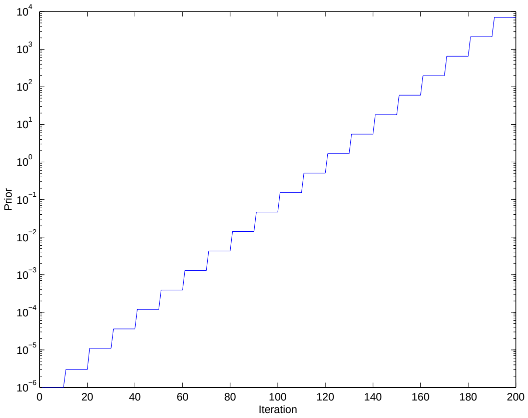


Figure 5.29: Priors w.r.t. iterations

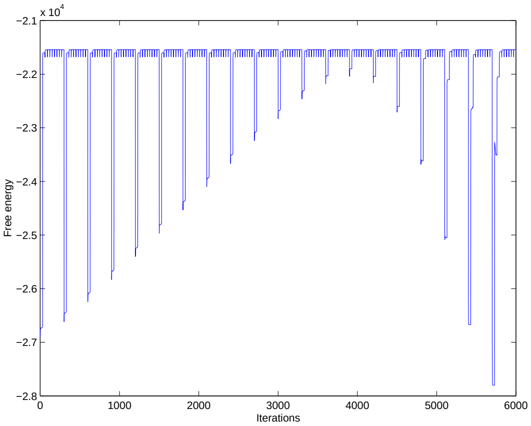


Figure 5.30: Free energy vs. iteration with prior optimization

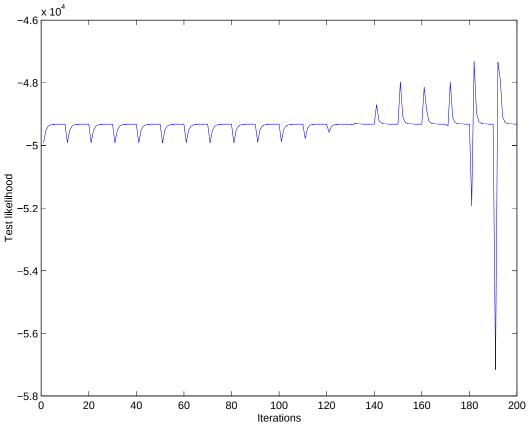


Figure 5.31: Test likelihood with prior optimization

## 5.5 VB model selection

In this section we explore the use of the VB framework for making model selection on a synthetic data problem where the exact component number is known and a real data problem. The theoretical basis of this section can be found in sections 2.6 and 3.4.

At first 1000 observation are generated using a 3 component Gaussian mixture model. This data are used as training set for learning different GMM gradually increasing the component number from 1 to 10. The 10 different models are learned using ML and VB criteria and they are scored using respectively BIC and VB free energy. We investigated as well the dependence of the VB solution from $\tau$. Scores are plotted in figure 5.32. Both BIC and VB select the 3 component models and VB result is robust to prior distribution.

In the second experiment we use 1000 acoustic vectors (MFCC) and repeated the same experiment. The theorical BIC selects a value of 10. Figure 5.33 plots VB free energy for 10 different component models. When the prior $\tau$ is small the chosen value is 4 while when prior strength increases, free energy has same problems as the BIC diverging towards more complex models.

In summary when the data has intrinsically a best model (as in the synthetic case) BIC and VB provide the right solution. On the other hand in real data applications, BIC definitely needs to be tuned to provide reasonable results. VB is sensitive as well to prior distributions in the model selection task.

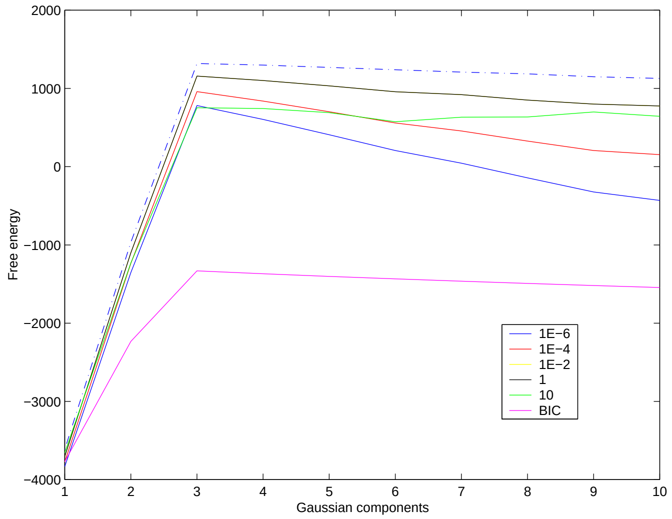


Figure 5.32: Free energy and model selection for synthetic data

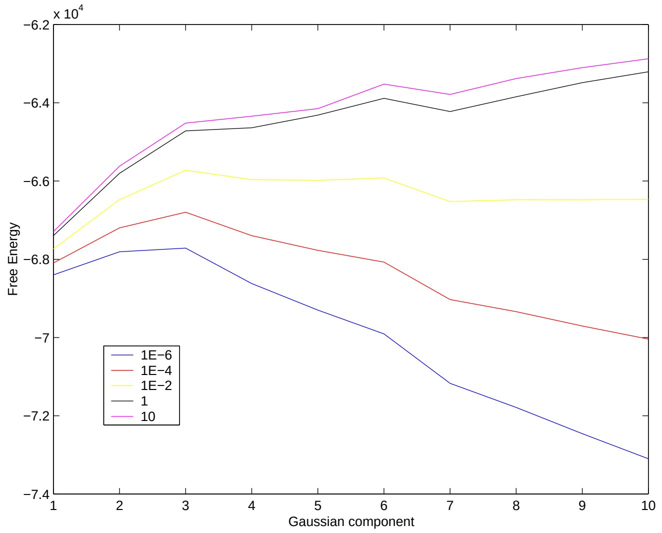


Figure 5.33: Free energy and model selection for real data

## 5.6 Variational Bayesian Adaptation

In this section we compare MAP and VB on the adaptation task using an empirical prior distribution. Empirical Bayes approach considers a prior model given from the knowledge that someone has on the basis of data estimation for instance. In speech and speaker recognition tasks, it generally means the use of a speaker independent model that can be adapted with the current data.

We aims here at comparing the MAP and the VB on the speaker adaptation task with empirical Bayes prior. In order to estimate the prior model, a 32 Gaussian component GMM is estimated using 50k acoustic vectors from different speakers. This background model is adapted using different amount of data from a given speaker and the adapted model is scored on a test set from the same speaker. Prior distribution sharpness is regulated again with a parameter $\tau$ that produces distributions with the same form as in [46] i.e. given a background model $B = \sum_{i=1}^{M} c_i^b N(\mu_i^b, \Sigma_i^b)$ and a strength factor $\tau_i$ (one for each Gaussian), hyperparameters for prior distributions can be estimated as :

$$\lambda_{0i} = c_i^b \sum_i \tau_i \tag{5.7}$$

$$\rho_{0i} = \mu_i^b \tag{5.8}$$

$$\xi_{0i} = \tau_i^b \tag{5.9}$$

$$a_{0i} = \tau_i \tag{5.10}$$

$$B_{0i} = \tau_i \Sigma_i^{b-1} \tag{5.11}$$

We set $\tau_i = \tau \ \ \forall i$. A small $\tau$ results in not sharped prior distribution, on the other side a large $\tau$ results in very sharped prior.

As already noted in section 2.5, application of MAP to GMM may result in divergence problems for the weights estimation (i.e. $c_i = \frac{\lambda_i - 1}{\sum_i^M \lambda_i - 1}$ if $\lambda_i \leq 1$). In order to avoid this limit case, the empirical adaptation formula proposed in [113] is used i.e.:

$$c_i = \alpha_i c_i^b + (1 - \alpha_i)\frac{\gamma_i}{T} \tag{5.12}$$

$$\alpha_i = \frac{\gamma_i}{\tau_i} \tag{5.13}$$

Even if this solution is not mathematically derived from the MAP, it avoids degenerated case.

### 5.6.1 Performance vs. data for MAP adaptation and VB adaptation

In the first set of experiments, test log-likelihood is plotted w.r.t. the amount of adaptation data for different choices of $\tau = \{1E - 5, 1, 1E + 4\}$ in figures 5.34,5.36,5.38. As in previous experiments for the same scenarios accumulator variance is plotted in figures 5.35,5.37,5.39.

The test log-likelihood inferred using the VB is higher than the one inferred using the MAP. The difference is bigger when few training data are used and when small $\tau$ is used but there is no large gain compared with experiments in section 5.1 when non-empirical Bayes prior are used. In fact using an empirical background model that has a lot of information about the model reduces the difference between the two approaches. On the other hand, it is interesting to check the accumulator variance. When large $\tau$ is used MAP and VB accumulator variances coincide; when small $\tau$ is used we can identify two phases: when few data are available MAP clusters data harder than VB; when huge amount of data is used VB clusters data harder than MAP. This can be seen an attempt of VB to build a model avoiding overfitting problems i.e. when few data are used they are splitted all over Gaussians avoiding some Gaussian to fit in a strong way the few data and when more data are used it avoid to fit all Gaussians giving a model that's too generic.

In other words we can see here the averaging property of the VB method that does not aim at finding some parameters from some data but aims at estimating distribution; as direct consequence this estimation is smoother compared with other parameter estimation techniques.

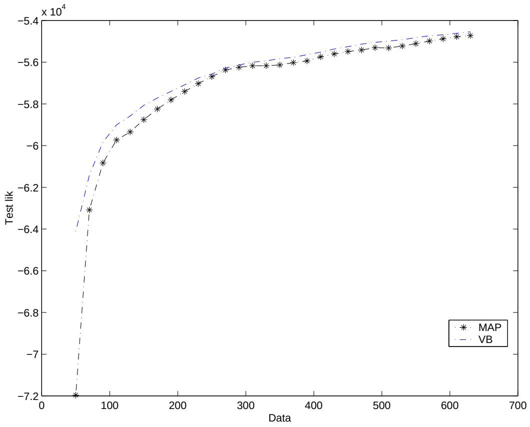


Figure 5.34: Test log likelihood for 32 components GMM with (prior = 1E-5 )

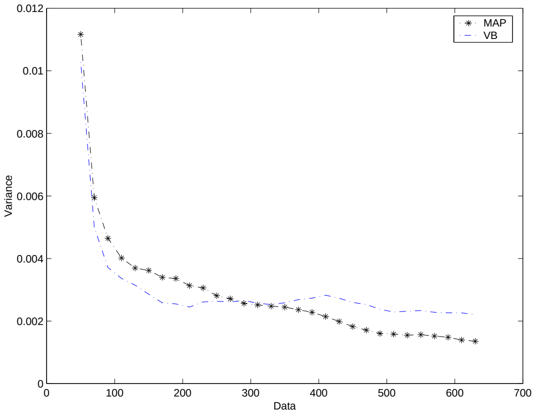


Figure 5.35: Accumulator variance for 32 components GMM with (prior = 1E-5 )

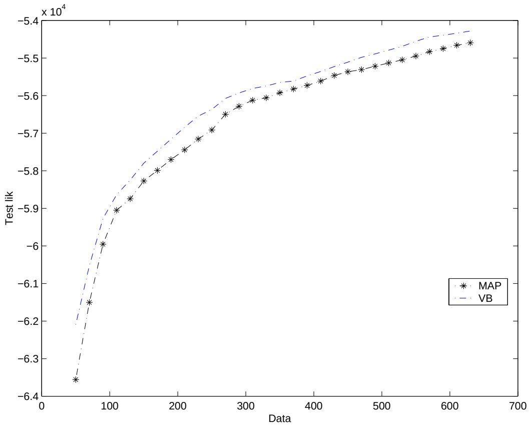


Figure 5.36: Test log likelihood for 32 components GMM with (prior = 1 )

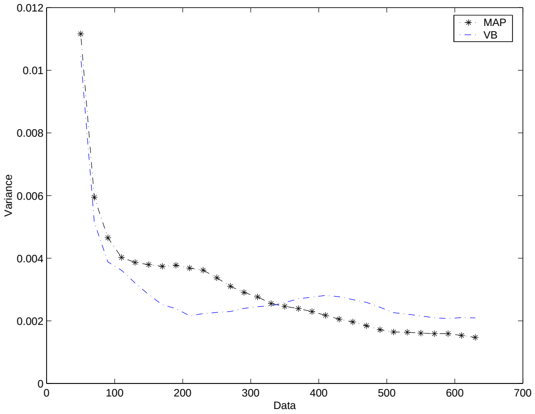


Figure 5.37: Accumulator variance for 32 components GMM with (prior = 1 )

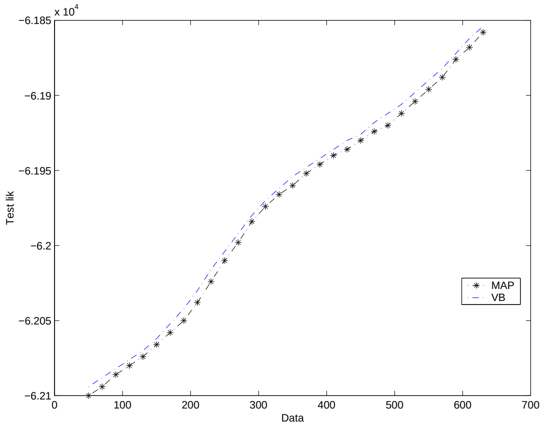


Figure 5.38: Test log likelihood for 32 components GMM with (prior = 1E+4 )

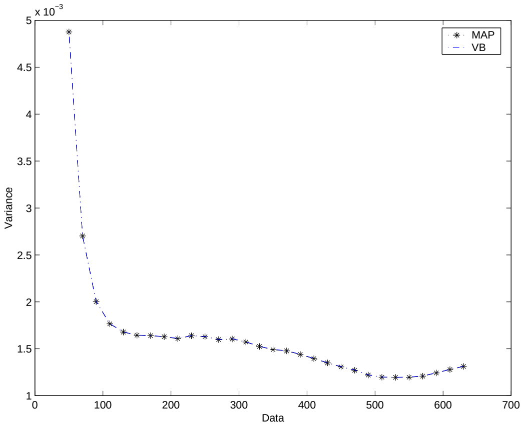


Figure 5.39: Accumulator variance for 32 components GMM with (prior = 1E+4 )

### 5.6.2 Performance vs. prior for MAP adaptation and VB adaptation

Figures 5.40,5.42 and 5.44 plot the test-log likelihood w.r.t. the prior factor $\tau$ for three different amounts of adaptation data (100, 400 and 1000). As expected in the case of small amount of training data and weak prior distributions, the VB approach largely outperforms the MAP. After a peak at the optimal value of $\tau$ performances degrade when priors become too strong.

On the other hand when the amount of data increases, the difference between MAP and VB is marginal at all values of $\tau$.

## 5.7 Occam factor in the the E step

Looking at EM and VBEM equations, we can notice a certain similarity with MAP. Since conjugate priors are used the M step consists for both MAP and VB in the update of hyperparameters with new statistics coming from the data. The E step is significantly different because in the case of VB, parameters are marginalized w.r.t. their distributions. This directly takes into consideration the Occam factor ([93]).

The E-steps for MAP and VB learning are

$$E_{MAP} = N(y_t|\theta_{MAP}) \tag{5.14}$$

$$E_{VB} = exp(\int Q(\theta)\, log\, N(y_t|\theta)d\theta) \tag{5.15}$$

We know by assumption that $\theta_{MAP} = argmax_\theta P(y|\theta)P(\theta)$ where $P(\theta)$ is the parameter prior distribution and $Q(\theta)$ is the variational distribution. Let us define $\alpha$ as:

$$\alpha = \frac{\int Q(\theta) log\, N(y_t|\theta)d\theta}{log\, N(y_t|\theta_{MAP})} \tag{5.16}$$

Let us rewrite $E_{VB}$ as function of $\alpha$:

$$E_{VB} = exp(\alpha\, log\, N(y_t|\theta_{MAP})) = exp(log\, N(y_t|\theta_{MAP})^\alpha) = N(y_t|\theta_{MAP})^\alpha. \tag{5.17}$$

i.e. the variational E step is the same as the MAP E step with an exponential factor $\alpha$. The value of $\alpha$ will actually depend on the shape of the distributions $N(y_t|\theta_{MAP})$,$Q(\theta)$ and of course on $y_t$; If $Q(\theta)$ is extremely peaked around parameter set $\theta_{MAP}$ then $\alpha \to 1$.

In order to show the impact of $\alpha$ into clustering let us consider a simple case of 2 component GMM with a training set of only one point. Let us suppose that providing some kind of initialization the MAP E-step give us $\{a, b\}$. Considering the previous discussion, the VB E-step will provide $\{a^\alpha, b^\beta\}$ where $\alpha, \beta$ are Occam factors for VB expectations. $\{a, b\}$ must actually be normalized into:

$$c_{MAP} = \frac{a}{a+b} = \frac{1}{1+\frac{b}{a}} \qquad c_{VB} = \frac{a^\alpha}{a^\alpha + b^\beta} = \frac{1}{1+\frac{b^\beta}{a^\alpha}} \tag{5.18}$$

We are actually interested for determining the law that regulates the difference between $\pi_{MAP}$ and $\pi_{VB}$ i.e. values of $a, b, \alpha, \beta$ that make the VB estimation different from the MAP estimation. So let us consider the following inequality:

$$\frac{b^\beta}{a^\alpha} \geq \frac{b}{a} \tag{5.19}$$

Considering a given couple of $a, b$, the inequalities has the form of a plane with intersection with axes given by $r_\beta, t_\alpha$:

$$\beta \log b - \alpha \log a \geq \log b - \log a \tag{5.20}$$

$$r_\beta = 1 - \frac{\log a}{\log b} \qquad t_\alpha = 1 - \frac{\log b}{\log a} \tag{5.21}$$

Knowing that $a, b$ are expectations they can only be positive values and that $\alpha, \beta > 0$, the plane may intersect the possible values of $\alpha, \beta$. Conditions for $r_\beta, t_\alpha > 0$ are complementary:

$$r_\beta > 0 \Rightarrow \{b > a\} \tag{5.22}$$

$$t_\alpha > 0 \Rightarrow \{a > b\} \tag{5.23}$$

Iso-surfaces for the given function of $\alpha, \beta$ are planes. Let us now consider relation between accumulator variances that can be rewritten as:

$$\pi_{MAP}^2 - \pi_{MAP} \geq \pi_{VB}^2 - \pi_{VB} \tag{5.24}$$

That has always as solution:

$$a^\alpha \leq \frac{b^\beta \times (c - \sqrt{c^2 - 4})}{2} \vee a^\alpha \geq \frac{b^\beta \times (c + \sqrt{c^2 - 4})}{2} \tag{5.25}$$

where $c = \frac{a}{b} + \frac{b}{a}$ and $c > 2$. Those are again equations of two planes.

Let us plot $\pi_{MAP}, \pi_{VB}$ and $var_{MAP}, var_{VB}$ (variance accumulator for MAP and VB) for a simple couple {a = 1E-5,b = 3E-5} function of $\alpha, \beta$ in figures 5.46,5.47.

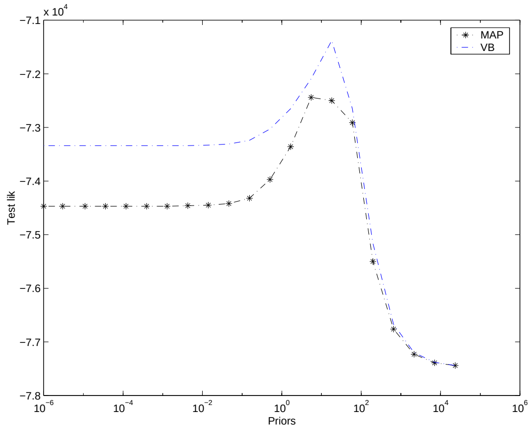
When a set of N data is used the result for the final Gaussian weight is a sum of function with the same shape as 5.46 and the iso surface will be no more a plane. For 5 points, for example the result is plotted in figures 5.48-5.49.

## 5.8 Conclusion

In this chapter we have experimentally verified some theoretical results in a GMM task. ML, MAP and VB have the same performances in the inference task when enough training data are provided. On the other hand when some extreme situations are considered like very poor amount of training data, very large models or particularly strong prior VB outperforms both MAP and ML.

In the adaptation task the averaging property of VB is even more evident: when few data are available they are clustered harder than the MAP criterion while when lots of data are used they are not concentrated on few Gaussians but they are spread all over other components. This can be seen as combination of two different effects: one coming from the Occam factor and the other coming from the averaging effect. The Occam factor tries to use just degrees of freedom necessary to the modeling task and it explains why few Gaussian components only are modified when few data are used contrarily to MAP. On the other hand when more data are used the averaging effect of VB results in a more soft clustering.

In some sense Occam factor and averaging effect are both consequence of the Bayesian integral and the final result can be very different depending on the prior distribution as described in section 5.7.



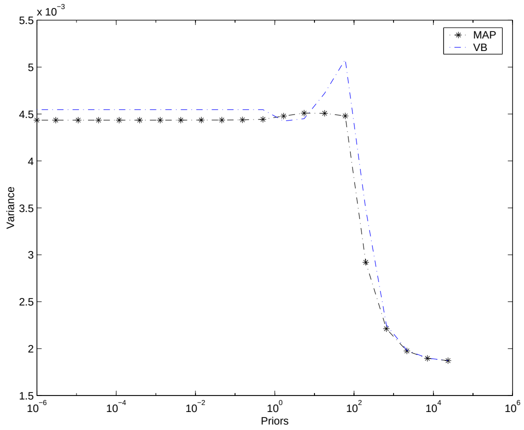
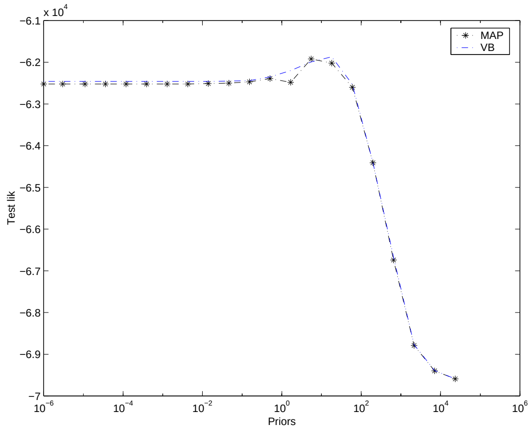


Figure 5.40: Test log likelihood for 32 components GMM with (data = 100)

Figure 5.41: Accumulator for 32 components GMM with (data = 100)



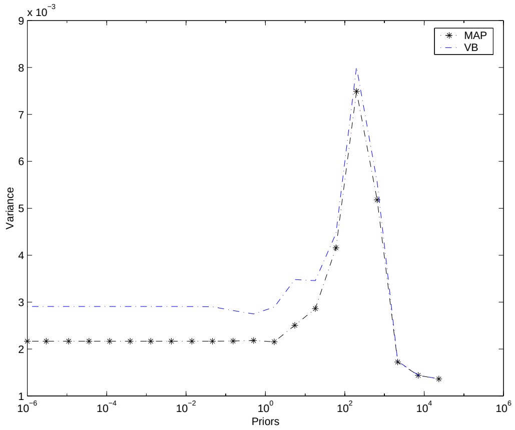
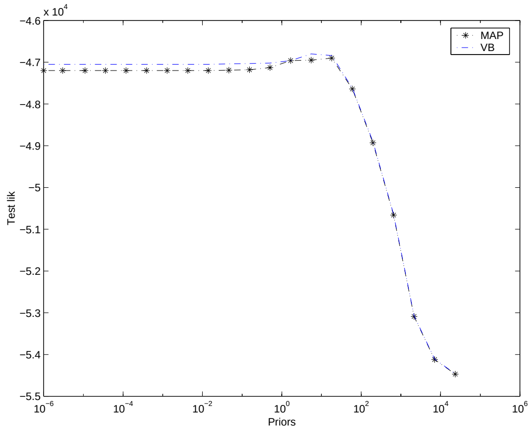


Figure 5.42: Test log likelihood for 32 components GMM with (data = 400)

Figure 5.43: Accumulator likelihood for 32 components GMM with (data = 400)



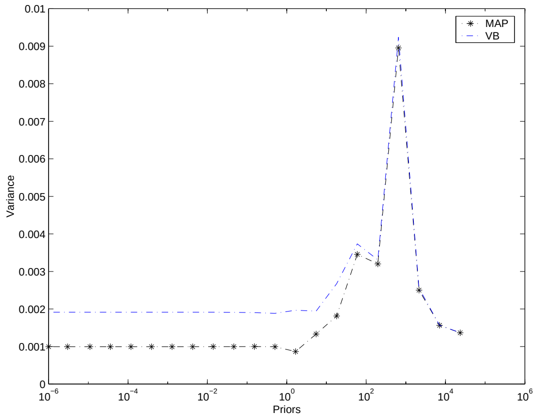


Figure 5.44: Test log likelihood for 32 components GMM with (data = 1000)

Figure 5.45: Accumulator likelihood for 32 components GMM with (data = 1000)

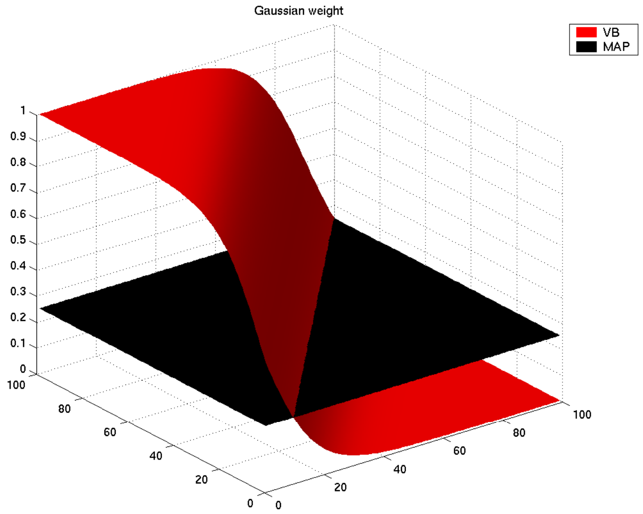


Figure 5.46: Weight estimation for VB and MAP

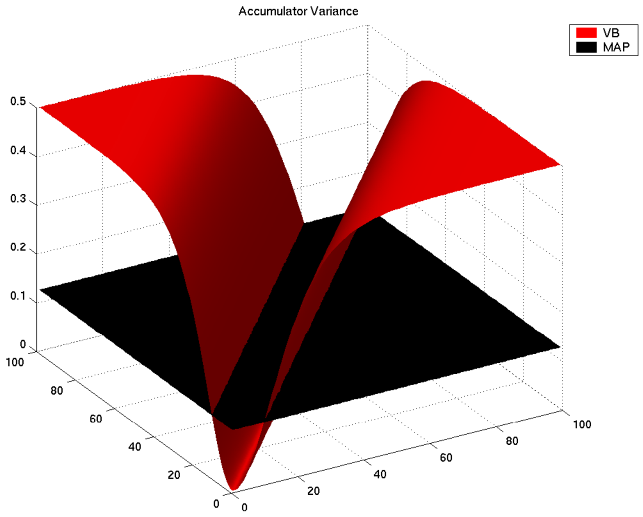


Figure 5.47: Accumulator Variance for VB and MAP

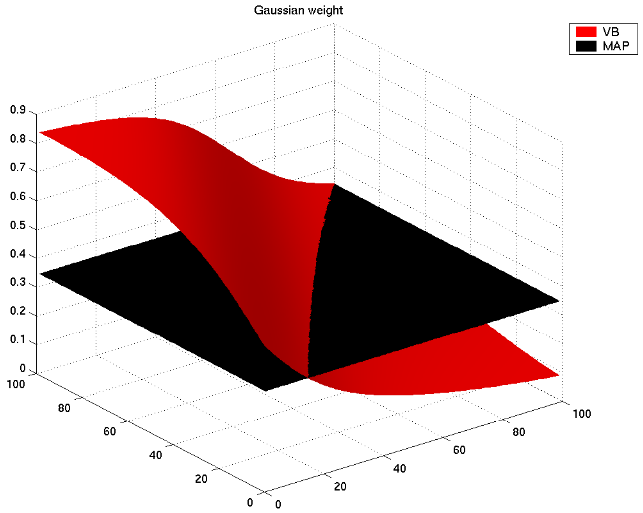


Figure 5.48: Weight estimation for VB and MAP

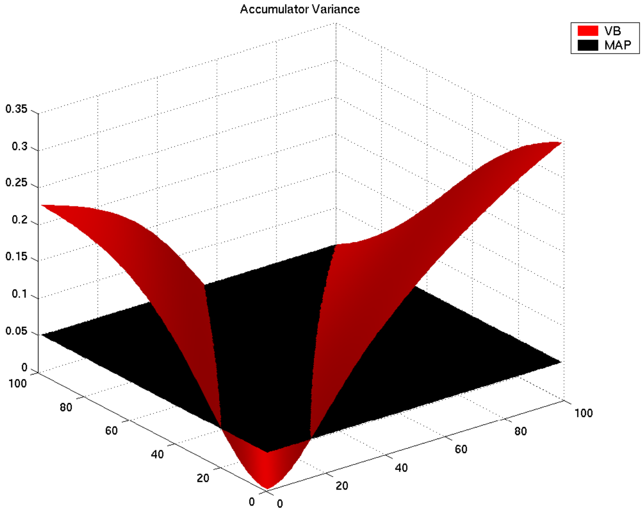


Figure 5.49: Accumulator Variance for VB and MAP

# Chapter 6

# Variational Bayesian Speaker Change Detection

An important problem to solve in audio document indexing is to detect the speaker change[1]. In this chapter we explore how VB learning may contribute to this task. The classical solution to this problem consists in supposing two competing models, one for the speaker change (i.e. for two speakers) and another for one speaker. The best model is selected using a selection criterion or a thresholded measure (e.g. the Gish distance).

We detail in the following chapter some theorical problems related to the derivation of the BIC formulation, LLR formulation and we propose the use of the VB metric for speaker change detection.

## 6.1 State of the art

In many audio systems, data segmentation is a task of main importance. For instance in large vocabulary speech recognition systems for broadcast news, obtaining an initial segmentation in homogeneous blocks of audio data is a necessary step for adaptation and decoding purposes (see [45],[32],[20]).

In the literature there are three main groups of speaker change algorithms: the decoder based segmentation, the model based segmentation and the metric-based segmentation.

The decoder-guided segmentation is simply the result of the decoding of the audio file. Segmentation is then obtained cutting at the decoded silence points generated from the recognizer (see [81],[143]). Anyway this kind of solution considers the silence points, and does not explicitly take care of changes in the acoustic data.

In the model-based segmentation different models represent different acoustic

[1] Speaker Change Detection is also referred in literature as Speaker Turn

data, for instance speech/non-speech and eventually sub-models like narrow band speech, wide band speech, noisy speech etc. (see [12],[78]). The segmentation is obtained with a Viterbi decoder with all models in parallel. Obviously if there is a mismatch between the model and the data the technique becomes ineffective on the unseen data.

Metric-based segmentation system are the most popular solution. They are based on two neighboring windows and a similarity measure between the two windows in order to determine if a changing point takes place. The most common choices for metrics are the KL divergence ([123]), the LLR ([28], [38], [37],[102]) and the BIC ([31],[127],[38],[90],[145],[135]). All metric based methods need an heuristic threshold to be fixed in order to obtain effective results.

## 6.2 Mathematical formulation

Let us consider a window $Y = \{y_1, \dots, y_N\}$ in which we suppose a changing point at time $t$. The hypothesis of changing ($H_{m12}$) point considers that $Y_1 = \{y_1, \dots, y_t\}$ and $Y_2 = \{y_t, \dots, y_N\}$ are generated by two different speakers while the hypothesis of no speaker change ($H_{m0}$) consider that $Y = \{Y_1, Y_2\}$ is generated by a single speaker. In terms of model, $H_{m0}$ is parameterized with a single Gaussian with parameters $\theta_{m0}$ while $H_{m12}$ is parameterized with two Gaussian distributions with parameters $\theta_{m1}, \theta_{m2}$ estimated respectively on $Y_1, Y_2$.

The metric based method aims at comparing the two hypotheses based on some dissimilarity criterion in order to infer one of the two hypothesis. Most common criteria for doing this kind of inference are the Log Likelihood Ratio (LLR) and the Bayesian Information Criterion (BIC).

### 6.2.1 Log Likelihood Ratio

In the Log Likelihood Ratio (LLR) technique the two hypothesis are represented as the ratio of the likelihood of the hypothesis $H_{m0}$ (no speaker change) and hypothesis $H_{m12}$ (speaker change). Mathematically speaking we have:

$$L_{m0} = \sum_{Y=\{Y_1,Y_2\}} log\, p(y|\theta_{m0}) \tag{6.1}$$

$$L_{m12} = \sum_{Y_1} log\, p(y|\theta_{m1}) + \sum_{Y_2} log\, p(y|\theta_{m2}) \tag{6.2}$$

where $\theta_{m0}, \theta_{m1}, \theta_{m2}$ are parameters defined in previous section.

It must be anyway pointed out that 6.2 is not a real log likelihood because it is not possible to write a probability density function for this form.

The decision is taken considering the LLR as:

$$LLR = L_{m0} - L_{m12} \geq \lambda \tag{6.3}$$

where $\lambda$ is a threshold heuristically set. As long as we have $L_{m12} > L_{m0}$, the LLR will be always negative; for this reason the empirical threshold must be set in order to make a decision on the changing point.

### 6.2.2 Bayesian Information Criterion

As already mentioned in a previous chapter, the BIC penalizes more complex models and was introduced in the speaker change detection framework in [31].

In this case the threshold is simply given by the complexity of the two models i.e.:

$$BIC = L_{m12} - L_{m0} - \frac{\lambda}{2} P \, log \, N \tag{6.4}$$

where $P$ is the difference of number of parameters in the two models.

The tuning parameter $\lambda$ was introduced in [127] in order to improve performances versus the theorical BIC that provides generally very poor results. Introducing the "tuning" parameter BIC is basically a thresholded LLR.

The BIC is based on an approximation of the probability density function but in the hypothesis $H_{m12}$ there is no valid pdf. This is an improper application of the BIC.

Tuning of the parameter $\lambda$ can be done on a development data set like in [38] but as soon as conditions slightly change the criterion becomes less effective.

## 6.3 Variational Bayesian Speaker change detection

The VB model selection framework has a natural application in the speaker change detection problem. Instead of comparing log-likelihood or penalized log-likelihood it is in fact enough to compare the variational free energies of two models. The problem that the two Gaussian model is not a valid pdf is overcome with a very simple trick: the two Gaussian model is considered as a 2 components GMM in which during the learning hidden variables are forced to belong to one of the two Gaussians. In this way it is possible to obtain a valid pdf to approximate with variational methods.

Let us consider Gaussians with parameters $\theta_{m0} = \{\mu_{m0}, \Gamma_{m0}\}, \theta_{m1} = \{\mu_{m1}, \Gamma_{m1}\}, \theta_{m2} = \{\mu_{m2}, \Gamma_{m2}\}$. In the Variational Bayesian formulation prior probability functions over parameters must be defined first. Then choosing probability functions in the conjugate-exponential family we define:

$$p(\mu_{m0}|\Gamma_{m0}) = N(\mu_{m0}|\rho_0, \beta_0 \Gamma_{m0}) \quad p(\Gamma_{m0}) = W(a_0, \Phi_0) \tag{6.5}$$

$$p(\mu_{m1}|\Gamma_{m1}) = N(\mu_{m1}|\rho_0, \beta_0 \Gamma_{m1}) \quad p(\Gamma_{m1}) = W(a_0, \Phi_0) \tag{6.6}$$

$$p(\mu_{m2}|\Gamma_{m2}) = N(\mu_{m2}|\rho_0, \beta_0 \Gamma_{m2}) \quad p(\Gamma_{m2}) = W(a_0, \Phi_0) \tag{6.7}$$

where $W()$ designate a Wishart distribution and $\{\rho_0, \beta_0, a_0, \Phi_0\}$ are distribution hyperparameters. Estimation of variational posterior distributions is easygoing because they have the same form as prior but with updated hyperparameters (see section 3.3). Let us define the following quantities:

$$\bar{\mu}_{m0} = \frac{1}{N}\sum_{i=1}^{N} y_i \quad \bar{\mu}_{m1} = \frac{1}{t}\sum_{i=1}^{t} y_i \quad \bar{\mu}_{m2} = \frac{1}{N-t}\sum_{i=t}^{N} y_i \tag{6.8}$$

$$\bar{\Sigma}_{m0} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{\mu}_{m0})^T(y_i - \bar{\mu}_{m0}) \tag{6.9}$$

$$\bar{\Sigma}_{m1} = \frac{1}{t}\sum_{i=1}^{t}(y_i - \bar{\mu}_{m1})^T(y_i - \bar{\mu}_{m1}) \tag{6.10}$$

$$\bar{\Sigma}_{m2} = \frac{1}{N-t}\sum_{i=t}^{N}(y_i - \bar{\mu}_{m2})^T(y_i - \bar{\mu}_{m2}) \tag{6.11}$$

Variational distributions have the following form:

$$q(\mu_{m0}|\Gamma_{m0}) = N(\rho_{m0}|\beta_{m0}\Gamma_{m0}) \quad q(\Gamma_{m0}) = W(a_{m0}, \Phi_{m0}) \tag{6.12}$$

$$q(\mu_{m1}|\Gamma_{m1}) = N(\rho_{m1}|\beta_{m1}\Gamma_{m1}) \quad q(\Gamma_{m1}) = W(a_{m1}, \Phi_{m1}) \tag{6.13}$$

$$q(\mu_{m2}|\Gamma_{m2}) = N(\rho_{m2}|\beta_{m2}\Gamma_{m2}) \quad q(\Gamma_{m2}) = W(a_{m2}, \Phi_{m2}) \tag{6.14}$$

where

$$\beta_{m0} = N + \beta_0 \tag{6.15}$$

$$\beta_{m1} = t + \beta_0 \tag{6.16}$$

$$\beta_{m2} = N - t + \beta_0 \tag{6.17}$$

$$a_{m0} = N + a_0 \tag{6.18}$$

$$a_{m1} = t + a_0 \tag{6.19}$$

$$a_{m2} = N - t + a_0 \tag{6.20}$$

$$\rho_{m0} = \frac{N\bar{\mu}_{m0} + \beta_0\rho_0}{N + \beta_0} \tag{6.21}$$

$$\rho_{m1} = \frac{N\bar{\mu}_{m1} + \beta_0\rho_0}{N + \beta_0} \tag{6.22}$$

$$\rho_{m2} = \frac{N\bar{\mu}_{m2} + \beta_0\rho_0}{N + \beta_0} \tag{6.23}$$

$$\Phi_{m0} = \Phi_0 + T\,\bar{\Sigma}_{m0} + T\,\beta_0(\bar{\mu}_{m0} - \rho_0)^T(\bar{\mu}_{m0} - \rho_0)/(T + \beta_0) \tag{6.24}$$

$$\Phi_{m1} = \Phi_0 + T\,\bar{\Sigma}_{m1} + T\,\beta_0(\bar{\mu}_{m1} - \rho_0)^T(\bar{\mu}_{m1} - \rho_0)/(T + \beta_0) \tag{6.25}$$

$$\Phi_{m2} = \Phi_0 + T\,\bar{\Sigma}_{m2} + T\,\beta_0(\bar{\mu}_{m2} - \rho_0)^T(\bar{\mu}_{m2} - \rho_0)/(T + \beta_0) \tag{6.26}$$

The decision over the speaker change is taken now simply comparing the variational posterior over models for hypothesis $H_{m0}$ and $H_{m12}$. Let us designates with $F_{m0}$ and $F_{m12}$ free energies for models $H_{m0}$ and $H_{m12}$ (even if $H_{m12}$ is not a valid model). Variational posterior over models, designated as $q(m0)$ and $q(m12)$ are:

$$q(m0) = exp((F_{m0})p(m0) \tag{6.27}$$

$$q(m12) = exp((F_{m12})p(m12) \tag{6.28}$$

The decision is taken comparing $q(m0)$ and $q(m1)$. Because there is no available prior information on the prior model probabilities, it is reasonable to set $p(m0) = p(m1)$. Furthermore the exponential function is a monotonic function that means that the decision can be simply taken on the free energies difference:

$$VB_m = F_{m0} - F_{m12} \geq 0 \tag{6.29}$$

Both $F_{m0}$ and $F_{m12}$ embed a penalty term given by the KL divergence between prior and posterior distributions. The decision does not require any threshold.

Anyway in Bayesian approaches prior distributions over parameters must be defined. We will see later in this chapter that the VB speaker change detection is extremely sensitive to prior distributions.

### 6.3.1 Free energy for speaker changes

The term $VB_m$ can be easily written using results of section 4.1. The expression of $F_{m0}$ is a form mathematically correct:

$$\begin{aligned} F_{m0} = \sum_{i=1}^{N} \int q(\mu_{m0}|\Gamma_{m0})q(\Gamma_{m0})\, log\, p(y_i|\mu_{m0}, \Gamma_{m0}) + \\ -KL(q(\mu_{m0}|\Gamma_{m0})||p(\mu_{m0}|\Gamma_{m0}))) - KL(q(\Gamma_{m0})||q(\Gamma_{m0})) \end{aligned} \tag{6.30}$$

The first integral can be evaluated as in section 4.1.2 and KL divergences can be found in appendix A.

On the other side $F_{m12}$ is not a valid free energy because there is no valid probability density function to bound. As in the BIC this is approximated by the two free energy of single Gaussians:

$$\begin{aligned} F_{m12} \quad &= \textstyle\sum_{i=1}^{t} \int q(\mu_{m1}|\Gamma_{m1})q(\Gamma_{m1})\, log\, p(y_i|\mu_{m1}, \Gamma_{m1}) + \\ &+ \textstyle\sum_{i=t+1}^{N} \int q(\mu_{m2}|\Gamma_{m2})q(\Gamma_{m2})\, log\, p(y_i|\mu_{m2}, \Gamma_{m2}) \\ &-KL(q(\mu_{m1}|\Gamma_{m1})||p(\mu_{m1}|\Gamma_{m1}))) - KL(q(\Gamma_{m1})||q(\Gamma_{m1})) \\ &-KL(q(\mu_{m2}|\Gamma_{m2})||p(\mu_{m2}|\Gamma_{m2}))) - KL(q(\Gamma_{m2})||q(\Gamma_{m2})) \end{aligned} \tag{6.31}$$

Each term in 6.31 is computable as before.

## 6.4 Search algorithm

In order to compare the BIC and the VB solutions in the fairest way, a common experimental framework is fixed. The search algorithm used in our experiments is the same as [31]. This is a classical algorithm for speaker change point detection and it is based on a window of growing size in which the changing point is searched. If no changing point is detected the window grows of a size equal to $MOREFRAMES$; once the maximum size i.e. $MAXWINDOW$ is achieved the whole window is considered as produced by the same speaker and the search continues on the remaining data. The algorithm can be summarized as follows:

```
1 Initialize the window $[a, b]$ where $a = 0$ and $b = MINWINDOW$
2 Find the changing point in $[a, b]$
  for BIC find the point of local maxima of $BIC(m) \geq 0$
  for VB find the point of local maxima of $VB(m) \geq 0$
3 if no change is detected in $[a, b]$ then $b = b + MOREFRAMES$
  else if $t$ is the detected changing point in $[a, b]$ then $a = t + 1$, $b = a + MOREFRAMES$
4 if $b - a > MAXWINDOW$ then $a = b - MAXWINDOW$
5 go to point 2
```

Value of $MOREFRAMES$ is experimentally set to 1 second and value of $MAXWINDOW$ is set to 10 seconds.

## 6.5 Experiments

### 6.5.1 Database description

Speaker change algorithms are tested on the four files of the HUB4-96BN database for a total of almost 2 hours of broadcast news in very different conditions. From labels we can obtain 528 different speaker change points that represent a consistent number of data for experimental tests. Anyway those files contains large non-speech parts; we will not consider changing points inside the non-speech parts because there is no label to determine if the change is correct or not. On the other side we will consider changing point from speech to non-speech segments.

### 6.5.2 Metric

Two kinds of errors are generally considered in literature (see [31]). The type I error also referred as recall (RCL) when a true change is not found in a given window and the type 2 error also referred as precision (PRC) when a detected change is not referred in the label file. To summarize:

$$PRC = \frac{\text{number of correct detected speaker changes}}{\text{total number of detected speaker changes}} \tag{6.32}$$

$$RCL = \frac{\text{number of correct detected speaker changes}}{\text{total number of real speaker changes}} \tag{6.33}$$

An average measure between PRC and RCL is the so called F-measure (see [114]) defined as:

$$F = \frac{2 \times PRC \times RCL}{PRC + RCL} \tag{6.34}$$

Measure 6.34 can be more intuitively rewritten as

$$\frac{1}{F} = 2(\frac{1}{PRC} + \frac{1}{RCL}) \tag{6.35}$$

### 6.5.3 Setting prior distributions

Variational Bayesian algorithms must be provided with prior distributions over parameters and we expect a final result (in terms of model training and model selection) affected by the choice of those prior distributions.

In this specific application, prior distributions are set as Normal-Wishart distributions with hyperparameters $\{\beta_0, a_0, \mu_0, \rho_0, \Phi_0\}$. The same choice as in [46] is assumed for tying together different hyperparameters in order to obtain a more robust estimation. Mathematically speaking we set:

$$a_0 = \beta_0 = \tau \tag{6.36}$$

$$\Phi_0 = \tau \, I \tag{6.37}$$

$$\rho_0 = \bar{y} \tag{6.38}$$

where $I$ is the identity matrix and $\bar{y}$ is the mean computed over the whole file. Performances can be studied now w.r.t. the tuning parameter $\tau$.

A physical interpretation of $\tau$ can be proposed as well. In fact the higher the value of $\tau$ is, the more peaked prior distributions are. In this sense $\tau$ defines the “strength” of the prior distribution.

### 6.5.4 Results

In this section we describe experimental results for the BIC and for the VB framework previously described. The search algorithm and window parameters are the same for the two approaches; in this way the model selection properties of BIC and VB are compared in the fairest way.

Figures 6.1,6.3,6.5 plots the F-measure, the PRC and the RCL for the BIC w.r.t. the threshold $\lambda$ while figures 6.2,6.4,6.6 plots the F-measure, the PRC and the RCL for the VB method w.r.t. the prior strength $\tau$. In table 6.1 results for the best BIC and for the best VB are reported.

In the BIC based system a low value of $\lambda$ produces a large number of speaker changes resulting in RCL close to 1 and in very low PRC. The best $\lambda$ value is 14.5 and results in a $F = 0.63$.

In the VB based system a low value of $\tau$ produces on the contrary a low number of speaker changes resulting in PRC close to 1 and a huge $\tau$ results in high PRC. First of all this application is extremely sensitive to the value of the prior distribution contrarily to the robustness of the VB speaker clustering system defined in chapter 7. This is obviously due to the limited size of the window that makes the model selection very difficult with very few data.

Anyway the use of a Bayesian factor to perform the model selection allows an extremely fine tuning compared to the BIC tuning. This can be seen in results for the best systems: best BIC can achieve $F = 0.63$ while best VB can achieve $F = 0.70$.

| | PRC | RCL | F |
|---|---|---|---|
| BIC $\lambda = 1$ | 0.04 | 1 | 0.07 |
| BIC $\lambda = 14.5$ (best) | 0.88 | 0.49 | 0.63 |
| VB $\tau = 1E - 10$ (best) | 0.75 | 0.66 | 0.70 |

Table 6.1: Value of PRC, RCL and F for the theorical BIC, the tuned BIC and the tuned VB

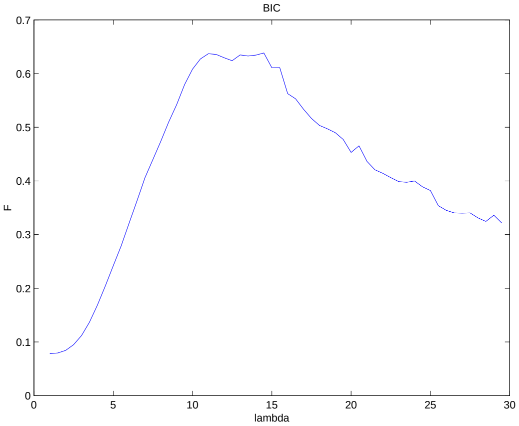


Figure 6.1: F score for BIC method

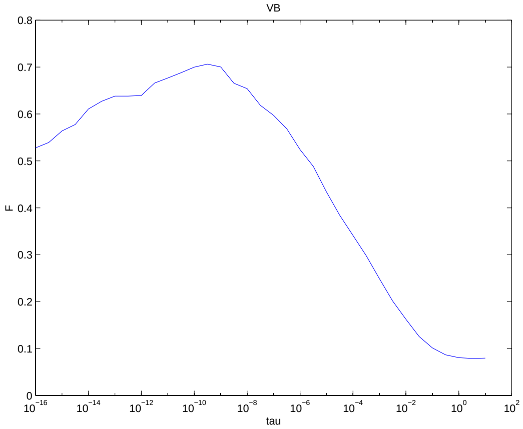


Figure 6.2: F score for VB method

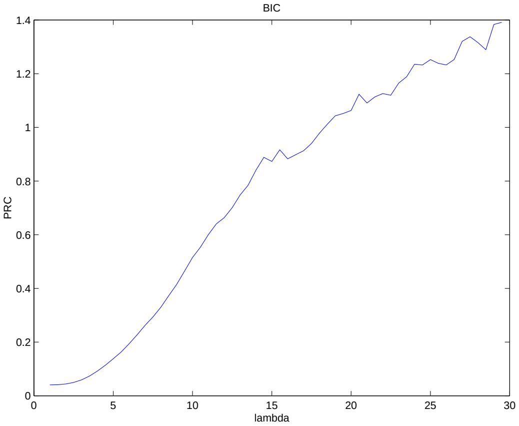


Figure 6.3: PRC score for BIC method

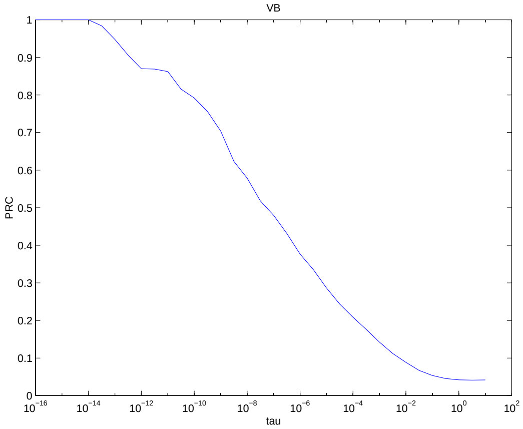


Figure 6.4: PRC score for VB method

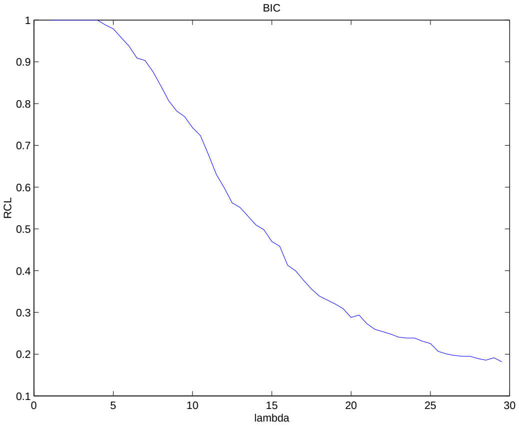


Figure 6.5: RCL score for BIC method

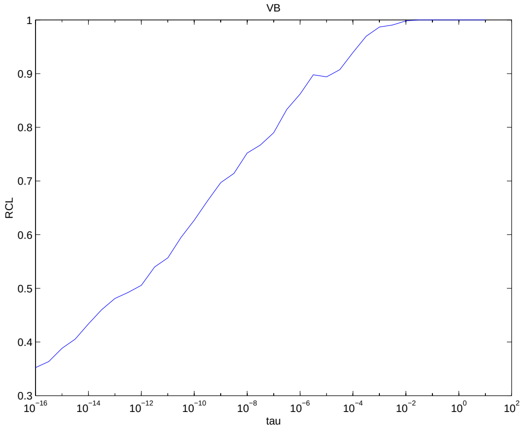


Figure 6.6: RCL score for VB method

# Chapter 7

# Variation Bayesian Speaker Clustering

In this chapter we describe a speaker clustering system based on Variational Bayesian learning. The need of Variational methods arises because the cluster (speaker) number is often unknown and must be estimated from data using different model selection methods. The state of the art systems often use the BIC criterion in order to obtain an estimation of the cluster number. As discussed in chapter 2, BIC is a first order approximation of the Bayesian integral that is only valid in large data limit. On the other hand speaker clustering real problems have often to deal with limited amount of data provided by speakers making the BIC criterion extremely inefficient. The most common trick to balance the poor quantity of data is to multiply the penalty term by a factor $\lambda$ that should bring the criterion closer to the experimental conditions. It is well known that setting the best $\lambda$ value is extremely difficult and it is generally done in an heuristic way. Furthermore the BIC model selection is composed of two completely separate steps: a first step in which the model is trained with a Maximum Likelihood or Maximum a Posteriori criterion and a step in which the penalty term completely independent of the trained model is appended in order to create the score.

The use of Variational Bayesian learning allows simultaneous model learning and model selection: in fact the same objective function (the free energy) is used to determine parameter distributions and to make model selection out of the possible models. The advantage of variational methods is that they are not based on asymptotic properties of the function but on a lower bound derived using the Jensen inequality always valuable even when the amount of data is extremely poor. In this sense, no manual adjustment is required i.e. no heuristic factor $\lambda$ is required to adapt the model selection criterion to the real conditions because it is always valid. Anyway when only poor amount of data is available the problem of the tightness of the bound becomes a main point in fact the bound (that is always verified) may become so vague that model selection may become inefficient.

Furthermore the VB model selection is a biased model selection towards simpler models as described in section 3.4.3.

The other appealing property of Variational Learning consists in its capability of avoiding the model overfitting as experimentally verified in GMM experiments of chapter 5 and of pruning extra degrees of freedom when they are not used. Those properties are useful for indexing purposes when different amount of data are produced by different speakers as it is often the case in speaker segmentation problems; in those situations the overfitting is avoided by the reduction degrees of freedom when only few data are available and the model is too complex.

In this chapter we will describe our speaker segmentation system based on the Variational Bayesian learning/model selection and we will compare it with a system based on ML or MAP for the learning part and BIC for the selection part that is actually the state-of-the-art system for audio indexing.

## 7.1 State of the art

Many different systems for speaker clustering have been proposed based on different techniques but the basic architectures can be classified in three classes according to [101]:

- The ascendant architecture
- The evolutive architecture
- The sequential architecture or the real time architecture

Let us consider in details the three architectures.

### 7.1.1 Ascendant architecture

The ascendant architecture is the very classical architecture for speaker indexing problems proposed by Gish in [55]. This work is the foundation for many of the current indexing systems.

The first step consists in a first phase of silence detection that gives a first rough segmentation into different blocks. This segmentation is simply based on energy thresholding.

Then a speaker change detection is operated as described in section 6.1 based on the BIC (or on the LLR) criterion. This step gives a more precise segmentation into different speakers. After that different segments are merged together using again a similarity measure like BIC or LLR.

This architecture produces extremely pure clusters from a speaker point of view but from the other side the segmentation obtained in the speaker change detection step is not reconsidered in the other steps. This can generate errors if the segmentation is not exact.

### 7.1.2 Evolutive architecture

The evolutive architecture has been proposed in [99] and [100]; this system is called *evolutive* because there is a close interaction between the different steps of the indexing system and the segmentation is refined at each step.

The file is modeled with an HMM where states represents different speakers and transitions model speaker changes (as in proposed in [107]). Anyway in this architecture the HMM is not static but evolves during the different algorithm steps.

At first the whole file is represented with a single state GMM that represent a single speaker. Then a state is added to the model representing another speaker. Models are then adapted using the MAP method and a Viterbi alignment is run on the new model. In this way another segmentation is obtained. Adaptation and alignment are iterated until a maximum of log-likelihood function is achieved. Other speakers (or HMM states) are progressively added and the procedure repeated until an increase in the log-likelihood function is no longer observed.

The interesting point in this kind of approach is that the segmentation is reconsidered after each step and an initial segmentation error in the initial phase may be reconsidered in the next steps. Another advantage consists in the fact that some a priori information can be added using a Universal Background Model for adaptation purposes: this solution has the advantage of requiring less data for obtaining a robust model.

### 7.1.3 Real time architecture

The real time architecture or sequential architecture is an extremely recent solution proposed in [89]. Its advantages consist in the fact that it requires a reduced execution time and the whole signal is not necessary for obtaining a partial segmentation that can be obtained in real time at signal reception.

Contrarily to the hierarchical approach that have a quadratic complexity with the size of the signal, the algorithm proposed in [89] has only a linear complexity.

Real time algorithms are based on three different methods: a "Leader Follower Clustering" (LFC) algorithm as proposed in [41], a dispersion based speaker clustering (DSC) algorithm and an hybrid speaker clustering (HSC) algorithm.

The LFC algorithm uses a clustering technique based on k-means clustering. Given a new observation, only the nearest centroid is modified; if all centroids are too far a new class (speaker) is created. The DSC algorithm is based on an intra-class dispersion criterion: the goal is minimizing the measure that consists in the dispersion. This method has the advantage of being threshold independent. The hybrid algorithm is a combination of the DSC and the LFC because they do not produce the same error. Performances can be improved by using the two techniques at the same time.

Real time architecture performs worst than the classical hierarchical clustering because the use of the whole audio file allows a better selection of the number of classes. On the other hand online speaker clustering is interesting in many applications where the audio stream is processed in real time.

## 7.2 Speaker indexing system

### 7.2.1 Model topology

The model topology chosen for our system is the very classical fully connected HMM in which each state represent a cluster (i.e. a speaker) introduced in [107]. This model is defined by the ensemble of states $\{s_1, \ldots, s_S\}$ where $S$ is the number of states, the transition probabilities from state $r$ to state $j$, $\alpha_{rj}$ and the emission probabilities of an observation $y_t$ given a state $j$ i.e. $p(y_t|s_j)$.

In our model we make an approximation on transition probabilities in order to simplify the problem considering $\{\alpha_{rj}\}$ independent of the original state $r$ i.e.

$$\alpha_{rj} = \alpha_{r'j} = \alpha_j \ \forall \ r, r' \ with \ j = \{1, \ldots, S\} \tag{7.1}$$

Under this approximation the likelihood of a sequence $Y = \{y_t\}$ can be written as a simple mixture model i.e.

$$p(Y) = \sum_S p(Y, S) \rightarrow_{under\,the\,hypothesis} \sum_{j=1}^{S} \alpha_j p(Y|s_j) \tag{7.2}$$

Making this simplification results in a model like the model depicted in figure 7.1.

The main drawback of such simplification is that the time information contained in the file is not used in the modeling task. Anyway when speaker clustering (and generally speaker recognition or verification) is done the temporal information is not essential as long as systems are based on Gaussian Mixture Models (see [112],[113]). Audio indexing systems are also generally based on algorithms that do not consider the time properties of the signal (see e.g. [99],[3],[83]). Anyway more complicated speaker indexing systems (as well as speaker verification or identification systems) have been proposed that uses high-level informations like phonetic or lexical properties that are aware of the time structure of the signal (for instance see [4],[11],[76]). In conclusion using a system that does not explicitly take care of time is still a reasonable approximation.

On the other hand a main advantage consists in the extremely simplified learning and decoding algorithms that can be obtained considering a mixture model instead of an HMM. The gain in terms of computational time and complexity is considerable. For this reason we will base our experimental system on this assumption.

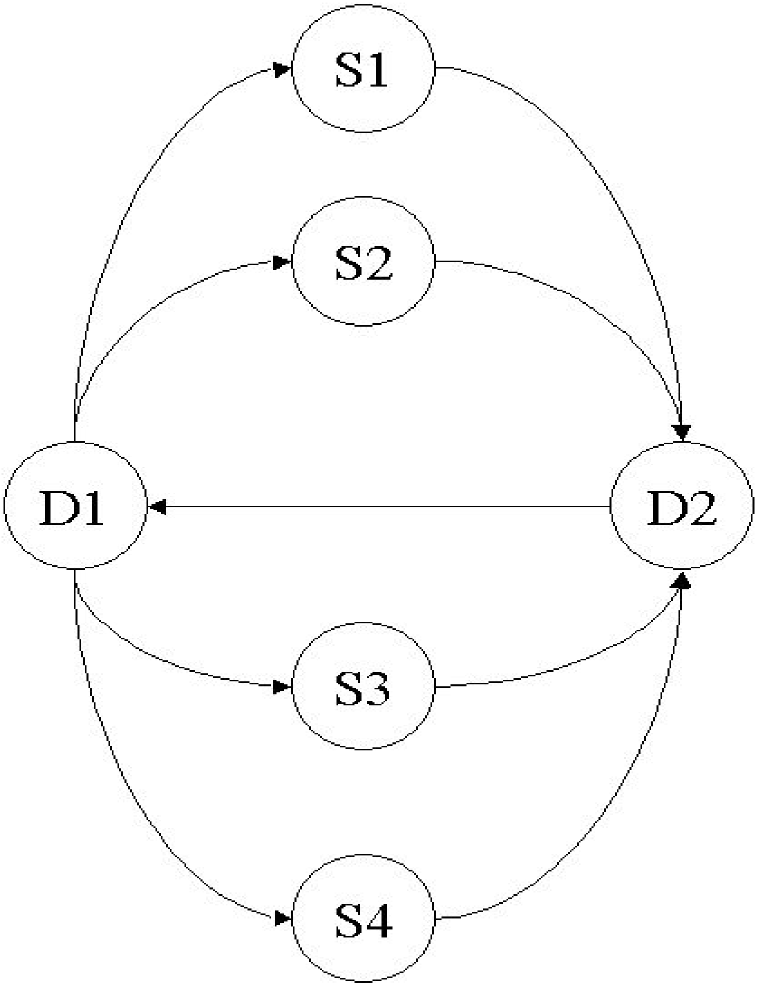


Figure 7.1: Ergodic HMM topology for a 4 states HMM that represents 4 speakers. States D1 and D2 are dummy states (non emitting) that represents the initial and the final state always visited

### 7.2.2 Duration constraint

A common problem in speech modeling consists in explicitly identifying the state duration density (see e.g. [42]) in order to include it in the HMM formulation. This approach has been largely used in speech recognition framework. As pointed out in [84], building a speaker model using very few frames (in the limit case just one frame) results in very poor clustering quality and for this reason an explicit duration per state must be considered. The difference with the speech recognition case is important anyway: in speech we can expect a given phoneme to have an intrinsic duration density determined by the generation mechanism of the phoneme; in the audio indexing task there is absolutely no information on how long the speaker is talking and how many data will be provided per speaker; for this reason it is useless to define a density duration model. The only point that matters in this case is to assume a minimum quantity of frames to estimate the speaker model in a robust way . Common choices are imposing a fixed segment length (see [84],[83]) of $D$ frames that are assumed generated by the same speaker or imposing a minimum duration constraint per speaker as in [3] in order to avoid spare solutions in which data are clustered using extremely small blocks.

In our system we will use a sequence of blocks of fixed length $D$ as in [84] in order to obtain a robust and non-spare solution. Mathematically speaking the observation sequence $Y = \{y_1, \ldots, y_t\}$ can be rewritten as a sequence of blocks of length $D$ designated by $O_t$ where $O_t = \{O_{t1}, \ldots, O_{tp}, \ldots, O_{tD}\}$. Each block $O_t$ is assumed to be generated entirely by the same speaker (state) $s_j$ following the law $p(O_t|s_j)$. The log-likelihood of the sequence $Y$ can be rewritten as:

$$log\, p(Y) = log\, p(\{O_t\}) = \sum_{t=1}^{T} log \sum_{j=1}^{S} p(O_t|s_j) \tag{7.3}$$

### 7.2.3 Emission probabilities

Most speaker indexing systems use Gaussian Mixture Models as choice for modeling the emission probability $p(O_t|s_j)$ but other proposed solutions are for instance vector quantization ([33]) or Self Organizing Maps (SOM) ([83]).

In our system we will consider GMM in order to model speaker emission probability and in section 7.5 we will experimentally verify how VB learning can solve many problems of the classical GMM based indexing systems. The probability $p(O_t|s_j)$ is modeled by an M component GMM with means $\mu_{ij}$, covariance matrix $\Gamma_{ij}$ and weights $\beta_{ij}$ where $i = \{1, \ldots, M\}$ designates the $i$th component of the $j$th state.

Under the hypothesis that frames in block $O_t$ are independent the likelihood can be written as:

$$p(O_t) = \prod_{p=1}^{D} \sum_{i=1}^{M} \beta_{ij} N(O_{tp}|\mu_{ij}, \Gamma_{ij}) \tag{7.4}$$

The number of Gaussian components is fixed for all speakers to the value of $M$, but we will show later that VB learning can prune out extra unused components.

It is possible now to write the log-likelihood for the observation sequence $Y$ as:

$$log\, p(Y|\theta) = \sum_{t=1}^{T} log \sum_{j=1}^{S} \alpha_j\, P(O_t|s_j) = \sum_{t=1}^{T} log \sum_{j=1}^{S} \alpha_j \, (\prod_{p=1}^{D} \sum_{i=1}^{M} \beta_{ij} N(O_{tp}|\mu_{ij}, \Gamma_{ij}) \tag{7.5}$$

where $\theta = \{\alpha_j, \beta_{ij}, \mu_{ij}, \Gamma_{ij}\}$. Up to this moment we have assumed that the state (speaker) number $S$ and the Gaussian component number $M$ are fixed and a priori known. Generally this is not the case in speaker indexing systems and a model selection criterion is necessary for determining the real number of speakers.

## 7.3 Model learning

In this section we show how it is possible to learn the model defined in section 7.2.1 according to the three criteria considered in this thesis: ML, MAP and VB. The hypothesis of known value of $S$ speakers is first considered here.

### 7.3.1 Maximum Likelihood Learning

The log-likelihood (7.5) is the objective function for the maximum likelihood criterion. Two kinds of hidden variables must be considered in this case: a variable $x_t$ that designates which speaker (or state) generated the block $O_t$ and an hidden variable $z_{tp}$ conditioned on the value of $x_t$ that denotes which Gaussian component generated the observation $O_{tp}$. $x_t$ can assume values in the interval $\{1, \dots, S\}$ while $z_{tp}$ can assume values in the interval $\{1, \dots, M\}$ and *must* always be interpreted as conditional to the value of $x_t$.

A simple application of the EM algorithm that furthermore considers duration constraint $D$ can provide a parameter estimation. Let us designate with $\gamma_{x_t=j}$ the probability of variable $x_t$ to be equal to state $j$ and with $\gamma_{z_{tp}=i|x_t=j}$ the probability of variable $z_{tp}$ to be equal to component $i$ conditioned on the value of $x_t$. The E-step is :

$$\gamma_{x_t=j} = P(x_t = j|O_t) = \frac{\alpha_j P(O_t|s_j)}{\sum_j \alpha_j P(O_t|s_j)} \tag{7.6}$$

$$\gamma_{z_{tp}=i|x_t=j} = P(z_{tp} = i|x_t = j, O_{tp}) = \frac{\beta_{ij} N(O_{tp}, \mu_{ij}, \Gamma_{ij})}{\sum_{i=1}^{D} \beta_{ij} N(O_{tp}, \mu_{ij}, \Gamma_{ij})} \tag{7.7}$$

Once $\gamma_{x_t=j}$ and $\gamma_{z_{tp}=i|x_t=j}$ are estimated, it is possible to update the parameters in the following M-step:

$$\alpha_j = \sum_{t=1}^{T} \gamma_{x_t=j}/T \tag{7.8}$$

$$\beta_{ij} = \frac{\sum_{t=1}^{T}\sum_{p=1}^{D} \gamma_{x_t=j}\gamma_{z_{tp}=i|x_t=j}}{\sum_{t=1}^{T}\sum_{p=1}^{D} \gamma_{x_t=j}} \tag{7.9}$$

$$\mu_{ij} = \frac{\sum_{t=1}^{T}\sum_{p=1}^{D} \gamma_{x_t=j}\gamma_{z_{tp}=i|x_t=j}O_{tp}}{\sum_{t=1}^{T}\sum_{p=1}^{D} \gamma_{x_t=j}\gamma_{z_{tp}=i|x_t=j}} \tag{7.10}$$

$$\Gamma_{ij} = \frac{\sum_{t=1}^{T}\sum_{p=1}^{D} \gamma_{x_t=j}\gamma_{z_{tp}=i|x_t=j}(O_{tp}-\mu_{ij})^T(O_{tp}-\mu_{ij})}{\sum_{t=1}^{T}\sum_{p=1}^{D} \gamma_{x_t=j}\gamma_{z_{tp}=i|x_t=j}} \tag{7.11}$$

It is easy to identify in the E-step (7.6-7.7) and in the M-step (7.8-7.11) a simple extension of the EM for GMM in order to handle two levels of mixtures: one related to the speaker model and one related to the Gaussian; in fact the model is now reduced to a simple mixture of mixtures.

### 7.3.2 Maximum a Posteriori Learning

In the MAP framework, the objective function is constituted by $log\, P(Y,\theta) = log\, P(Y|\theta) + log\, P(\theta)$ where $log\, P(Y|\theta)$ has the same form as 7.5 and $log\, P(\theta)$ represents the prior over parameters. The natural choice for prior parameter distributions is a set of distributions that belongs to the conjugate family in order to obtain posterior distributions with the same form of priors i.e.:

$$\begin{aligned} P(\alpha_j) = Dir(\lambda_{\alpha 0}) \quad P(\beta_{ij}) = Dir(\lambda_{\beta 0}) \\ P(\mu_{ij}|\Gamma_{ij}) = N(\rho_0, \xi_0\Gamma_{ij}) \quad P(\Gamma_{ij}) = W(\nu_0, \Phi_0) \end{aligned} \tag{7.12}$$

where as always $Dir()$ designates a Dirichlet distribution, $N()$ a Normal distribution and $W()$ a Wishart distribution (see appendix A).

The E-step will be the same as before i.e. equations (7.6-7.7). In the M-step we will have:

$$\bar{\mu}_{ij} = \frac{\sum_{t=1}^{T}\sum_{p=1}^{D} \gamma_{x_t=j}\gamma_{z_{tp}=i|x_t=j}O_{tp}}{\sum_{t=1}^{T}\sum_{p=1}^{D} \gamma_{x_t=j}\gamma_{z_{tp}=i|x_t=j}} \tag{7.13}$$

$$\bar{\Gamma}_{ij} = \frac{\sum_{t=1}^{T}\sum_{p=1}^{D} \gamma_{x_t=j}\gamma_{z_{tp}=i|x_t=j}(O_{tp}-\mu_{ij})^T(O_{tp}-\mu_{ij})}{\sum_{t=1}^{T}\sum_{p=1}^{D} \gamma_{x_t=j}\gamma_{z_{tp}=i|x_t=j}} \tag{7.14}$$

$$N_{ij} = \sum_{t=1}^{T}\sum_{p=1}^{D} \gamma_{x_t=j}\gamma_{z_{tp}=i|x_t=j} \tag{7.15}$$

$$N_j = \sum_{t=1}^{T} \gamma_{x_t=j} \tag{7.16}$$

and finally updated hyperparameters of posterior probability distributions can be computed:

$$\lambda_{\alpha_j} = N_j + \lambda_{\alpha 0} \tag{7.17}$$

$$\lambda_{\beta_{ij}} = N_{ij} + \lambda_{\beta 0} \tag{7.18}$$

$$\rho_{ij} = \frac{N_{ij}\,\bar{\mu}_{ij} + \xi_0\,\rho_0}{N_{ij} + \rho_0} \tag{7.19}$$

$$\xi_{ij} = N_{ij} + \xi_0 \tag{7.20}$$

$$\Phi_{ij} = N_{ij}\,\bar{\Gamma}_{ij} + \frac{N_{ij}\xi_0(\mu_{ij} - \rho_0)(\mu_{ij} - \rho_0)^T}{N_{ij} + \rho_0} + \Phi_0 \tag{7.21}$$

$$\nu_{ij} = N_{ij} + \nu_0 \tag{7.22}$$

The MAP parameter estimation can be obtained considering the maximum of posterior distributions:

$$\alpha_j = \frac{\lambda_{\alpha_j} - 1}{\sum_j \lambda_{\alpha_j} - 1} \tag{7.23}$$

$$\beta_{ij} = \frac{\lambda_{\beta_{ij}} - 1}{\sum_i \lambda_{\beta_{ij}} - 1} \tag{7.24}$$

$$\mu_{ij} = \rho_{ij} \tag{7.25}$$

$$\Gamma_{ij} = \frac{\bar{\Gamma}_{ij}}{\nu_{ij} - d} \tag{7.26}$$

where $d$ is the vector dimension.

### 7.3.3 Variational Bayesian Learning

In this section we consider the VB learning of the same model. In this case there will be no explicit parameter estimation but simply parameter variational posterior distribution estimation. In this section we will develop the VBEM for our specific model, show how it is possible to compute the free energy (because of model selection purposes) and finally derive the optimal hyperparameters.

#### VBEM

If prior distributions are defined as 7.12 the model belongs to the conjugate-exponential family and according to results of section 3.3.1 the VBEM algorithm can be applied.

In the E-step the joint variational distribution of both hidden variables must be considered $q(x_t, z_{tp})$ i.e.

$$q(x_t, z_{tp}) = q(x_t)q(z_{tp}|x_t) \propto exp\{< log\alpha_{x_t} > \\ + < log\beta_{x_t,z_{tp}} > + < log\,P(O_{tp}|x_t, z_{tp}) >\} \tag{7.27}$$

where with < . > we designate the expected value w.r.t. the variational distribution. Applying the variational E step we obtain the variational estimation of hidden variables designated by $\tilde{\gamma}_{x_t=j}$ and $\tilde{\gamma}_{z_{tp}=i|x_t=j}$:

$$\begin{aligned}
\tilde{\gamma}^*_{z_{tp}=i|x_t=j} &= \tilde{\beta}_{ij}\,\tilde{\Gamma}^{1/2}_{ij}\,exp\{-E\}\,exp\{\frac{-d}{2\nu_{ij}}\} \\
with\ E &= \frac{1}{2}(O_{tp}-\rho_{tp})^T\bar{\Gamma}_{ij}(O_{tp}-\rho_{tp}) && (7.28) \\
\tilde{\gamma}_{z_{tp}=i|x_t=j} &= q(\gamma_{z_{tp}}=i|\gamma_{x_t}=j) = \frac{\tilde{\gamma}^*_{z_{tp}=i|x_t=j}}{\sum_i \tilde{\gamma}^*_{z_{tp}=i|x_t=j}} && (7.29) \\
\tilde{\gamma}^*_{x_t=j} &= \tilde{\alpha}_j \prod_{p=1}^{D}\sum_{i=1}^{M}\tilde{\gamma}^*_{z_{tp}=i|x_t=j} && (7.30) \\
\tilde{\gamma}_{x_t=j} &= q(\gamma_{x_t}=j) = \frac{\tilde{\gamma}^*_{x_t=j}}{\sum_j \tilde{\gamma}^*_{x_t=j}} && (7.31)
\end{aligned}$$

with $d$ as before being the dimension of the observation vector. Parameters expected values can be computed as follows:

$$\begin{aligned}
log\,\tilde{\alpha}_j &= \Psi(\lambda_{\alpha_j}) - \Psi(\sum_j \lambda_{\alpha_j}) && (7.32) \\
log\,\tilde{\beta}_{ij} &= \Psi(\lambda_{\beta_{ij}}) - \Psi(\sum_j \lambda_{\beta_{ij}}) && (7.33) \\
log\,\tilde{\Gamma}_{ij} &= \sum_{i=1}^{g}\Psi((\nu_{ij}+1-i)/2) - log\,|\Phi_{ij}| + glog2 && (7.34) \\
\bar{\Gamma}_{ij} &= \nu_{ij}\Phi^{-1}_{ij} && (7.35)
\end{aligned}$$

where $\Psi$ is the digamma function ([1]).

In the M step, posterior distributions will have the same form of prior distributions. Reestimation formulas for parameters are given by:

$$\begin{aligned}
\alpha_j &= \frac{\sum_{t=1}^{T}\tilde{\gamma}_{x_t=j}}{T} && (7.36) \\
\beta_{ij} &= \frac{\sum_{t=1}^{T}\sum_{p=1}^{D}\tilde{\gamma}_{x_t=j}\tilde{\gamma}_{z_{tp}=i|x_t=j}}{\sum_{t=1}^{T}\sum_{p=1}^{D}\tilde{\gamma}_{x_t=j}} && (7.37) \\
\mu_{ij} &= \frac{\sum_{t=1}^{T}\sum_{p=1}^{D}\tilde{\gamma}_{x_t=j}\tilde{\gamma}_{z_{tp}=i|x_t=j}O_{tp}}{\sum_{t=1}^{T}\sum_{p=1}^{D}\tilde{\gamma}_{x_t=j}\tilde{\gamma}_{z_{tp}=i|x_t=j}} && (7.38) \\
\Gamma_{ij} &= \frac{\sum_{t=1}^{T}\sum_{p=1}^{D}\tilde{\gamma}_{x_t=j}\tilde{\gamma}_{z_{tp}=i|x_t=j}(O_{tp}-\mu_{ij})^T(O_{tp}-\mu_{ij})}{\sum_{t=1}^{T}\sum_{p=1}^{D}\tilde{\gamma}_{x_t=j}\tilde{\gamma}_{z_{tp}=i|x_t=j}} && (7.39)
\end{aligned}$$

and hyperparameters reestimation formulas are given by:

$$\lambda_{\alpha_j} = N_j + \lambda_{\alpha 0} \tag{7.40}$$

$$\lambda_{\beta_{ij}} = N_{ij} + \lambda_{\beta 0} \tag{7.41}$$

$$\rho_{ij} = \frac{N_{ij}\,\mu_{ij} + \xi_0\,\rho_0}{N_{ij} + \rho_0} \tag{7.42}$$

$$\xi_{ij} = N_{ij} + \xi_0 \tag{7.43}$$

$$\Phi_{ij} = N_{ij}\,\Gamma_{ij} + \frac{N_{ij}\xi_0(\mu_{ij} - \rho_0)(\mu_{ij} - \rho_0)^T}{N_{ij} + \rho_0} + \Phi_0 \tag{7.44}$$

$$\nu_{ij} = N_{ij} + \nu_0 \tag{7.45}$$

where $N_{ij} = \sum_{t=1}^{T}\sum_{p=1}^{D}\tilde{\gamma}_{x_t=j}\tilde{\gamma}_{z_{tp}=i|x_t=j}$ and $N_j = \sum_{t=1}^{T}\tilde{\gamma}_{x_t=j}$.

Finally variational posterior distributions are defined as:

$$q(\{\alpha_j\}) = Dir(\{\lambda_{\alpha_j}\}) \tag{7.46}$$

$$q(\{\beta_{ij}\}) = Dir(\{\lambda_{\beta_{ij}}\}) \tag{7.47}$$

$$q(\mu_{ij}|\Gamma_{ij}) = N(\rho_{ij}|\xi_{ij}\Gamma_{ij}) \tag{7.48}$$

$$q(\Gamma_{ij}) = W(\nu_{ij}, \Phi_{ij}) \tag{7.49}$$

### Free energy computation

We show in this section that it is possible to derive a close form for the variational free energy (3.7) when we consider a model like (7.5). The importance of a closed form for the free energy expression consists, as previously described, in the fact that it can be used as a model selection criterion that can be used instead of other model selection criterion (e.g. BIC, MML, etc.).

Let us re-write expression (3.7) for the model we are considering:

$$\begin{aligned} F(\theta,\gamma) &= \int d\theta d\gamma q(\gamma)q(\theta)log[p(O,\gamma,\theta)/q(\gamma)q(\theta)] \\ &= < log\,\frac{p(O,\gamma|\theta)}{q(\gamma)} >_{\gamma,\theta} - D[q(\theta)||p(\theta)] \end{aligned} \tag{7.50}$$

where accordingly to our previous discussion, hidden variable set $\gamma = \{\gamma_{z_{tp}|x_t}, \gamma_{x_t}\}$ consists of two variables: one referring to the cluster and the other referring to the component and where $q(\gamma)$ is the variational distribution over hidden variables. Hidden variables are actually discrete variables.

Considering the factorization $p(O,\gamma|\theta) = p(O|\gamma,\theta)p(\gamma|\theta)$ we can rewrite (7.50) as sum of three different terms:

$$\begin{aligned} F(\theta,\gamma) = &\int d\theta d\gamma q(\gamma)q(\theta)[log(p(O|\gamma,\theta)) + log(p(\gamma|\theta))] + \\ &- \int d\theta d\gamma q(\gamma)q(\theta)logq(\gamma) - D[q(\theta)||p(\theta)] \end{aligned} \tag{7.51}$$

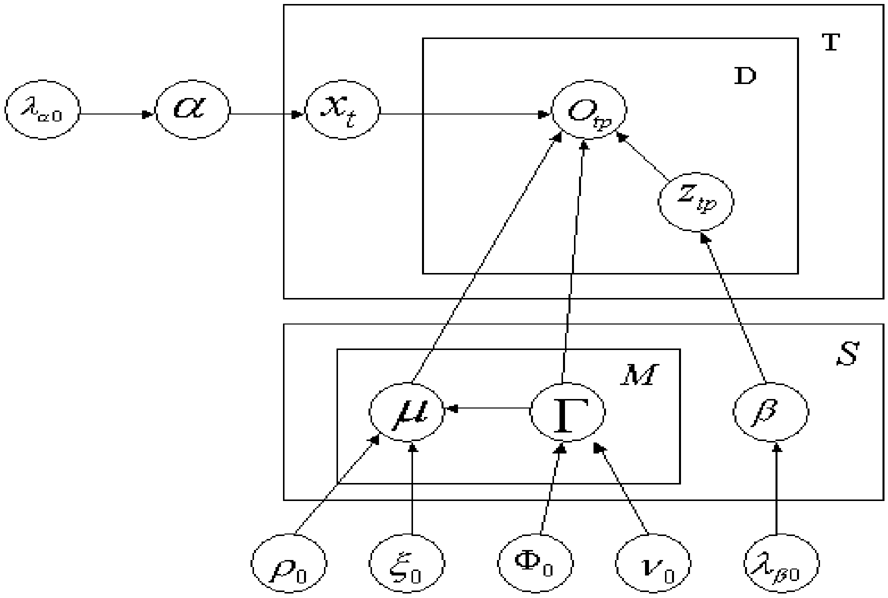


Figure 7.2: Direct graph that represent the Bayesian model for speaker clustering. $x_t$ and $z_{tp}$ are hidden variables. The box indicates that the elements inside must be repeated a number of times equal to the value in the top-right corner.

Considering the fact that $q(\gamma_{z_{tp}} = i, \gamma_{x_t} = j) = q(\gamma_{x_t} = j)\, q(\gamma_{z_{tp}} = i|\gamma_{x_t} = j)$ and considering the same notation as before:

$$\gamma_{z_{tp}=i|x_t=j} = q(\gamma_{z_{tp}} = i|\gamma_{x_t} = j) \quad (7.52)$$

$$\gamma_{x_t=j} = q(\gamma_{x_t} = j) \quad (7.53)$$

Coming back to expression (7.51) we will consider separately the three terms.

- the first term is:

$$\int d\theta d\gamma q(\gamma) q(\theta)[log(p(O|\gamma,\theta)) + log(p(\gamma|\theta))] \quad (7.54)$$

  Because of the fact hidden variables are actually discrete variables, integral w.r.t. $\gamma$ becomes a sum over states and mixtures. Let us explicit expression (7.54) w.r.t $T$,$D$ and hidden variables:

$$\sum_{t=1}^{T}\sum_{p=1}^{D}\sum_{j=1}^{S}\sum_{i=1}^{M} \gamma_{z_{tp}=i|x_t=j}\gamma_{x_t=j} \int d\theta q(\theta)[$$
$$log(p(O_{tp}|,\theta\gamma_{z_{tp}=i,x_t=j})) + log(p(\gamma_{z_{tp}=i,x_t=j})|\theta))] \quad (7.55)$$

Considering now the factorization $p(\gamma_{z_{tp}=i,x_t=j}|\theta) = p(\gamma_{x_t=j}|\theta)p(z_{tp} = i|x_t = j,\theta) = \alpha_j\beta_{ij}$ and using the previously defined quantity $\tilde{\gamma}^*_{z_{tp}=i|x_t=j}$, it is possible to rewrite (7.55):

$$\sum_{t=1}^{T}\sum_{j=1}^{S}\gamma_{x_t=j}[log\,\tilde{\alpha}_j + \sum_{p=1}^{D}\sum_{i=1}^{M}\gamma_{z_{tp}=i|x_t=j}\,log\,\tilde{\gamma}^*_{z_{tp}=i|x_t=j}] \tag{7.56}$$

All elements in (7.56) are explicit and known.

- Let us now consider the second term in (7.51):

$$\int d\theta d\gamma q(\gamma)q(\theta)logq(\gamma) = \int q(\gamma)logq(\gamma)d\gamma \tag{7.57}$$

Let us explicit this expression w.r.t. time, duration and hidden variables. The result is:

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{p=1}^{D}\sum_{j=1}^{S}\sum_{i=1}^{M}\gamma_{x_t=j}\gamma_{z_{tp}=i,x_t=j}\,log\,[\gamma_{x_t=j}\gamma_{z_{tp}=i|x_t=j}] =\\ &= \sum_{t=1}^{T}\sum_{j=1}^{S}\{\gamma_{x_t=j}[log\,\gamma_{x_t=j} + \sum_{p=1}^{D}\sum_{i=1}^{M}\gamma_{z_{tp}=i|x_t=j}log\,\gamma_{z_{tp}=i|x_t=j}]\}\end{aligned} \tag{7.58}$$

Again in (7.58) all terms are explicit and known.

- The last term to consider is the KL divergence between posterior distributions and prior distributions. Because of independence between parameter distributions, it is possible to write:

$$\begin{aligned}D[q(\theta)||p(\theta)] &= D(Dir(\lambda_{\alpha j})||Dir(\lambda_{\alpha 0}))\\ &+\sum_j D(Dir(\lambda_{\beta ij})||Dir(\lambda_{\beta 0}))\\ &+\sum_j\sum_i D(N(\rho_{ij},\xi_{ij}\nu_{ij}\Phi_{ij}^{-1})||N(\rho_0,\xi_0\nu_{ij}\Phi_{ij}^{-1}))\\ &+\sum_i\sum_j D(W(\nu_{ij},\Phi_{ij})||W(\nu_0,\Phi_0))\end{aligned} \tag{7.59}$$

A close form for all KL divergence in 7.59 can be found (a useful summary can be found in [108] or Appendix A).

### Optimizing prior distributions

In this section we derive equation that allows to optimize hyperparameters given posterior distributions.

The criterion to optimize hyperparameters is the bound on the marginal likelihood obtained averaging all possible models. Considering prior over models $p(m)$ and variational posterior $q(m)$ we can write the lower bound:

$$log\, p(Y) \geq F = \sum_m q(m) \cdot [F_m + log\frac{p(m)}{q(m)}] \tag{7.60}$$

Terms that depend on priors are $F_m$ and variational posterior over models $q(m)$ (see 3.61). Our goal is to find optimal hyperparameters denoted with $h = \{\lambda_{\alpha 0}, \lambda_{\beta 0}, \rho_0, \xi_0, \nu_0, \Phi_0\}$ that optimize F. Deriving F w.r.t $h$ and $q(m)$ we obtain:

$$\frac{\partial F}{\partial h} = \sum_m q(m) \frac{\partial F_m}{\partial h} = 0 \tag{7.61}$$

$$\frac{\partial F}{\partial q(m)} = [F_m + log\frac{p(m)}{q(m)} - 1] = 0 \tag{7.62}$$

Solving we obtain:

$$q(m) \propto e^{F_m} \cdot p(m) \tag{7.63}$$

$$\sum_m q(m) \cdot [-\frac{\partial KL(p(\theta|m)||q(\theta|m))}{\partial h}] = 0 \tag{7.64}$$

Partial derivatives in 7.64 must consider all different parameters KL divergences i.e. all terms of equation 7.59. Let us consider separately each term:

First term: $D(Dir(\lambda_{\alpha j})||Dir(\lambda_{\alpha 0}))$. Equation (7.64) gives:

$$\sum_m q(m) \frac{\partial D(q(\lambda_{\alpha j}|m)||q(\lambda_{\alpha 0}))}{\partial \lambda_{\alpha 0}} = \sum_m q(m) \frac{\partial D(Dir(\lambda_{\alpha jm})||Dir(\lambda_{\alpha 0}))}{\partial \lambda_{\alpha 0}} = 0 \tag{7.65}$$

Developing and using Dirichlet distribution expected values we obtain:

$$\frac{\partial}{\partial \lambda_{\alpha 0}} \sum_m q(m)[log\Gamma(S_m \lambda_{\alpha 0}) - S_m log\Gamma(\lambda_{\alpha 0}) +$$
$$\sum_{j=1}^{S_m} \{(\lambda_{\alpha 0} - 1) \cdot (\Psi(\lambda_{\alpha j}) - \Psi(\sum \lambda_{\alpha j}))\}] = 0 \tag{7.66}$$

where $S_m$ denotes the number of speaker (i.e. clusters) of the $m$th system.

Deriving now 7.66 w.r.t $\lambda_{\alpha 0}$ we get;

$$\sum_m q(m)[S_m \cdot \Psi(S_m \lambda_{\alpha 0}) - S_m \Psi(\lambda_{\alpha 0}) + \sum_j^{S_m} \{\Psi(\lambda_{\alpha j}) - \Psi(\sum \lambda_{\alpha j})\}] = 0 \quad (7.67)$$

In order to solve 7.67 a Newton-Raphson method can be used. Second order derivate uses derivative of the Psi function called polygamma function (see [1]).

Second term: $\sum_j D(Dir(\lambda_{\beta ij}) || Dir(\lambda_{\beta 0}))$. Proceeding in an analogous way we can write:

$$\frac{\partial}{\partial \lambda_{\beta 0}} \sum_m q(m)[S_m ln\Gamma(M_m \lambda_{\beta 0}) - S_m M_m ln\Gamma(\lambda_{\beta 0}) +$$

$$(\lambda_{\beta 0} - 1) \cdot \sum_j^{S_m} \sum_i^{M_m} \{\Psi_{\lambda_{\beta ij}} - \Psi \sum_i \lambda_{\beta ij}\}] = 0 \qquad (7.68)$$

Deriving (7.68) w.r.t. $\lambda_{\beta 0}$

$$\sum_m q(m)[S_m M_m \Psi(M_m \lambda_{\beta 0}) - S_m M_m \Psi(\lambda_{\beta 0}) + \sum_j^{S_m} \sum_i^{M_m} (\Psi(\lambda_{\beta ij}) - \Psi(\sum \lambda_{\beta ij}))] = 0 \qquad (7.69)$$

Again Newton-Raphson method can be used to find out solution of the equation.

Third term: $\sum_j \sum_i D(N(\rho_{ij}, \xi_{ij}\nu_{ij}\Phi_{ij}^{-1}) || N(\rho_0, \xi_0 \nu_{ij} \Phi_{ij}^{-1}))$. In the case of diagonal covariance matrix we can further simplify the KL divergence using product over the dimension $l$.

$$\sum_j \sum_i D(N(\rho_{ij}, \xi_{ij}\nu_{ij}\Phi_{ij}^{-1}) || N(\rho_0, \xi_0 \nu_{ij} \Phi_{ij}^{-1})) =$$

$$= \sum_j \sum_i \sum_l D(N(\rho_{ijl}, \xi_{ij}\nu_{ij}\Phi_{ijl}^{-1}) || N(\rho_{0l}, \xi_0 \nu_{ij} \Phi_{ijl}^{-1})) \qquad (7.70)$$

Averaging over models we have:

$$\sum_m q(m) \sum_i \sum_j \int N(\rho_{ij}, \xi_{ij}\nu_{ij}\Phi_{ij}^{-1}) \sum_l [log(\xi_0 \nu_{ij} \Phi_{ijl}^{-1}) - (\mu_{ijl} - \rho_{0l})^2 \cdot \xi_0 \nu_{ij} \Phi_{ijl}^{-1}] \qquad (7.71)$$

Deriving 7.71 w.r.t. $\rho_0$ (or $\rho_{0l}$ component by component) we have:

$$\sum_m q(m) \sum_i \sum_j \int N(\rho_{ij}, \xi_{ij}\nu_{ij}\Phi_{ij}^{-1}) \sum_l [(\mu_{ijl} - \rho_{0l})] = 0 \qquad (7.72)$$

that gives

$$\rho_{0l} = \sum_m q(m) \frac{1}{S_m M} \sum_i \sum_j < \mu_{ijl} > \tag{7.73}$$

Deriving 7.71 now w.r.t. $\xi_0$ and solving we have:

$$\sum_m q(m) \sum_i \sum_j \sum_l \xi_0 = \sum_m q(m) \sum_i \sum_j \sum_l < (\mu_{ijl} - \rho_{0l}) \cdot (\mu_{ijl} - \rho_{0l}) > \Gamma_{ijl}^{-1} \tag{7.74}$$

That gives

$$\xi_0 = \frac{\sum_m q(m) \sum_i \sum_j \sum_l < (\mu_{ijl} - \rho_{0l}) \cdot (\mu_{ijl} - \rho_{0l}) > \Gamma_{ijl}^{-1}}{\sum_m q(m) S_m M} \tag{7.75}$$

The solution is a close form solution in this case and there is no need for numerical methods.

Fourth term: $\sum_i \sum_j D(W(\nu_{ij}, \Phi_{ij}) || W(\nu_0, \Phi_0))$.

We can write averaging over all models:

$$\sum_m q(m) \sum_i \sum_j D(W(\nu_{ij}, \Phi_{ij}) || W(\nu_0, \Phi_0)) = 0 \tag{7.76}$$

Deriving 7.76 w.r.t $\nu_0$ and $\Phi_0$ we find first moment and log first moment of the Wishart distribution.

$$\sum_m q(m) [\sum_i \sum_j \{\sum_{l=1}^{L} \Psi(\nu_0 + 1 - l) - 2ln2 + ln\Phi_0\}] = \sum_m q(m) \sum_i \sum_j < log\Gamma_{ij} > \tag{7.77}$$

$$\sum_m q(m) \sum_i \sum_j \nu_0 \Phi_0^{-1} = \sum_m q(m) \sum_i \sum_j < \Gamma_{ij} > \tag{7.78}$$

In the case of diagonal $\Phi_0$ matrix it is possible to reduce the first equation to a one-variable expression, and solve it using Newton-Raphson.

### 7.3.4 Optimal segmentation

Once the model is learned, it is a straightforward task to obtain the segmentation applying the Viterbi algorithm described in sections 2.2.2. Viterbi segmentation find the best sequence of speakers that generated observations we will designate it with $S_{best}$. The duration constraint can be easily handled with minor modifications in the algorithm.

Anyway an important difference must be pointed out between the MAP/ML methods and the VB method. In fact while MAP and ML provide parameters, VB provides distributions over parameters. In practice it means that the VB segmentation must be obtained integrating out probabilities w.r.t. variational distributions.

In other words while MAP/ML during the Viterbi algorithm consider the emission probability $p(x_t = s_j)$, the variational Bayesian Viterbi algorithm considers $q(x_t = s_j)$ i.e. expression $\tilde{\gamma}_{z_{tp}=i|x_t=j}$ (see equation 7.29).

The complexity overload is minimum compared with the classical algorithm in the sense that parameter expected values are used instead of parameters.

## 7.4 Estimation of speaker number

In section 7.2 we have assumed that the number of speakers (i.e. states in the ergodic HMM) is a priori known. In real problems, this is rarely the case and the cluster number $S$ must be estimated as another parameter of the model.

The most common choice is to turn the problem into a model selection problem. As already discussed in section 2.6, the observation likelihood of a given model will generally increase as long as the complexity of the model increases. For this reason the likelihood is not a useful quantity for the model selection purposes. The most used model selection criterion in speech processing is the Bayesian Information Criterion (see section 2.6.4) introduced for segmentation purposes in [31]. The popularity of the BIC is largely due to its simplicity. In speech processing manual tuning of the penalty term is generally required (see [127]).

Ideally the search algorithm in the model space should be an exhaustive search followed by a model selection. For instance if we suppose that in a given audio file there are no more than $S_{max}$ speakers, all possible models with a number of clusters in between $S_{max}$ and 1 should be learned and then the best model should be selected using for instance the BIC. This technique is rarely applicable for computational reasons and the search algorithm is reduced to a greedy search.

In next sections we will describe the search algorithm we used in our experiments.

### 7.4.1 Search algorithm

The greedy search algorithms generally used in the speaker clustering task are of two types: bottom-up algorithms or top-down algorithms (see [70],[71]). In the bottom-up algorithms, the observation sequence is initially considered belonging to one cluster. The algorithm progressively splits the file into smaller clusters until a termination criterion (e.g. a model selection criterion is met). On the other hand the top-down algorithms initially over-segment data in a number of clusters larger than the real one and successively merge couples of clusters until a termination criterion is met.

The top-down algorithm is generally preferred to the bottom-up because merging clusters is easier than splitting a cluster in more clusters. In our system we use a top-down algorithm. The first step consists so in over-clustering the audio file in a number of clusters hypothesized larger than the actual one.

### Initialization

Initial over-segmentation can be achieved in many different manners. Possible choices are:

- Finding the changing points (for review of those techniques see chapter 6) in the audio file and assigning each segment to a different cluster.

- Running a K-means algorithm with $S_{max}$ different clusters where $S_{max}$ is chosen to be larger than the real cluster number.

- Uniformly dividing the audio file in $S_{max}$ different segmentation and assigning each segment to a cluster (this solution is analogous to the flat start initialization in speech recognition [144]).

Once the over-segmentation is fixed, a GMM can be obtained for each cluster using the EM algorithm.

According to the type of over-segmentation different initialization of the transition probabilities can be realized. For example probabilities could be proportional to the number of frames in the cluster. Anyway experimentally we verified what already stated in [2]: the initial transition probabilities value have a very small impact on the final result.

### Reducing the cluster number

The search algorithm in the space of all possible clusters combination should move from the initial $S_{max}$ to the optimal value $S_{opt}$. As stated before the exhaustive search of all possible combinations is impossible and a greedy solution that progressively reduces the cluster number one by one is preferred. Again different solutions are possible:

- Merging iteratively two clusters according to some similarity measure or a model selection criterion until a termination condition is met.

- Eliminating iteratively a cluster according to some criterion and running again the EM algorithm using previous models as initialization until a termination criterion is met.

Of course the quality of the search algorithm is strongly influenced by criteria chosen for the cluster merging and for the termination condition. In next section we review possible criteria.

### Merging and Stopping criteria

Generally the criterion for merging and stopping different clusters consists in some kind of penalized score in order to consider the fact that the likelihood monotonically increases with the number of clusters. The penalty is generally considered under the form of a threshold.

In [123], the KL divergence between distributions of two different clusters is used as criterion for determining if they belong to the same speaker or not. The distance is compared with an heuristic threshold that must be determined case by case and there is no general rule for fixing it.

In [67], the use of the Gish-distance (see [55]) is proposed. Gish distance is basically a Log-likelihood ratio (LLR) based measure and was first used in the context of speaker identification [54]. Let us consider two clusters $s_1$ and $s_2$ with observations $y_1$ and $y_2$ and with models $\theta_1$ and $\theta_2$. Let us designate with $y = y_1 \cup y_2$ and with $\theta$ the model learned using $y$. The LLR criterion is:

$$LLR = L - L_1 - L_2 = \sum_{y_i} log\, p(y_i|\theta) - \sum_{y_{1i}} log\, p(y_{1i}|\theta_1) - \sum_{y_{2i}} log\, p(y_{2i}|\theta_2) \tag{7.79}$$

In order to make a decision if clusters $s_1$ and $s_1$ should be merged into $s$ the LLR is compared with an heuristic threshold. In [67], the Gish-distance is used together with a penalty term in order to avoid the over-segmentation.

The use of both LLR and KL is proposed in [124] together with the use of the cluster purity criterion that we will use later (see 7.6).

Another alternative measure that can be used is the (AHS) arithmetic harmonic sphericity measure proposed in [22] in the context of speaker recognition.

Anyway the most common criterion for speaker clustering purposes is the BIC introduced for this task in [31]. The BIC explicitly sets the threshold as the difference of number of free parameters between models $\{\theta_1, \theta_2\}$ and $\theta$:

$$BIC = L - L_1 - L_2 - 0.5\, P\, logN \lesseqgtr 0 \tag{7.80}$$

$$P = card\{\theta\} - card\{\theta_1\} - card\{\theta_2\} \tag{7.81}$$

As previously discussed the BIC is a rough approximation of the Bayesian integral and in order to be effective, the penalty term must be adjusted by an heuristic threshold (see [127]). In [45], two different penalty terms are considered in a BIC fashion: one that penalizes too many clusters and one that penalizes too many speakers in order to have a finer control on the clustering task with the drawback of two tuning parameters to set.

A solution to alleviate this problem proposed in [2] is to consider the model complexity in order to obtain $P = 0$. This solution partially solves the problem of the heuristic tuning of the penalty term but is anyway based on the asymptotic assumption of the BIC.

Other stopping criteria can be considered like stopping when occupancy goes beyond a certain value (see [70]) or when the reduction of gain in the data likelihood goes beyond a certain threshold (see [123]).

An alternative method proposed in [83] for reducing the number of clusters from $S_{max}$ to 1 uses no cluster merging but the cluster with the smaller occupancy is progressively removed and the EM is re-run using the remaining models as initialization. As long as the new model is close to the old, very few iterations are needed for the convergence. The BIC criterion is used to score the $S_{max}$ different segmentation and the final segmentations is the one with the highest score.

From one side it seems that this algorithm is more computationally expensive than the merging solution because all $S_{max}$ models must be obtained. On the other hand there is no additional cost in the search of possible clusters to merge: this cost can be extremely high if the initial over-segmentation is realized with a high number of clusters.

In this work we introduce as alternative to those measures the use of the Variational Free energy as threshold free criterion for model selection i.e.:

$$\delta F = F_s - F_{s_1} - F_{s_2} \lesseqgtr 0 \qquad (7.82)$$

where $F_s$ is the free energy of the merged clusters and $F_{s_1}$ and $F_{s_2}$ are free energies of unmerged clusters. The algorithm stops when there is no further gain in term of the considered criterion in merging clusters.

## 7.5 Experimental Framework

In this section we describe the experimental framework we used in the experiments section. The goal of simulations is to compare in the fairest way the Variational Bayesian learning used to learn models and to select best models with classical algorithms based on MAP/ML for the learning step and on BIC for the selection task. In the rest of this work we will refer as system 1 the system based on ML/BIC, as system 2 the system based on MAP/BIC and as system 3 the system based on the VB framework.

The model considered here is the ergodic HMM with emission probabilities modeled as GMM with M components described in section 7.2.1 and fixed duration constraint over states equal to $D$.

The over-segmentation is achieved supposing the initial state (speaker number) $S_{max}$ very large. The observation file is splitted into $S_{max}$ uniform segments and a GMM is trained using the EM algorithm. Transition probabilities are set as uniform. This initialization corresponds somehow to a flat start initialization. Systems 2 and 3 need as well an initialization for the prior distributions; different prior settings will be investigated.

Once the system is initialized the learning algorithms as described in sections 7.3.1 and 7.3.2,7.3.3 can be run in order to obtain a model estimation that will produce parameters in the first two cases and parameter distributions in the VB case. Segmentation is then obtained running a Viterbi algorithm. Score is computed following the metric described in section 7.6.

The cluster with the minimum occupancy i.e. the one with the smaller frames number is then removed and learning algorithms are run again with the full observation set and the reduced state number $S_{max}-1$ with the previously learned models as initialization. This procedure is iteratively repeated until the number of cluster is reduced to 1.

For model selection purposes a score must be computed for each intermediate model. In system 1 (ML learning) the score can be simply computed using the BIC:

$$BIC(S_i) = log\, P(Y|\theta_i) - 0.5 \cdot \lambda \cdot P \cdot log\, N \tag{7.83}$$

$$P = S_i\, M\,(1 + d + d) \tag{7.84}$$

where we made the assumption that covariance matrix are diagonal matrix.

For system 2 (MAP learning) the BIC must be modified in order to consider the use of a prior distribution in the following way:

$$BIC_{MAP}(S_i) = log\, P(Y|\theta_i) + log\, p(\theta) - 0.5 \cdot \lambda \cdot P \cdot log\, N \tag{7.85}$$

The additional term in 7.85 is the log probability of MAP parameters; in the asymptotical approximation i.e. $N \to \infty$ this term becomes negligible respect to other terms.

For system 3 the score for model selection is the same objective function as for the training i.e. the free energy computed as in section 7.3.3.

The selected speaker number is the one that holds the highest score. The choice of this algorithm is partially due to the fact that a curve of the score of the system w.r.t. the model selection score can be obtained for all possible speaker numbers. This is a convenient way to study how correlated the two considered criteria (model selection score and clustering score) are. Figure 7.3 summaries the clustering algorithm proposed.

## 7.6 Evaluation and Metric

In order to evaluate the quality of the clustering task we chosen the same metric as in [3] and [83]. As long as experiments were run on the same database, the use of the same evaluation criterion will simplify the comparison.

The metric is based on the computation of two quantities the *average cluster purity* (acp) and the *average speaker purity* (asp). Acp was introduced in [124] and consider the purity of a given cluster w.r.t. a given reference speaker. The motivation for the acp comes from the initial application of the speaker clustering task to adaptation purposes in speech recognition systems. In fact if a cluster has a high acp, it can be successfully used for adaptation purposes on a reference speaker. Anyway the drawback of this metric is that a simple way to obtain a high score is considering a large number of different clusters spreading the reference speakers in more clusters than needed. For this reason the *asp* has been introduced as the dual measure of the *acp* in the sense that considers the purity of a speaker in terms of clusters among whom his observations have been spread. Symmetrically to the *acp* it should be enough to consider a single cluster to obtain an *asp* equal to 1. The geometric mean between *asp* and *acp* is used to balance both effects.

In real speaker clustering application (e.g. broadcast news data) together with the speech data under different conditions, there are different types of non speech data like music, noises or others. Generally those segments are removed in a pre-segmentation step of speech/non-speech segmentation. Anyway we are interested as well in running experiments with the whole data in order to study the capacity of the algorithm to cluster the speech and the non-speech as well: thus no prior speech/non-speech segmentation is considered. To express this in the evaluation metric we use the same solution as in [83] in which non-speech data are considered as a "speaker" (i.e. an extra state) in the computation of the *acp* and ignored in the computation of the *asp*. In this way, non-speech spread over different clusters will not reduce the global score. The drawback of this choice is that the number of selected clusters may be larger than the real one because of many spurious non-speech data.

Mathematically speaking we define:

- $R$: number of speakers
- $S$: number of clusters
- $n_{ij}$: total number of frames in cluster $i$ spoken by speaker $j$
- $n_{.j}$: total number of frames spoken by speaker $j$, $j = 0$ means non-speech frames
- $n_{i.}$: total number of frames in cluster $i$

- $N$: total number of frames in the file

- $N_s$: total number of speech frames

It is now possible to define the purity of a cluster $p_{i.}$ and the purity of a speaker $q_{.j}$ as:

$$p_{i.} = \sum_{j=0}^{R} \frac{n_{ij}^2}{n_{i.}^2} \tag{7.86}$$

$$q_{.j} = \sum_{i=0}^{S} \frac{n_{ij}^2}{n_{.j}^2} \tag{7.87}$$

Averaging respectively w.r.t. all clusters and all speakers we obtain the *acp* and the *asp*:

$$acp = \frac{1}{N} \sum_{i=0}^{S} p_{i.}\, n_{i.} \tag{7.88}$$

$$asp = \frac{1}{N_s} \sum_{j=1}^{R} q_{.j}\, n_{.j} \tag{7.89}$$

Finally the geometric mean of *acp* and *asp* can be used as score for measuring the quality of the clustering resulting in:

$$K = \sqrt{acp \cdot asp} \tag{7.90}$$

### 7.6.1 Interpretation of results

In the following experiments we will report results in term of $K$; anyway it is interesting to study not only the value of the system as selected by the model selection criterion but also other scores that give a better insight on the clustering process. We will consider together with the selected system the best absolute score obtained by the system computed using the known labels and the score obtained initializing the system with the real number of clusters obtained by labels. In the ideal case those three results should be the same. Anyway we will experimentally verify that when the system is initialized with the true number of clusters the performance is always very poor compared to the best system. On the other hand the efficiency of the model selection criterion will be determined comparing the best result with the selected result.

## 7.7 About prior distributions

Bayesian approaches (both VB and MAP) require in the initialization step prior distributions over parameters. The choice of those distributions is a critical point for learning and for model selection. For tractability issues we have chosen those distributions in the conjugate-exponential family. Anyway the problem of setting the hyperparameters still remains.

In section 7.5, we will study in detail the impact of the hyperparameter choice on the final clustering score. We will consider two cases of priors: fixed priors and empirical Bayesian priors.

In the case of fixed priors we are interested in studying simply the impact of the strength of the prior on the final result. Because the number of hyperparameters can be large, we adopted in this case the same tying approach described in [46]. We suppose here a full tying; mathematically speaking we fix:

$$\lambda_{\alpha_0} = \lambda_{\beta_0} = \xi_0 = a_0 = \tau \tag{7.91}$$

$$B_0 = \tau \cdot I \tag{7.92}$$

$$\rho_0 = \bar{y} \tag{7.93}$$

where $I$ is the identity matrix and $\bar{y}$ is the mean of the observation sequence $Y$ and $\tau$ is a strength factor that determine the sharpness of the prior distribution. In the case of flat broad prior (that are supposed to be non-informative i.e. to add as few information as possible) the factor $\tau$ will be small and negligible w.r.t. the data contribution in the posterior distributions. On the other hand the strength of the prior distribution can be increased with a larger value of $\tau$. We are particularly interested in studying the impact of $\tau$ on the clustering performances.

Furthermore we are interested in verifying if the empirical optimal prior distributions estimated as proposed in section 7.3.3 correspond to the optimal clustering score function of the hyperparameters.

On the other hand, prior distributions can be estimated from a separate data set in order to capture important a priori information to be used as starting point for the system. In speech and speaker recognition this kind of approach based on the Maximum a Posteriori criterion is referred to as MAP adaptation (see for instance [46] and [113]). In speaker clustering approaches, the MAP adaptation is largely used (e.g. see [99]) because the speech provided by a single speaker is often not sufficient to generate a robust model. For this reason a background model is used to initialize prior distributions. Currently used MAP adaptation techniques as described in [113] are anyway far from the Bayesian spirit because they generate a set of updated parameters ignoring their distributions. Furthermore the best adaptation formulae are sometimes empirically derived and not motivated by the MAP framework (e.g. the GMM weight adaptation).

In an analogous way, VB system can be initialized with a background model. In this case the result will be a fully Bayesian framework (even if approximated) that will generate posterior distributions over parameters that can be used for segmentation. Because prior distributions have the same form in the MAP and the VB framework, it could be thought that the adaptation process will results in the same formula. However, two very big differences must be pointed out. First of all the E-step is completely different in the two systems: MAP consider parameters and VB parameter expected values resulting in different accumulator estimation. On the other hand VB does not adapt any parameter but adapt the variational posterior distribution over parameters. Those distributions then will be used for segmentation purposes.

## 7.8 Database description

Database we used in our experiments is the Hub4 1996 evaluation set. It is composed of 4 files with respectively 7, 13 15, and 20 speakers. Together with the speech data, a large amount of non-speech is present in those files except in file 2 where the non-speech part is almost negligible compared to the speech part. On the other hand, the speech part is not homogeneous and it includes narrow band and wide band speech. Furthermore speech is often corrupted by different noise sources like music or background speech. That makes the clustering task extremely challenging because the same speaker may be clustered into two different states if the background noise change.

Non-speech parts are very heterogeneous too with different types of non-speech events. There is no pre-segmentation speech/non-speech separation in our work and according to discussion of section 7.6, in the evaluation step we will augment the number of speaker by one in order to consider an extra cluster for non-speech. We will consider respectively 8, 14, 16 and 21 clusters for the four files.

## 7.9 Experiments

In this first experimental setup we initialized the three systems with $S_{max} = 30$ and $M = 15$. Duration constraint is set to $D = 200$ frames i.e. 2 seconds. The feature vector consists of 12 MFCC coefficients.

Table 7.1 shows results for system 1 (based on ML/BIC). The first line (ML known) is the result obtained initializing the system with the actual speaker number; the second line is the best result obtained over all possible speaker numbers. Last three lines are results for the selected systems as functions of the parameter $\lambda$.

As a general remark, the best system never corresponds in our experiments to the system initialized with the known speaker number. This is probably due to the finer coarse that can be obtained by merging small pure clusters. The BIC based selection task is extremely sensitive to the choice of $\lambda$. In fact for $\lambda = 1$ the selected system has very low performance. In Files 1 , 2 and 4 the best system is chosen when $\lambda = 3$ while in file 2, the best system is already selected with $\lambda = 2$. Visually the dependence of the system 1 to the threshold $\lambda$ in terms of performance (value of K) and in terms of inferred number of cluster is depicted in figures 7.4 and 7.5. Not only the best threshold is far from the ideal case ($\lambda = 1$) but it changes from file to file.

Figures (7.8-7.11) depict how the BIC is related with the clustering score function of $\lambda$. The green line represents K function of number of clusters. Blue, red and black lines represents the BIC value for $\lambda = \{1, 2, 3\}$. When $\lambda = 1$ or $\lambda = 2$ the BIC does not follow the K curve. Only when $\lambda = 3$ the BIC follows K closer. When the number of speaker is large the solution is extremely spiky; in fact when there are very few data for a given Gaussian in ML learning , the covariance matrix estimation gives extremely poor results. This problem does not exists in the Bayesian framework.

It must be noted also that in file 4 the selected score is smaller than the best system (second line). It means that in this case the BIC is completely inefficient in selecting the best system.

| File | File 1 | | | | File 2 | | | |
|---|---|---|---|---|---|---|---|---|
| | $N_c$ | acp | asp | K | $N_c$ | acp | asp | K |
| ML (known) | 8 | 0.61 | 0.89 | 0.74 | 14 | 0.76 | 0.66 | 0.71 |
| ML (best) | 10 | 0.80 | 0.88 | 0.84 | 15 | 0.83 | 0.69 | 0.76 |
| ML/BIC $\lambda = 1$ | 28 | 0.91 | 0.50 | 0.67 | 30 | 0.89 | 0.45 | 0.64 |
| ML/BIC $\lambda = 2$ | 19 | 0.90 | 0.67 | 0.77 | 20 | 0.86 | 0.57 | 0.70 |
| ML/BIC $\lambda = 3$ | 10 | 0.80 | 0.88 | 0.84 | 15 | 0.83 | 0.69 | 0.76 |
| File | File 3 | | | | File 4 | | | |
| | $N_c$ | acp | asp | K | $N_c$ | acp | asp | K |
| ML (known) | 16 | 0.76 | 0.77 | 0.77 | 21 | 0.72 | 0.66 | 0.69 |
| ML (best) | 15 | 0.79 | 0.84 | 0.82 | 12 | 0.64 | 0.82 | 0.72 |
| ML/BIC $\lambda = 1$ | 30 | 0.87 | 0.55 | 0.69 | 30 | 0.81 | 0.57 | 0.68 |
| ML/BIC $\lambda = 2$ | 15 | 0.79 | 0.84 | 0.82 | 21 | 0.71 | 0.65 | 0.68 |
| ML/BIC $\lambda = 3$ | 15 | 0.79 | 0.84 | 0.82 | 14 | 0.66 | 0.74 | 0.70 |

Table 7.1: Results on NIST 1996 HUB-4 evaluation test for speaker clustering,ML/BIC

The same experiment is now run using systems 2 and 3. Prior distributions for VB and MAP are set broad and non-informative with $\tau = 1E - 3$. Results for both systems are shown in tables 7.2 and 7.3. First of all, the best result and results obtained initializing the system with the actual number of speakers are the same for system 2 and system 3. This means that the amount of data is large enough to make the VB learning and the MAP learning converge to the same solution. On the other side they are different from the ML case; file 1 and 3 hold a score higher in the VB and MAP case while file 2 and file 4 hold a score higher in the ML case. Anyway the difference is never higher than $2-3\%$ in absolute showing that the performances of the three systems are almost equivalent with the considered setting.

Unsurprisingly the BIC applied to the MAP system has the same behavior observed in the ML system. The optimal value of $\lambda$ is equal to 3 for files 1, 2, 3, 4 . The theorical value of $\lambda = 1$ provides very poor results as before. Figures 7.6 and 7.7 plots the inferred number of speakers and the selected value of K as a function of the threshold value $\lambda$.

Figures (7.12-7.15) depict how BIC is related with the clustering score function of $\lambda$ for system 2 based on MAP learning. The green line represents K as a function of number of clusters. Blue, red and black lines represents the BIC value for $\lambda = \{1, 2, 3\}$. When $\lambda = 1$ or $\lambda = 2$ the BIC does not follow the K curve. Only when $\lambda = 3$ the BIC follows closer K as previously noticed for the ML/BIC system. Anyway this time, the score curves look less spiky than in the ML case. This is due to the regularization effect of the Bayesian approach that does not suffer from the reduced amount of data per Gaussian when the number of speakers is large.

On the other hand in the VB system the free energy *always* select the best system (see table 7.3). This is an extremely appealing result because it shows that the Variational Bayesian method, even if approximated provides an efficient model selection criterion.

Figures (7.16-7.19) show on the same graph for the four files the clustering score $K$ and the variational free energy $F_m$. It can be easily noticed that $F_m$ has the same form as $K$ and the maxima of $F_m$ and $K$ coincide.

To summarize the free energy closely follows the clustering score while the BIC requires some tuning to be effective in the model selection task. In next sections we will study the impact of prior distributions on the final score. We will consider both the cases of non-informative (small) priors and strong priors and finally consider the empirical optimal priors.

| File | File 1 | | | | File 2 | | | |
|---|---|---|---|---|---|---|---|---|
| | $N_c$ | acp | asp | K | $N_c$ | acp | asp | K |
| MAP (known) | 8 | 0.69 | 0.89 | 0.78 | 14 | 0.70 | 0.58 | 0.64 |
| MAP (best) | 10 | 0.84 | 0.90 | 0.87 | 9 | 0.70 | 0.78 | 0.74 |
| MAP/BIC $\lambda = 1$ | 35 | 0.87 | 0.29 | 0.52 | 32 | 0.89 | 0.41 | 0.60 |
| MAP/BIC $\lambda = 2$ | 16 | 0.85 | 0.74 | 0.79 | 22 | 0.88 | 0.54 | 0.69 |
| MAP/BIC $\lambda = 3$ | 10 | 0.69 | 0.71 | 0.87 | 9 | 0.70 | 0.78 | 0.74 |
| File | File 3 | | | | File 4 | | | |
| | $N_c$ | acp | asp | K | $N_c$ | acp | asp | K |
| MAP (known) | 16 | 0.77 | 0.78 | 0.77 | 21 | 0.75 | 0.64 | 0.69 |
| MAP (best) | 15 | 0.76 | 0.83 | 0.80 | 12 | 0.70 | 0.80 | 0.74 |
| MAP/BIC $\lambda = 1$ | 33 | 0.89 | 0.54 | 0.69 | 31 | 0.81 | 0.47 | 0.62 |
| MAP/BIC $\lambda = 2$ | 15 | 0.76 | 0.83 | 0.80 | 21 | 0.75 | 0.64 | 0.69 |
| MAP/BIC $\lambda = 3$ | 15 | 0.76 | 0.83 | 0.80 | 12 | 0.70 | 0.80 | 0.74 |

Table 7.2: Results on NIST 1996 HUB-4 evaluation test for speaker clustering,MAP/BIC

| File | File 1 | | | | File 2 | | | |
|---|---|---|---|---|---|---|---|---|
| | $N_c$ | acp | asp | K | $N_c$ | acp | asp | K |
| VB (known) | 8 | 0.72 | 0.92 | 0.81 | 14 | 0.69 | 0.64 | 0.67 |
| VB (best) | 10 | 0.84 | 0.90 | 0.87 | 9 | 0.70 | 0.78 | 0.74 |
| VB (selected) | 10 | 0.84 | 0.90 | 0.87 | 9 | 0.70 | 0.78 | 0.74 |
| File | File 3 | | | | File 4 | | | |
| | $N_c$ | acp | asp | K | $N_c$ | acp | asp | K |
| VB (known) | 16 | 0.76 | 0.84 | 0.80 | 21 | 0.71 | 0.65 | 0.68 |
| VB (best) | 14 | 0.76 | 0.84 | 0.80 | 12 | 0.69 | 0.79 | 0.74 |
| VB (selected) | 14 | 0.76 | 0.84 | 0.80 | 12 | 0.69 | 0.79 | 0.74 |

Table 7.3: Results on NIST 1996 HUB-4 evaluation test for speaker clustering, Variational Bayesian learning/model selection with non-informative priors

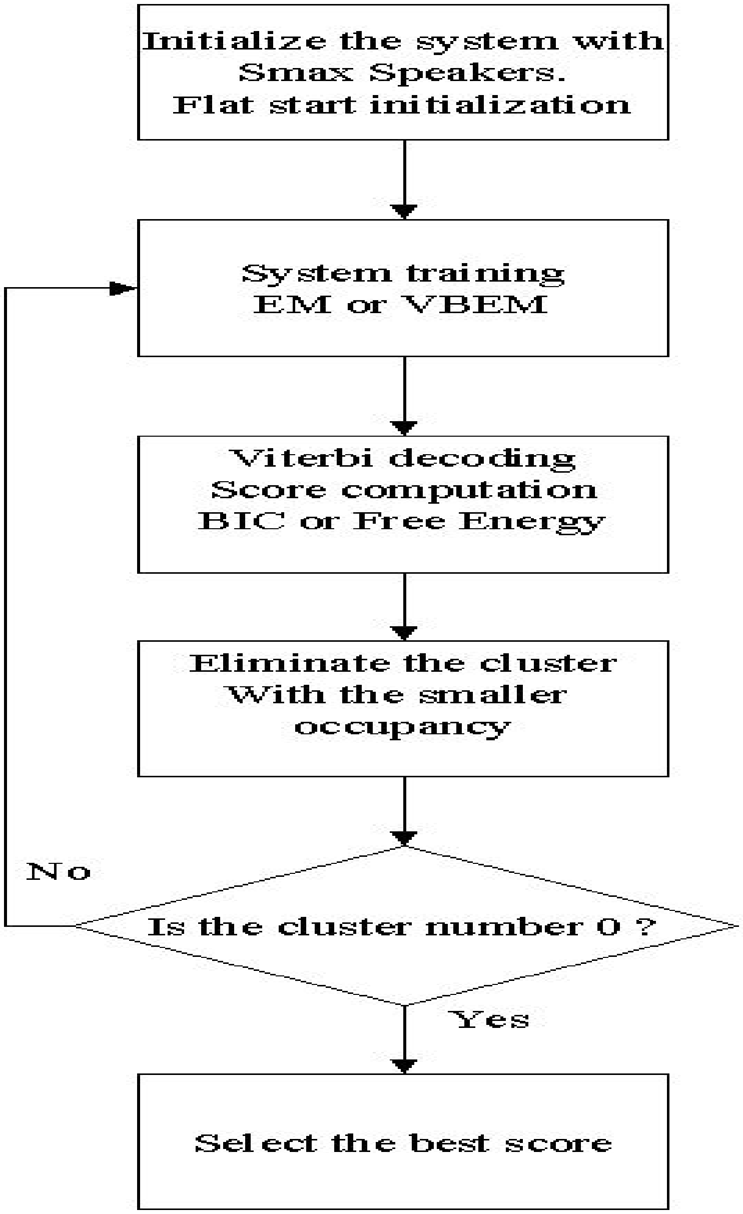


Figure 7.3: Flow chart of the clustering algorithm: the state number if progressively reduced from $S_{max}$ to 1 and all different systems are scored with the BIC or with the Free Energy; the best system is the system with the higher score.

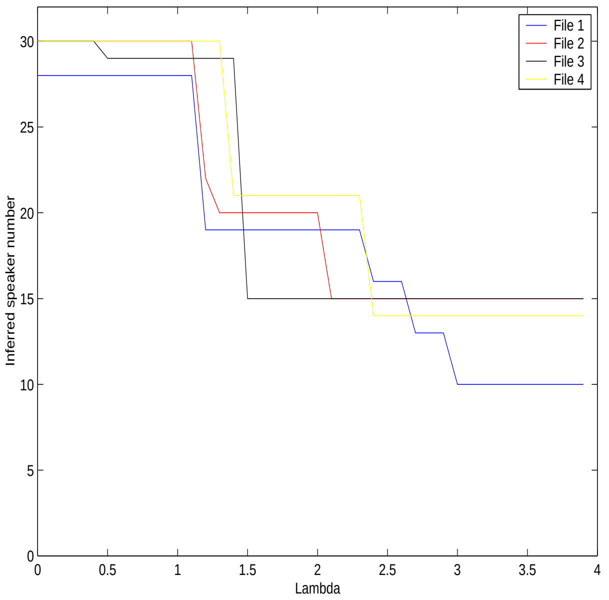


Figure 7.4: Speaker number inferred by BIC criterion w.r.t. $\lambda$ (System 1 ML/BIC)

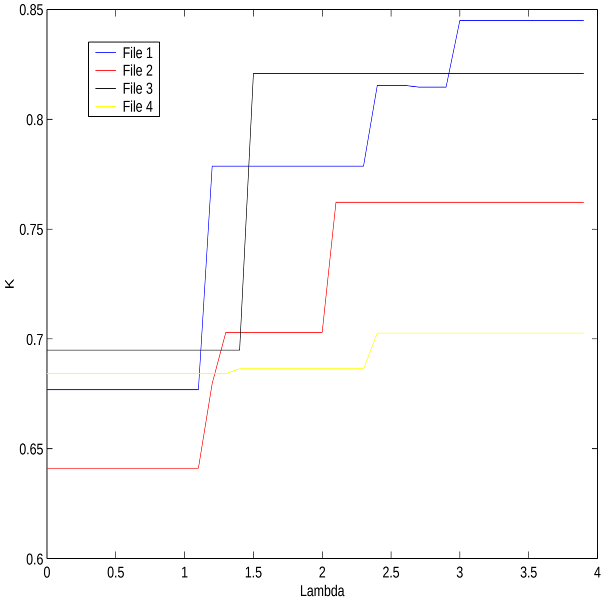


Figure 7.5: K values inferred by BIC criterion w.r.t $\lambda$ (System 1 ML/BIC)

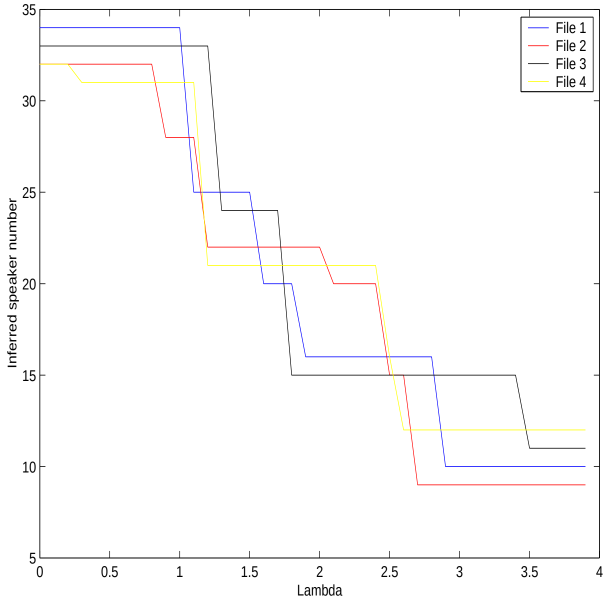


Figure 7.6: Speaker number inferred by BIC criterion w.r.t. $\lambda$ (System 2 MAP/BIC)

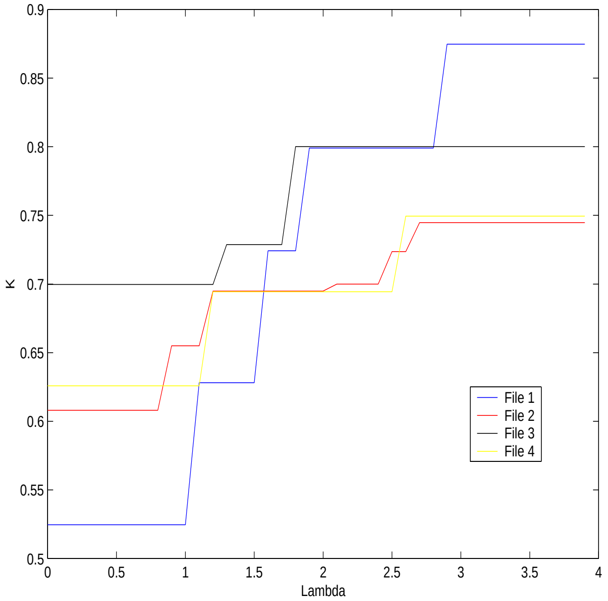


Figure 7.7: K values inferred by BIC criterion w.r.t $\lambda$ (System 2 MAP/BIC)

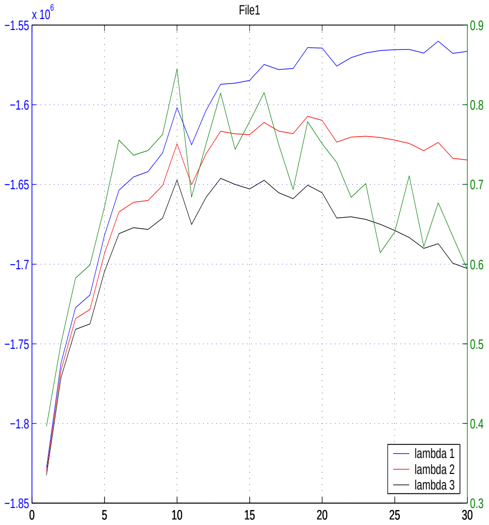


Figure 7.8: BIC for $\lambda = \{1, 2, 3\}$ and clustering score for file 1 - system 1 (ML/BIC)

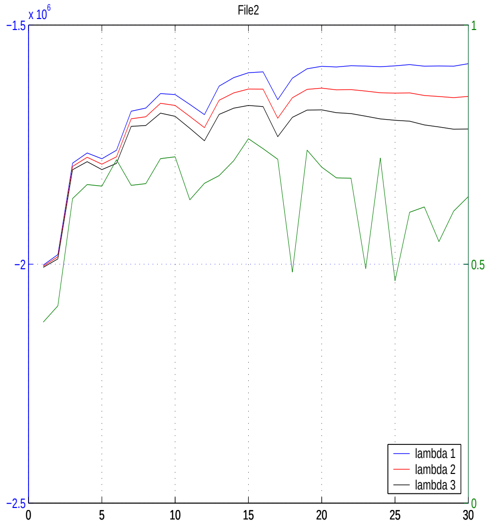


Figure 7.9: BIC for $\lambda = \{1, 2, 3\}$ and clustering score for file 2 - system 1 (ML/BIC)

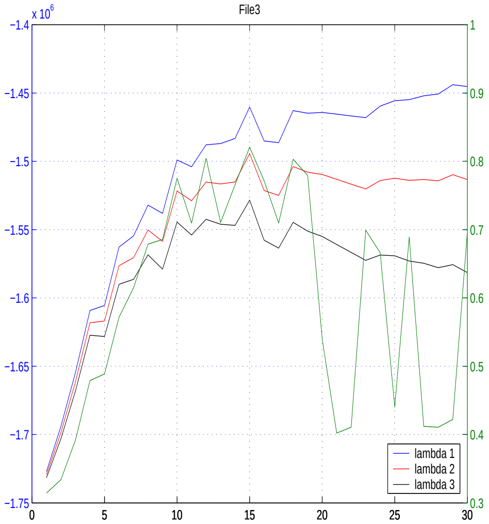


Figure 7.10: BIC for $\lambda = \{1, 2, 3\}$ and clustering score for file 3 - system 1 (ML/BIC)

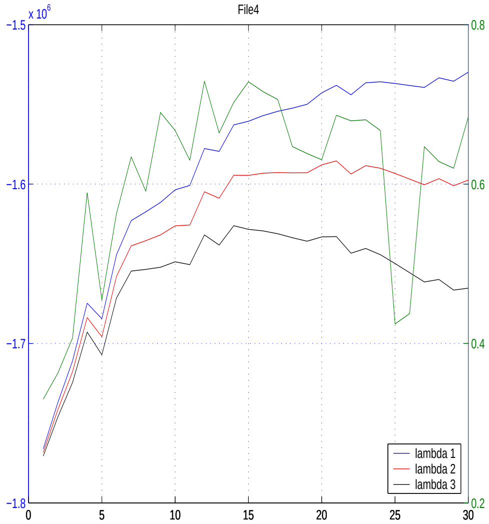


Figure 7.11: BIC for $\lambda = \{1, 2, 3\}$ and clustering score for file 4 - system 1 (ML/BIC)

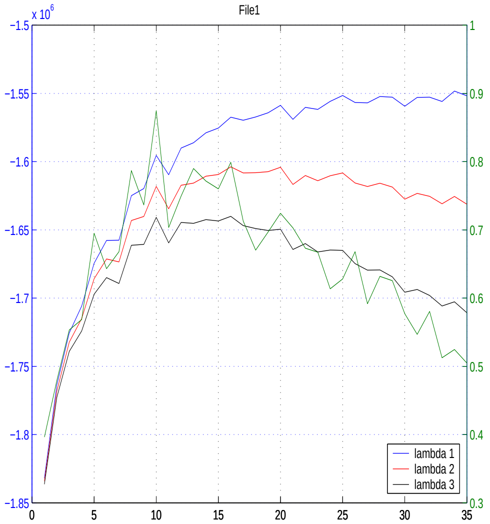


Figure 7.12: BIC for $\lambda = \{1, 2, 3\}$ and clustering score for file 1 - system 2 (MAP/BIC)

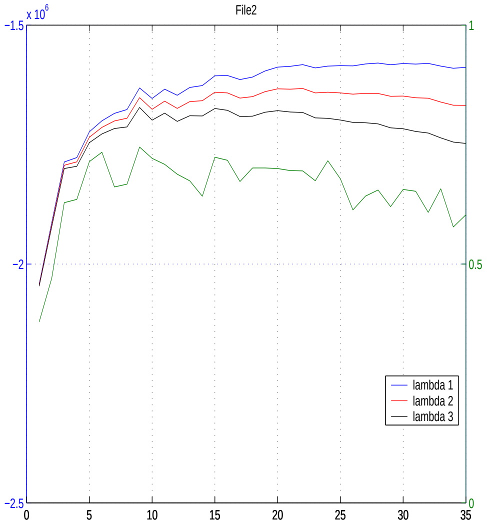


Figure 7.13: BIC for $\lambda = \{1, 2, 3\}$ and clustering score for file 2 - system 2 (MAP/BIC)

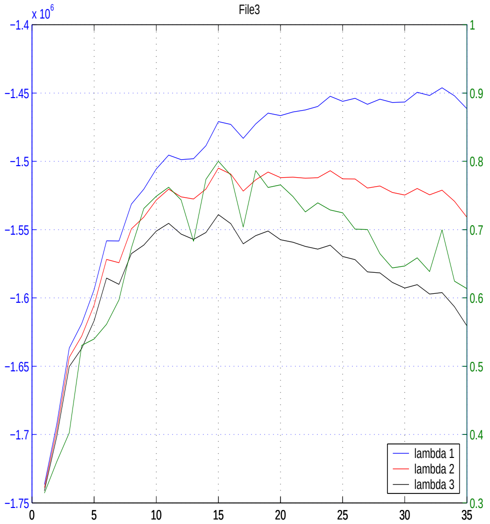


Figure 7.14: BIC for $\lambda = \{1, 2, 3\}$ and clustering score for file 3 - system 2 (MAP/BIC)

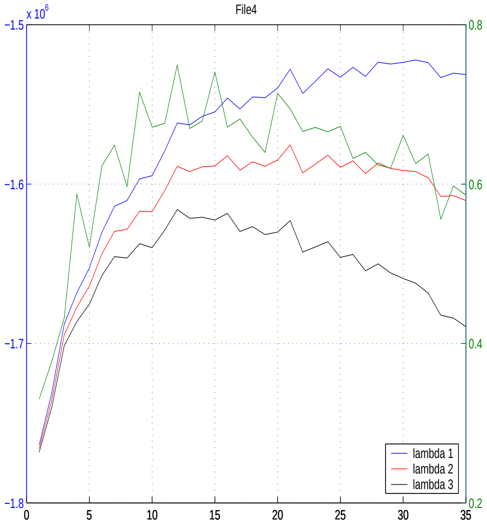
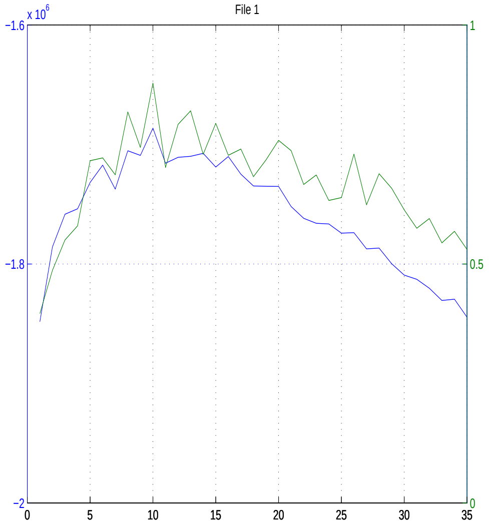


Figure 7.15: BIC for $\lambda = \{1, 2, 3\}$ and clustering score for file 4 - system 2 (MAP/BIC)



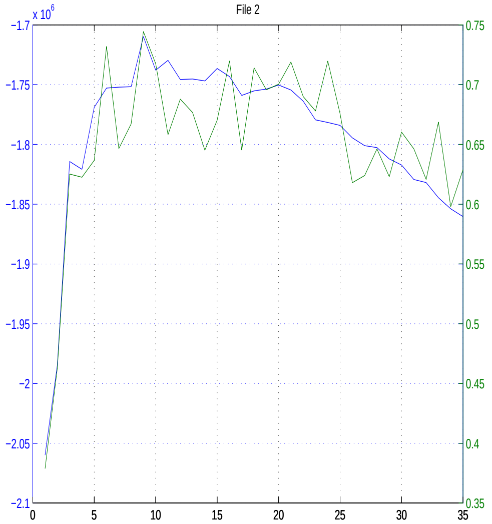
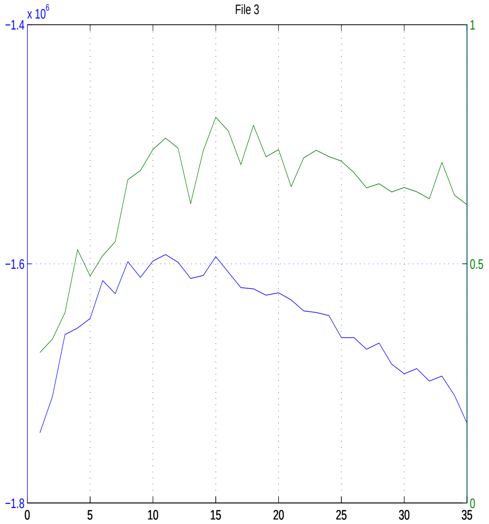


Figure 7.16: Free energy and clustering score for file 1

Figure 7.17: Free energy and clustering score for file 2



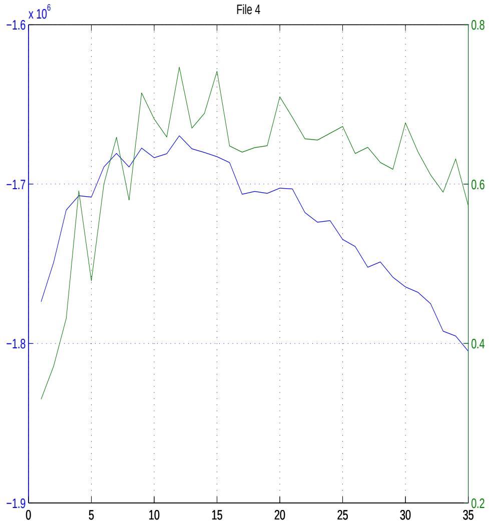


Figure 7.18: Free energy and clustering score for file 3

Figure 7.19: Free energy and clustering score for file 4

## 7.10 Dependence on non-informative priors

In this section we study the impact of the prior strength (i.e. the factor $\tau$) on the clustering result. We limit our investigations now to the case of broad non-informative priors obtained setting a small value of $\tau$.

Let us consider system 2 based on MAP/BIC. As long as the prior distribution is "weak" the clustering result and the final parameter estimation will be the same because priors do not add any kind of information. Concerning the model selection task, BIC just consider parameter estimation for the model selection: if parameters are always the same, the selected model will be the same.

The case of Variational Bayesian learning (system 3) is different. In fact small non-informative priors will produce as before the same posterior distributions over parameters (and so the same segmentation) but a different free energy. In fact the penalty term in the variational free energy consists in the KL divergence between posterior and prior distributions that will change according to the choice of the parameter $\tau$.

In the following experiments, we progressively moved the value of $\tau$ from $10^{-9}$ to 1. In figures (7.20-7.23) the best score (blue line) and the selected score (red line) are plotted on a semilogarithmic scale. When the prior is small VB free energy selects the best score while when the priors become informative w.r.t. data the selected system holds a smaller score.

It can be noticed that when priors are non informative the system is very robust in term of model selection capacity if the amount of data is large. The same conclusion was found in [137] in a large vocabulary speech recognition system.

## 7.11 Dependence on strong priors

The choice of flat priors is obviously not the best choice in terms of clustering results. In fact broad priors represent the case in which there is no prior knowledge on the data. It is in any case interesting to consider the influence of strong informative prior distributions on the speaker clustering result.

As in previous section we moved the hyperparameter $\tau$ considering this time value that cannot be neglected during the training phase. We increased the value of $\tau$ from 1 to 1000 by step of 100. Figures (7.24-7.27) shows the best score (in blue) and the selected score (in red) using the VB learning/model selection function of the prior $\tau$ for the four BN files. The black dashed line represents the baseline obtained using non-informative priors. In this case the best score coincides with the selected score.

At first we can notice in all the four graphs a peak in the best system performances for a value of $\tau$ in between 200 and 300. This means that the maximum of the scoring function is obtained for a value of strong value of $\tau$. Increasing the strength of the prior distribution decreases the value of the corresponding score.

On the other hand the red line shows the selected system and it can be noticed that the stronger the prior is, the weaker the correspondence between best score and free energy is. In other words a very strong prior distribution deteriorates both the model learning and the model selection.

Anyway in some interval of prior distributions the system performs better than the system with non-informative priors. In the empirical Bayesian approach the prior distribution parameters (i.e. hyperparameters) are optimized on the basis of data. We will consider these cases in section 7.13.

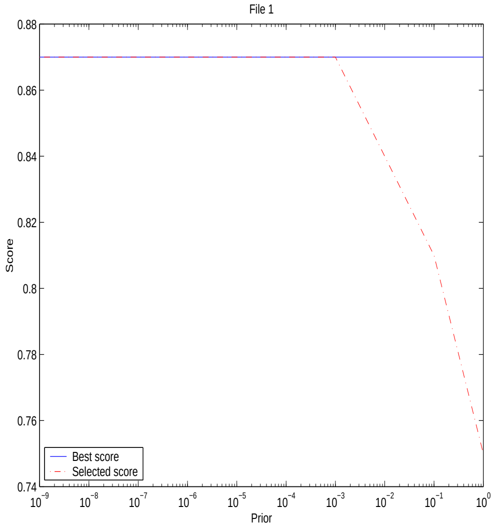


Figure 7.20: Dependence on weak prior distributions for file 1

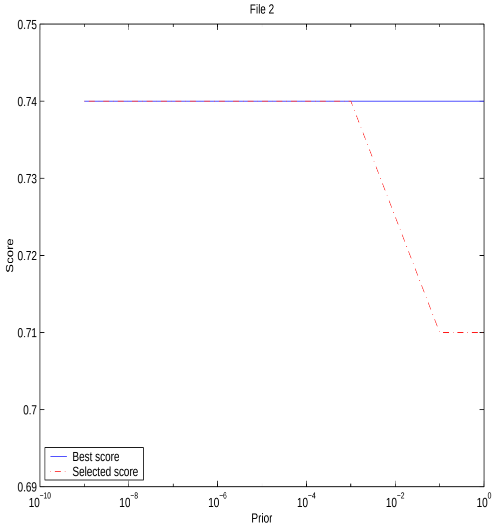


Figure 7.21: Dependence on weak prior distributions for file 2

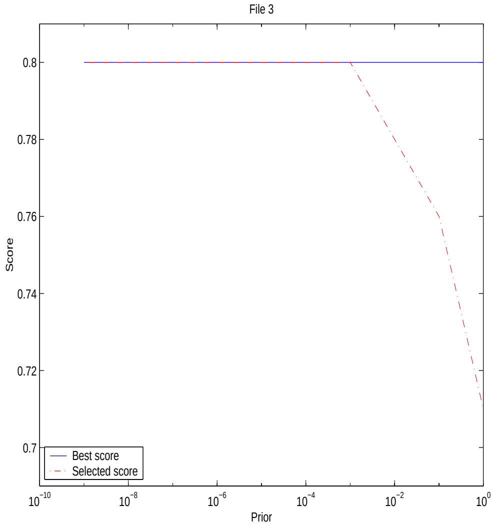


Figure 7.22: Dependence on weak prior distributions for file 2 for file 3

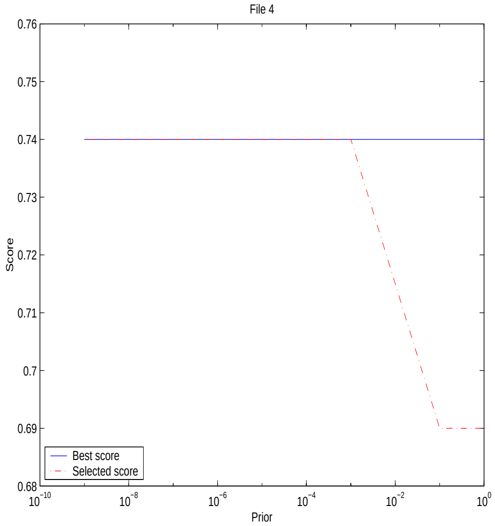


Figure 7.23: Dependence on weak prior distributions for file 4

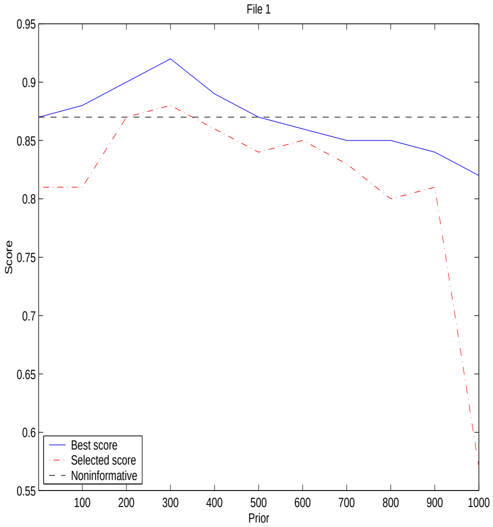


Figure 7.24: Dependence on strong prior distributions for file 1

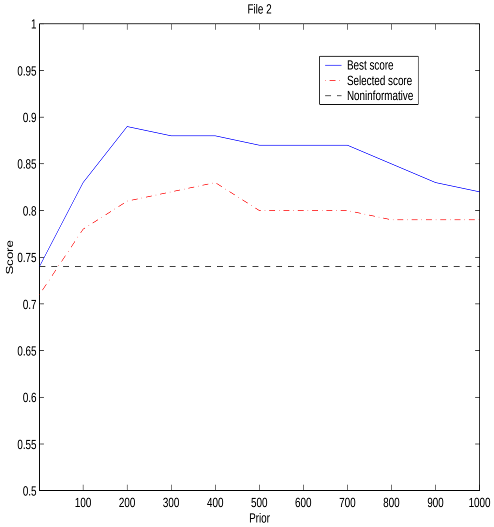


Figure 7.25: Dependence on strong prior distributions for file 2

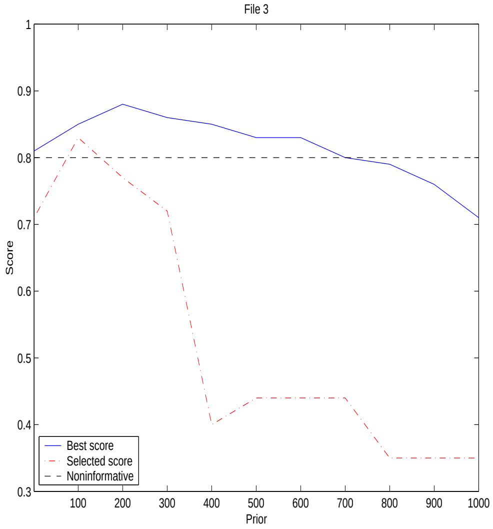


Figure 7.26: Dependence on strong prior distributions for file 2 for file 3

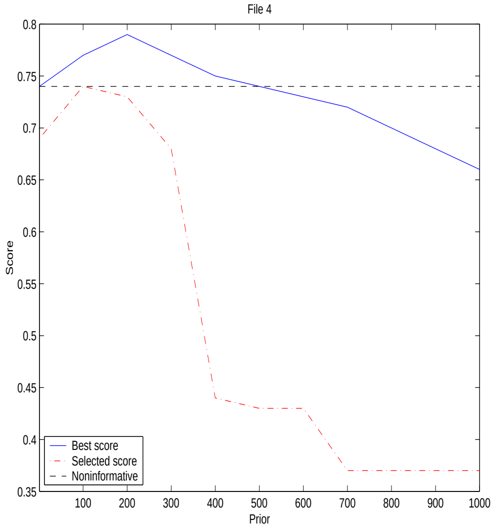


Figure 7.27: Dependence on strong prior distributions for file 4

### 7.11.1 Comparison with MAP

To make the comparison as complete as possible in this section we study the difference between VB and MAP with strong prior distributions. Both MAP and VB systems are initialized with the same strong prior distributions. Value of $\tau$ moves from 1 to 1000 by step of 100. Table 7.4 contains best scores for MAP and VB; NI designates the score for non-informative prior.

Scores for the two systems are very close for all values of $\tau$. VB performs slightly better for some prior values probably because of its capacity to average over all possible models in each case. Both approaches have a peak in the best system score around a value of $\tau$ equal to 300 and performances deteriorate when prior distribution becomes too strong. Figures (7.28-7.31) plot the MAP and the VB best scores w.r.t. $\tau$. Even if the VB score is higher than the MAP score, they are very close and have the same behavior w.r.t $\tau$ i.e. a peak around $\tau = 200$ and then performances degrade.

On the other side selecting the best system with the BIC presents some serious problems. In table 7.5 the best score obtained tuning $\lambda$ value is presented for different values of $\tau$. In brackets the notation y/n (yes/no) designates if the BIC score corresponds to the best absolute score. For values of $\tau$ close to 100 or 200 the BIC can select the best absolute score; when $\tau$ increases BIC is extremely inefficient. Different prior distributions lead to different values of $\lambda$, surprisingly when $\tau$ increases the best score is obtained with $\lambda = 0$. Anyway it is evident that optimal $\lambda$ is dependent on prior distributions.

An interpretation to the value of $\lambda = 0$ when $\tau$ is large can be provided. In fact when prior distributions are extremely peaked all the "mass" of the integral is concentrated in a point, the MAP estimation. It means that the likelihood computed on the MAP estimation concentrates all the model selection properties of the Bayesian integral. For this reason the best system is the one with no penalty at all i.e. $\lambda = 0$.

In summary both MAP and VB methods are extremely sensitive to initial prior distributions. Both methods have a peak in the best score for a given value in the prior distribution and performance degrades when prior becomes too strong. Anyway VB slightly outperforms the MAP for informative priors. A big drawback with the MAP/BIC system is the value of the optimal threshold $\lambda$ that changes with the prior distribution. For very strong prior distribution the BIC is completely unable to select the best result and the optimal threshold collapse to the solution $\lambda = 0$.

| File | NI | 1 | 100 | 200 | 300 | 400 |
|---|---|---|---|---|---|---|
| File1 MAP | 0.87 | 0.88 | 0.88 | 0.90 | 0.90 | 0.87 |
| File 1 VB | 0.87 | 0.87 | 0.88 | 0.90 | 0.92 | 0.89 |
| File2 MAP | 0.74 | 0.74 | 0.83 | 0.88 | 0.88 | 0.87 |
| File2 VB | 0.74 | 0.74 | 0.83 | 0.89 | 0.88 | 0.88 |
| File3 MAP | 0.80 | 0.80 | 0.85 | 0.87 | 0.86 | 0.84 |
| File3 VB | 0.80 | 0.81 | 0.85 | 0.88 | 0.86 | 0.85 |
| File 4 MAP | 0.74 | 0.74 | 0.76 | 0.76 | 0.73 | 0.73 |
| File 4 VB | 0.74 | 0.74 | 0.77 | 0.79 | 0.77 | 0.75 |

| File | 500 | 600 | 700 | 800 | 900 | 1000 |
|---|---|---|---|---|---|---|
| File1 MAP | 0.85 | 0.86 | 0.84 | 0.83 | 0.82 | 0.80 |
| File 1 VB | 0.87 | 0.86 | 0.85 | 0.85 | 0.84 | 0.82 |
| File2 MAP | 0.87 | 0.86 | 0.86 | 0.82 | 0.82 | 0.81 |
| File2 VB | 0.87 | 0.87 | 0.87 | 0.85 | 0.83 | 0.82 |
| File 3 MAP | 0.81 | 0.79 | 0.77 | 0.73 | 0.71 | 0.71 |
| File 3 VB | 0.83 | 0.83 | 0.80 | 0.79 | 0.76 | 0.71 |
| File 4 MAP | 0.70 | 0.70 | 0.70 | 0.67 | 0.64 | 0.64 |
| File 4 VB | 0.74 | 0.73 | 0.72 | 0.70 | 0.68 | 0.66 |

Table 7.4: Best score K function of prior $\tau$ for MAP/BIC and VB systems (NI = non-informative).

| File | NI | 1 | 100 | 200 | 300 | 400 |
|---|---|---|---|---|---|---|
| File 1 | 2.9(y) | 2.8(y) | 2.4(y) | 0.6(y) | 0.4(n) | 0(n) |
| File 2 | 2.7(y) | 2.7(y) | 0.4(y) | 0.5(n) | 0(n) | 0(n) |
| File 3 | 1.9(y) | 1.9(y) | 0.6(n) | 0.7(n) | 0(n) | 0(n) |
| File 4 | 2.6(y) | 2.6(y) | 1(y) | 0.9(n) | 0.9(n) | 0.2(n) |

| File | 500 | 600 | 700 | 800 | 900 | 1000 |
|---|---|---|---|---|---|---|
| File 1 | 0(n) | 0(n) | 0(n) | 0(n) | 0(n) | 0(n) |
| File 2 | 0(n) | 0(n) | 0(n) | 0(n) | 0(n) | 0(n) |
| File 3 | 0(n) | 0(n) | 0(n) | 0(n) | 0(n) | 0(n) |
| File 4 | 0(n) | 0(n) | 0(n) | 0(n) | 0(n) | 0(n) |

Table 7.5: Best value of BIC threeshold $\lambda$ function of prior distribution for the MAP/BIC system. In brackets (yes/no) if the BIC chooses the best result or not

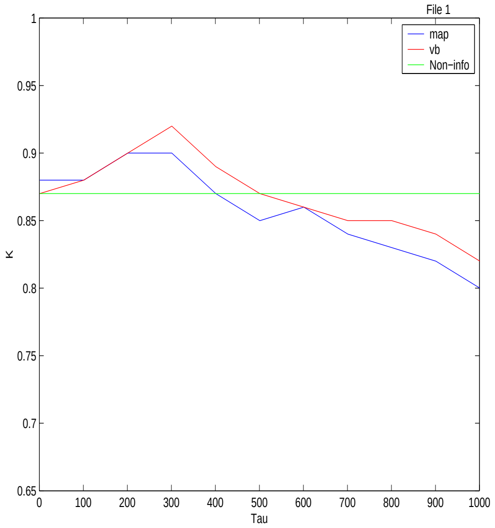


Figure 7.28: Comparison between MAP and VB learning function of $\tau$ for file 1

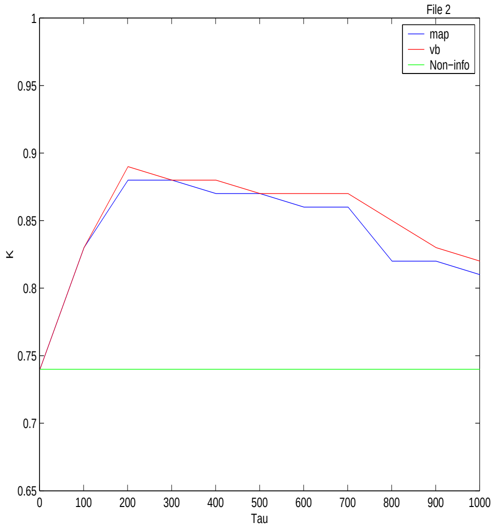


Figure 7.29: Comparison between MAP and VB learning function of $\tau$ for file 2

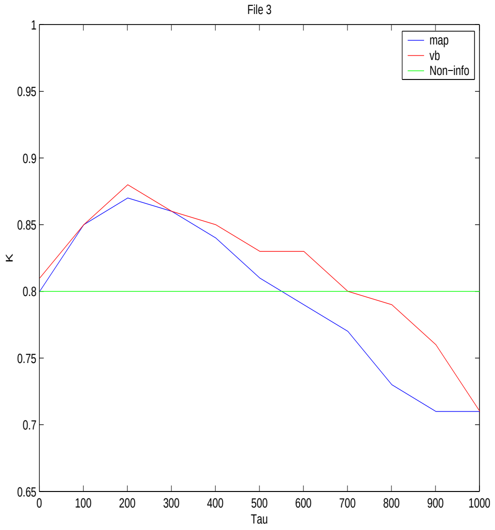


Figure 7.30: Comparison between MAP and VB learning function of $\tau$ for file 3

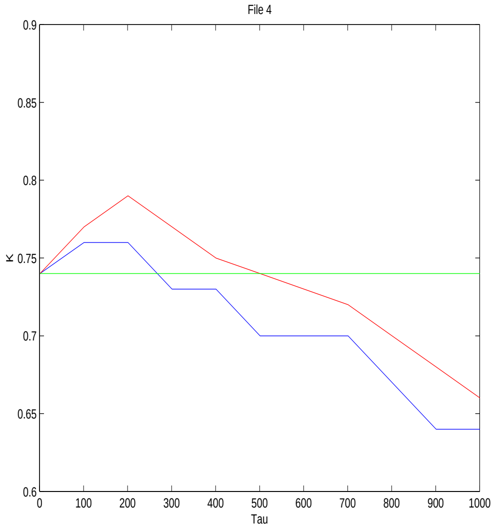


Figure 7.31: Comparison between MAP and VB learning function of $\tau$ for file 4

## 7.12 Automatic component death

As discussed in chapter 5 Variational Bayesian methods have the appealing property of self-pruning extra degree of freedom during the learning task. This interesting result turns out extremely useful when the model is not chosen accurately with respect to the data (e.g. many Gaussian components for limited amount of data).

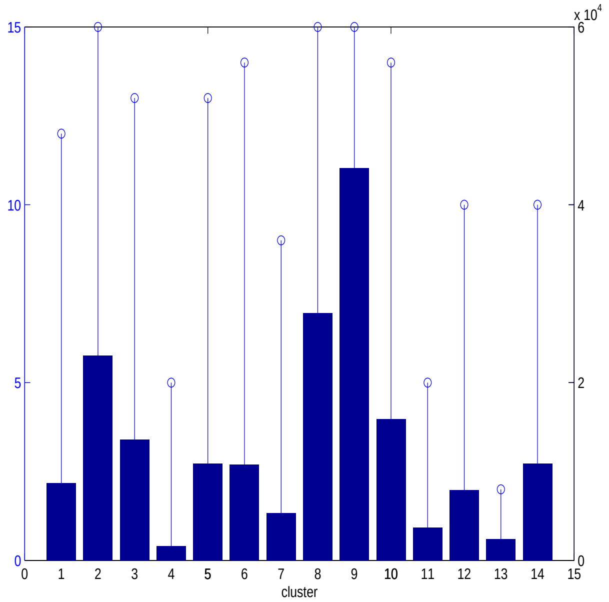


Figure 7.32: Number of final Gaussian components and amount of data per cluster (file 2)

In speaker indexing problems, the amount of data for each speaker is generally asymmetrical in the sense that some speakers provide very long segments while others pronounce just few utterances. It is logical to model the first one with a more complex model compared to the second.

Previous work in this sense can be found in [105]. The speaker model consists in a GMM when large amount of data is provided and in a Vector Quantization (VQ) model when poor amount of data is available. To make uniform the distance measure between the GMM and the VQ, the Vector Quantization model is realized through a CVGMM (Common Variance GMM) that generalizes the VQ. The model selection is anyway realized with the BIC and the additional cost of the model switching between the GMM and the VQGMM must be handled.

The VB solution to the same issue is simpler and more elegant. It is in fact enough to initialize the GMM with a large initial number of components and train the system using the VBEM algorithm. At the end of the training session components that are redundant in representing the data have a posterior probability equal to the prior probability resulting in a null contribution to the free energy. As long as the prior and the posterior are the same they will result in a null KL divergence too, the only term that will survive in the penalty is the one in the KL divergence of the Dirichlet distribution over component weights.

Figure 7.32 plots on the same graph the final Gaussian components (out of the initial 15 components per model) and the corresponding amount of data per cluster in the case of file 2. It is evident to notice that the initial number of components is preserved when the amount of data in the cluster is consistent while in the case of poor data the final number of components may dramatically reduce from 15 to just a few. In this way overfitting problem can be avoided and a more compact representation of the considered speaker can be obtained.

Anyway an important consideration must be done: speaker clustering is a simple modeling task and no inference on unseen data is required. It has been noticed in [96], that VB self-pruning may provide wrong results in some extreme situations (in [96] the case of a single learning point is treated). In some cases the pruning may eliminate degrees of uncertainty that could be useful in terms of generality of the model and make the inference on unseen data more efficient. Anyway in the speaker indexing task no inference is required and the result is simply the file segmentation. In conclusion the self-pruning effect is definitely an appealing property for this kind of task.

## 7.13 Optimal prior

In this section we compare best system results for three different hyperparameter settings: non-informative hyperparameters, heuristically optimal hyperparameters and the best value obtained manually tuning the prior distributions. Results are summarized in table 7.6. The procedure for inferring the optimal prior distributions is described in section 7.3.3. It basically consists in minimizing the KL divergence between the variational posterior and the prior. This approach could be defined as a "maximum likelihood" approach to hyperparameters and suffers of the same drawback of common ML approach. In fact point estimation of hyperparameters that define prior distributions is not faithful to the Bayesian framework. On the other side using non-informative prior is not the best solution in terms of performance (see 7.11).

Looking at performances of table 7.3.3, we can notice that using heuristic priors performances of system increases w.r.t. the non-informative prior for 3 of the 4 files while for file 2 the result is unchanged. In file 1 an extremely small improvement is noted (from 87% to 88%) while in file 3 and file 4 improvements are more consistent (respectively 7% and 5%).

Let us compare now the heuristic prior results with the tuned prior results. In the first two files heuristic prior results are far from best results obtained tuning the system. In file 3 they are close (87% and 88%). In file 4 they result in the same score (79%) but from two different solutions (i.e. different *acp* and *asp*). In summary the solution provided with heuristic prior is not that close to the solution obtained by manually tuning prior distributions. A possible explication is probably due to the fact that considering hyperparameters from a maximum likelihood point of view is not the optimal choice in this kind of problem.

Another interesting remark consists in the fact that the inferred number cluster for optimal prior system is higher than the non-informative prior system. In the same way the manually tuned system offers the highest number of inferred cluster. It basically means that with the appropriate prior distribution more potentiality of the model (i.e. more state) are used compared to the non-informative case.

To summarize the heuristic prior system provides higher performances compared to the non-informative case at the price of a higher number of clusters. On the other side optimal prior results in weaker performances compared to the manually tuned priors. We can conclude that heuristic prior is not close to the best manually tuned solution to the problem.

| File | File 1 | | | | File 2 | | | |
|---|---|---|---|---|---|---|---|---|
| | $N_c$ | acp | asp | K | $N_c$ | acp | asp | K |
| Non-info | 10 | 0.84 | 0.90 | 0.87 | 9 | 0.70 | 0.78 | 0.74 |
| Opt | 17 | 0.88 | 0.89 | 0.88 | 9 | 0.70 | 0.78 | 0.74 |
| Tuned | 19 | 0.90 | 0.93 | 0.92 | 17 | 0.87 | 0.91 | 0.89 |

| File | File 3 | | | | File 4 | | | |
|---|---|---|---|---|---|---|---|---|
| | $N_c$ | acp | asp | K | $N_c$ | acp | asp | K |
| Non-info | 14 | 0.76 | 0.84 | 0.80 | 12 | 0.69 | 0.79 | 0.74 |
| Opt | 18 | 0.82 | 0.91 | 0.87 | 22 | 0.81 | 0.77 | 0.79 |
| Tuned | 22 | 0.80 | 0.89 | 0.88 | 22 | 0.74 | 0.84 | 0.79 |

Table 7.6: Non informative, optimal and tuned value

## 7.14 Empirical priors i.e. VB/MAP adaptation

An advantage of Bayesian methods is the possibility of using previous knowledge about the problem in the prior term. In this section we consider the previous speaker clustering problem in which prior distributions are initialized according to an Universal Background Model (UBM) build from a separate set of data.

This is the same principle of MAP adaptation techniques ([113]). This time we will compare the MAP adaptation with the VB adaptation (as already described in section 5). Again while the MAP based system uses the BIC for model selection, the VB approach uses free energy instead.

The UBM consists in a 32 Gaussian component GMM trained with 2 hours of speech coming from regions FO, F1 and F2 of the training data set BN-96. In order to estimate prior probabilities from the UBM parameters we used the same choice of [47] i.e. fixed a set of relevance factor $\tau_i$ and an UBM with parameters $B = \sum_{i=1}^{M} c_i^b N(\mu_i^b, \Sigma_i^b)$ (as we already discussed in section 5.6), hyperparameters are set as:

$$\lambda_{0i} = c_i^b \sum_i \tau_i \tag{7.94}$$

$$\rho_{0i} = \mu_i^b \tag{7.95}$$

$$\xi_{0i} = \tau_i^b \tag{7.96}$$

$$a_{0i} = \tau_i \tag{7.97}$$

$$B_{0i} = \tau_i {\Sigma_i^b}^{-1} \tag{7.98}$$

We set $\tau_i = \tau \quad \forall i$. Parameter $\tau$ determines the strength of the a priori distribution: small values of $\tau$ result in a non-informative prior while increasing values of $\tau$ increase the relevance of the prior model. In order to study clustering results, we have considered different values of $\tau$ and reported speaker clustering results in table 7.7. The range of values for $\tau$ is different from the one considered in the non-empirical case because systems have a different behavior. Also in this case the VB system outperforms the MAP/BIC. The difference becomes larger when the parameter $\tau$ increases; this is due to the averaging effect of the variational method. Figures (7.33)-(7.36) depict the dependence of MAP and VB system w.r.t. parameter $\tau$. On the other side the VB system with empirical prior suffers from the same model selection problem of the non empirical system. In fact when the value of $\tau$ becomes too large, the free energy selects a model that does not correspond to the best score. Figures (7.37)-(7.40) show for the four files the best score and the selected score function of $\tau$. When the value of $\tau$ is low the selected model is close to the best value; the difference becomes larger when the value of $\tau$ is increased. From figures (7.37)-(7.40) we can notice a peak in performances around a value of $\tau = 900$. In order to have an idea of best clustering values that can be obtained by adaptation we report in table 7.8 results VB systems and in table 7.9 results for MAP/BIC and for a value of $\tau = 900$. BIC was heuristically tuned with a threshold $\lambda$ in order to achieve the best performance. Elements in table 7.8 have the same meaning as in previous sections.

| File | 100 | 1000 | 2000 | 3000 | 4000 | 5000 |
|---|---|---|---|---|---|---|
| File1 MAP | 0.81 | 0.87 | 0.86 | 0.84 | 0.83 | 0.82 |
| File 1 VB | 0.85 | 0.87 | 0.87 | 0.86 | 0.86 | 0.85 |
| File2 MAP | 0.80 | 0.89 | 0.86 | 0.84 | 0.83 | 0.82 |
| File2 VB | 0.80 | 0.90 | 0.88 | 0.86 | 0.86 | 0.84 |
| File3 MAP | 0.82 | 0.83 | 0.80 | 0.75 | 0.72 | 0.68 |
| File3 VB | 0.83 | 0.85 | 0.84 | 0.82 | 0.80 | 0.78 |
| File 4 MAP | 0.73 | 0.72 | 0.68 | 0.68 | 0.64 | 0.63 |
| File 4 VB | 0.72 | 0.75 | 0.73 | 0.72 | 0.72 | 0.71 |

Table 7.7: Best score K function of prior $\tau$ for MAP/BIC and VB systems

| File | File 1 | | | | File 2 | | | |
|---|---|---|---|---|---|---|---|---|
| | $N_c$ | acp | asp | K | $N_c$ | acp | asp | K |
| VB (known) | 8 | 0.68 | 0.86 | 0.77 | 14 | 0.70 | 0.82 | 0.76 |
| VB (best) | 24 | 0.88 | 0.85 | 0.87 | 18 | 0.90 | 0.90 | 0.90 |
| VB (selected) | 17 | 0.84 | 0.88 | 0.86 | 17 | 0.84 | 0.90 | 0.87 |
| File | File 3 | | | | File 4 | | | |
| | $N_c$ | acp | asp | K | $N_c$ | acp | asp | K |
| VB (known) | 16 | 0.79 | 0.87 | 0.83 | 21 | 0.70 | 0.75 | 0.73 |
| VB (best) | 21 | 0.85 | 0.86 | 0.85 | 19 | 0.70 | 0.80 | 0.75 |
| VB (selected) | 22 | 0.83 | 0.85 | 0.84 | 23 | 0.73 | 0.76 | 0.74 |

Table 7.8: Results on NIST 1996 HUB-4 evaluation test for speaker clustering, Variational Bayesian learning/model selection with empirical priors and $\tau = 900$

| File | File 1 | | | | File 2 | | | |
|---|---|---|---|---|---|---|---|---|
| | $N_c$ | acp | asp | K | $N_c$ | acp | asp | K |
| MAP/BIC (known) | 8 | 0.69 | 0.94 | 0.80 | 14 | 0.70 | 0.85 | 0.77 |
| MAP/BIC (best) | 25 | 0.89 | 0.83 | 0.86 | 20 | 0.87 | 0.90 | 0.89 |
| MAP/BIC (selected) | 25 | 0.89 | 0.83 | 0.86 | 21 | 0.89 | 0.85 | 0.87 |
| File | File 3 | | | | File 4 | | | |
| | $N_c$ | acp | asp | K | $N_c$ | acp | asp | K |
| MAP/BIC (known) | 16 | 0.76 | 0.85 | 0.80 | 21 | 0.71 | 0.68 | 0.69 |
| MAP/BIC (best) | 23 | 0.85 | 0.83 | 0.84 | 19 | 0.71 | 0.77 | 0.74 |
| MAP/BIC (selected) | 23 | 0.85 | 0.83 | 0.84 | 23 | 0.73 | 0.71 | 0.72 |

Table 7.9: Results on NIST 1996 HUB-4 evaluation test for speaker clustering, for MAP/BIC with empirical priors and $\tau = 900$

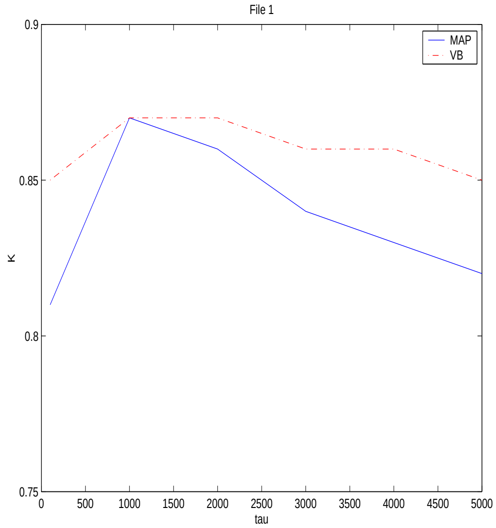


Figure 7.33: Dependence of MAP and VB with empirical prior function of $\tau$ for file 1

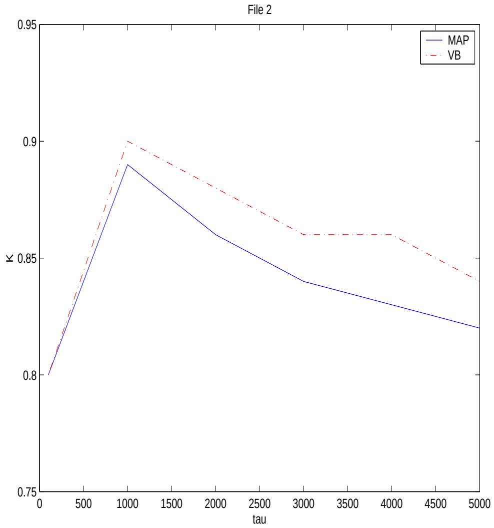


Figure 7.34: Dependence of MAP and VB with empirical prior function of $\tau$ for file 2

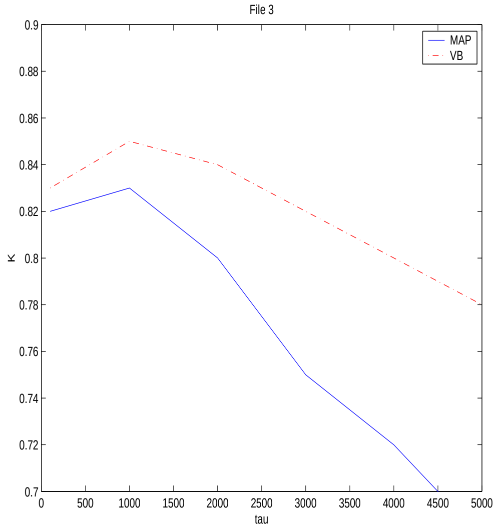


Figure 7.35: Dependence of MAP and VB with empirical prior function of $\tau$ for file 3

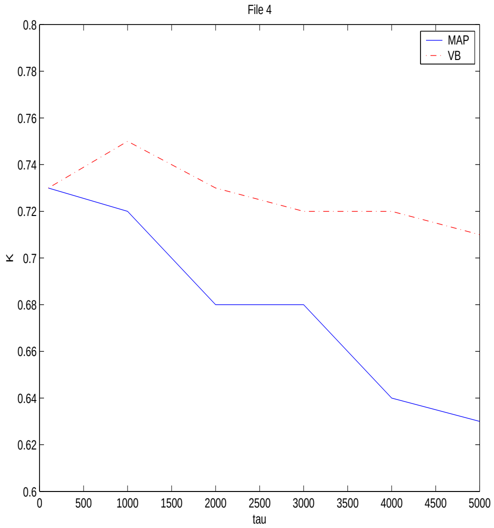
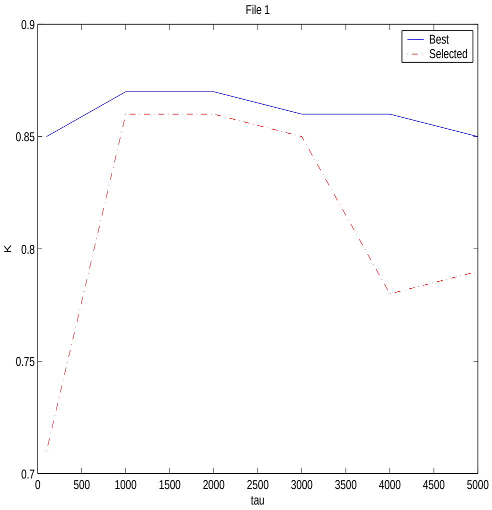


Figure 7.36: Dependence of MAP and VB with empirical prior function of $\tau$ for file 4



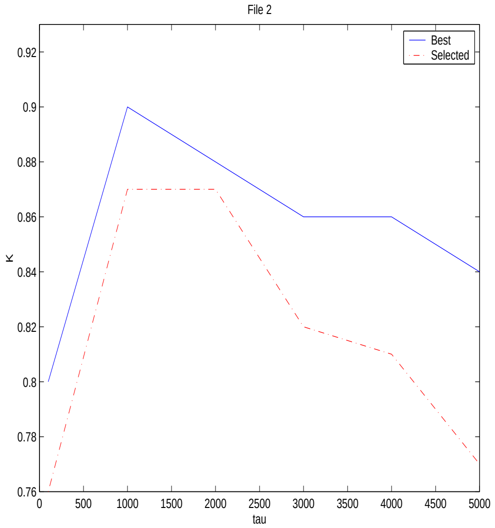
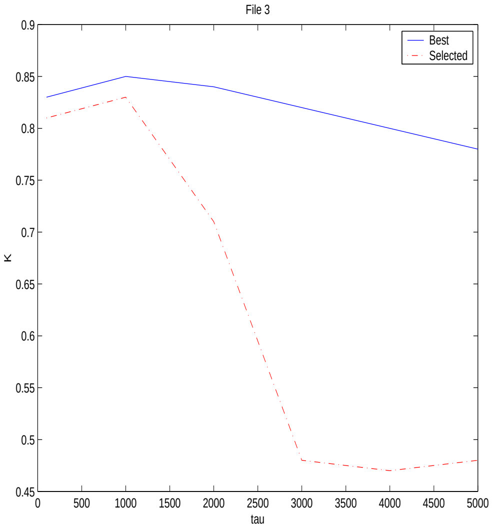


Figure 7.37: Best score and selected score for the VB system function of $\tau$ for file 1

Figure 7.38: Best score and selected score for the VB system function of $\tau$ for file 2



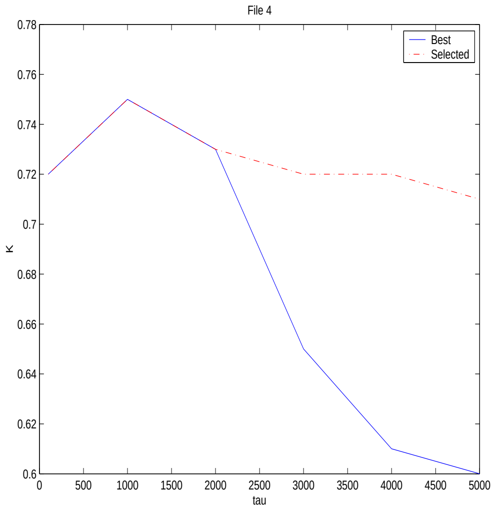


Figure 7.39: Best score and selected score for the VB system function of $\tau$ for file 3

Figure 7.40: Best score and selected score for the VB system function of $\tau$ for file 4

# Chapter 8

# Contributions and Conclusion

In this thesis we have discussed the use of Variational Bayesian methods in an audio indexing problem. Classical indexing problems have often been formulated as a model selection problem and solved with very rough approximations like the Bayesian information Criterion. The VB framework offers an elegant solution for both model learning and model selection. In our work we have proposed for the first time the use of this technique for speaker clustering and speaker change detection purposes. The main contribution of this thesis consists in the proposal and the study of a fully bayesian framework in an audio processing applications.

In the first part of this work we have discussed the theoretical framework of Variational Bayesian methods as developed in machine learning field in the last ten years. We developed as well the computation for popular model like HMM or GMM. A particularly emphasis has been put on the possibility of optimizing prior distributions i.e. finding optimal hyperparameters. In fact VB techniques as all Bayesian techniques are sensitive on prior distributions (depending on the current amount of available data). This dependence can be reduced by optimizing hyperparameters even if it looses the Bayesian spirit of the approach.

In experiments, we first have shown how VB is definitely more robust w.r.t. classical learning criteria like MAP or ML in a GMM learning/testing task. Experiments discussed in chapters 6 and 7 shows that improvements are obtained as well on real data. In the speaker changing point detection task, the VB allows a definitely better resolution than the BIC. Because of the reduced amount of data in this task, VB shows a high sensitivity to prior distributions.

On the other hand in speaker clustering based on both no a priori informations and on an UBM model for Broadcast News, segmentation shows a good robustness to non informative prior distributions. The problem of strong prior distributions and optimal prior distributions has also been addressed. In all cases the VB has outperformed both the ML/BIC and the MAP/BIC.

Overload brought by Variational Bayesian is extremely reduced because parameters expected values can be computed in closed form for the considered parameter distributions. Advantages on the other hand are interesting both in term of robustness (to overfitting, to poor data set) and in term of model selection (free energy is a model quality measure).

Interest in VB methods is anyway recent and lot of possible developments are still under investigation.

First of all an interesting progress may come from the possibility of making the technique as independent as possible from prior distributions. In this thesis we have used an empirical prior approach that has not shown a good robustness to different tasks. The investigation of hierarchical prior distributions (that have proven their efficiency in other tasks) may be a possible solution to prior distributions problems.

On the other hand VB has been applied to other challenging tasks like statistical signal processing and speech recognition. An important issue must be anyway pointed out. We have applied VB to problems that are inherently generative. Other tasks like speech recognition are discriminative tasks that have a completely different goal. For this reason the use of VB methods for doing model selection in ASR systems may be ineffective in the sense that a discriminative model is controlled by a generative criterion. Future research should focus on methods that are simultaneously discriminative and Bayesian in order to benefit of both model selection and discriminative training. A problem with current discriminative methods like MMI and MCE is that it is not straightforward to obtain a prior-posterior relation for parameter distributions; this mean that in classical HMM/GMM discriminative framework the bayesian solution is not directly applicable. As a final remark we would like to underline that those kinds of solutions can be applied in systems that use kernels (e.g. Support Vector Machine) in which Variational Bayesian approximations are easily applicable (see [121]). In those cases model selection in the bayesian sense is directly feasible because of the form of the discriminative function.

# Appendix A

# Some KL divergences

In this section we explicit the KL divergences used in this thesis; distributions considered here are Dirichlet distributions, Gaussian distributions and Wishart distributions.

### A.0.1 Dirichlet distribution

Given an hidden variable $\pi = \{\pi_1, \dots, \pi_k\}$, the Dirichlet distribution with hyperparameters $\alpha = \{\alpha_1, \dots \alpha_k\}$ designated with $\pi = Dir(\alpha)$ can be written as :

$$p(\pi|\alpha) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\dots\Gamma(\alpha_k)}\pi_1^{\alpha_1 - 1}\dots\pi_k^{\alpha_k - 1} \tag{A.1}$$

where

$$\alpha_0 = \sum_{j=1}^{k} \alpha_j \tag{A.2}$$

$$\alpha_j > 0 \tag{A.3}$$

$$\pi_j > 0 \tag{A.4}$$

$$\sum_{j=1}^{k} \pi_j = 1 \tag{A.5}$$

The KL divergence between two distributions with parameters $\alpha$ and $\beta$ can be explicited as:

$$KL(\alpha||\beta) = log\,\frac{\alpha_0}{\beta_0} - \sum_{j=1}^{k}[log\,\frac{\Gamma(\beta_j)}{\Gamma(\alpha_j)} - (\alpha_j - \beta_j)(\Psi(\beta_j) - \Psi(\alpha_j))] \tag{A.6}$$

In this work the first log moment is used as well:

$$\int p(\pi|\alpha)\, log\pi_j = \Psi(\alpha_j) - \Psi(\alpha_0) \tag{A.7}$$

where $\Psi(.)$ is the digamma function and $\Gamma(.)$ is the Gamma function (see [1]).

### A.0.2 Gaussian distribution

The Gaussian distribution or multivariate normal distribution with mean $\mu$ and covariance $\Sigma$ designated with $N(y|\mu,\Sigma)$ is defined as:

$$p(y|\mu,\Sigma) = (2\,\pi)^{\cdot/2}|\Sigma|^{-1/2}exp(-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)) \tag{A.8}$$

The KL divergence between $N(y|\mu_1,\Sigma_1)$ and $N(y|\mu_2,\Sigma_2)$ can be explicited as:

$$\begin{aligned} KL(\mu_1,\Sigma_1|\mu_2,\Sigma_2) = -\frac{1}{2}(log|\Sigma_1\Sigma_2| \\ +tr[I-[\Sigma_1+(\mu_1-\mu_2)(\mu_1-\mu_2)^T]\Sigma_2^{-1}]\,log\,e) \end{aligned} \tag{A.9}$$

### A.0.3 Wishart distribution

The Wishart distribution with $a$ degree of freedom and $B$ precision matrix is defined as:

$$p(W|a,B) = \frac{1}{Z_{aB}}|W|^{(a-d-1)/2}exp(-\frac{1}{2}tr[B^{-1}W]) \tag{A.10}$$

where $d$ is the coefficient dimension and

$$Z_{aB} = 2^{ad/2}\pi^{d(d-1)/4}|S|^{a/2}\prod_{i=1}^{d}\Gamma(\frac{a+1-i}{2}) \tag{A.11}$$

The KL divergence between two Wishart distributions $W(a_1,B_1)$ and $W(a_2,W_2)$ can be written as:

$$KL(a_1,B_1||a_2,W_2) = log\,\frac{Z_{a_2B_2}}{a_1B_1} + \frac{(a_1-a_2)}{2} < log\,W >_1 + \frac{1}{2}a_2tr[B_2^{-1}B_1-I] \tag{A.12}$$

where

$$< log\,W >_1 = \sum_{k=1}^{d}\Psi(\frac{a_1+1-k}{2}) + d\,log\,2 + log\,|B_1| \tag{A.13}$$

# Bibliography

[1] M. Abramovitz and I. A. Stegun. *Handbook of Mathematical Functions.* Dover Publications, 1972.

[2] J. Ajmera. *Robust audio segmentation.* PhD thesis, EPFL - IDIAP, 2004.

[3] J. Ajmera, H. Bourlard, and I. Lapidot. Unknown-multiple speaker clustering using hmm. *Proceedings of ICSLP 2002*, pages 573–576, 2002.

[4] W. Andrews, M. Kohler, J. Campell, and J. Godfrey. Phonetic, idiolectal and acoustic speaker recognition. *Speaker Odyssey Workshop*, 2001.

[5] H. Attias. Inferring parameters and structures of latent variable models by variational bayes. *Proceedings of the 15th Conference on Uncertainty in Artificial Int elligence*, pages 21–30, 1999.

[6] H. Attias. A variational bayesian framework for graphical models. *Advances in Neural Information Processing Systems*, 12:209–215, 2000.

[7] H. Attias. New em algorithms for source separation and deconvolution. *Proceedingsof the IEEE 2003 International Conference on Acoustics, Speech, and Signal Processing*, 2003.

[8] H. Attias and L. Deng. A new approach to speech enhancement with a microphone array using em and mixture models. *Proceedings of the 7th International Conference on Spoken Language Processing*, 2002.

[9] H. Attias, L. Deng, A. Acero, and J.C. Platt. A new method for speech denoising and robus t speech recognition using probabilistic models for clean speech and for noise. *Proceedings of the 7th European Conference on Speech Communication and Technology*, 2001.

[10] H. Attias, J.C. Platt, A. Acero, and L. Deng. Speech denoising and dereverberation using probabilistic models. *Advances in Neural Information Processing Systems*, 13:758–764, 2001.

[11] B. Baker, V. Robbie, M. Manson, and Sridha Sridharan. Improved phonetic and lexical speaker recognition through map adaptation. *Speaker Odyssey Workshop*, 2004.

[12] R. Bakis. Transcription of broadcast news shows with the ibm large vocabulary speech recognition system. *Proceedings of the speech recognition workshop*, 1997.

[13] D. Barber and C.M. Bishop. Ensemble learning for multi-layer networks. *Advances in Neural Information Processing Systems*, 10:395–401, 1998.

[14] O. Barndor-Nielsen. *Information and Exponential Families in Statistical Theory.* John Wiley & Sons, 1978.

[15] L.E. Baum and al. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.

[16] L.E. Baum and J.A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model of ecology. *Bulletin of America Mathematical Society*, 73:360–363, 1967.

[17] Thomas Bayes. Essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 1793.

[18] M. Beal. *Variational Algorithm for Approximate Bayesian Inference.* PhD thesis, The Gatsby Computational Neuroscience Unit, University College London, 2003.

[19] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden markov model. *Neural Information Processing Systems*, 14, 2002.

[20] P. Beyerlein. Large vocabulary continuous speech recognition of broadcast news the philips/rwth approach. *Speech Communication*, 37(1-2):109–131, 2002.

[21] J. Bilmes. A gentle tutorial on the em algorithm and its application to paramete r estimation for gaussian mixture and hidden markov models. Technical Report ICSI-TR-97-021, ICSI, 1997.

[22] F. Bimbot and L. Mathan. Text-free speaker recognition using an arithmetic harmonic sphericity measure. *Proceedings of EUROSPEECH'93*, 1993.

[23] C. M. Bishop, S. Spiegelhalter, and J. Winn. Vibes: A variational inference engine for bayesian networks. In S. Becker, S. Thrun, and K. Obermeyer, editors, *Advances in Neural Information Processing Systems*, volume 15, 2002.

[24] C. M. Bishop and M. Svensén. Robust bayesian mixture modelling. *Proceedings Twelfth European Symposium on Artificial Neural Net works*, pages 69–74, 2004.

[25] C. M. Bishop and J. Winn. Structured variational distributions in vibes. In C. M. Bishop and B. Frey, editors, *Proceedings Artificial Intelligence and Sta tistics*. Society for Artificial Intelligence and Statistics, 2003.

[26] C.M. Bishop. Variational pca. *Proceedings of ICANN*, 1999.

[27] C.M. Bishop, D. Spiegelhalter, and J. Winn. Vibes: a variational inference engine for bayesian networks. *Advances in Neural Information Processin Systems*, 15, 2003.

[28] J.F. Bonastre, P. Delacourt, C. Fredouille, T. Merlin, and C. Wellekens. A speaker tracking system based on speaker turn detection for nist evalutations. *Proceedings of Int. Conf. on Acoustic Speech and Signal Processing (ICASSP)*, 2000.

[29] L.D. Brown. *Fundamentals of Statistical Exponential Families.* Institute of Mathematical Statistics, 1986.

[30] P. Cheeseman and J. Stutz. Bayeisan classification (autoclass): Theory and results). *Advances in Knowledge Discovery and Data Mining*, pages 153–180, 1996.

[31] S. Chen and P. Gopalakrishnan. Speaker, environment and channel change detection and clustering via the bayesian information criterion. *Proceedings of the DARPA Workshop*, 1998.

[32] S.S. Chen and al. Automatic transcription of broadcast news. *Speech Communication*, 37(1-2):69–87, 2002.

[33] A. Cohen and V. Lapidus. Unsupervised text independent speaker classification. *Proc. of the Eighteenth Convention of Electrical and Electronics Engineers in Israel*, 1995.

[34] A. Corduneanu and C. M. Bishop. Variational bayesian model selection for mixture distributions. *Proceedings Eighth International Conference on Artificial Intelligence and Statistics*, pages 27–34, 2001.

[35] T. Cover and J. Thomas. *Elements of Information Theory.* Wiley, 1991.

[36] Heckerman D. A tutorial on learning with bayesian network. In M. I. Jordan, editor, *Learning in Graphical Models.* Kluwer, 1998.

[37] P. Delacourt, D. Kryze, and C.J. Wellekens. Detection of speaker changes in an audio document. *Proceedings of the European Conf. on Speech Communication and Technology (EUROSPEECH)*, 1999.

[38] P. Delacourt and C.J. Wellekens. Distbic: a speaker based segmentation for audio data indexing. *Speech Communication*, 32, 2000.

[39] A. P. Dempster, N.M. Laird, and D.B. Rubin. Maximum-likelihood from incomplete data via the em algorithm. *Journal of Royal Statistical Society ser. B.*, 39:1–38, 1977.

[40] P. Diaconis and D. Ylvisaker. Conjugate priors for exponential families. *The Annals of Statistics*, 2(7):269–281, 1979.

[41] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* John Wiley & Sons, Second Edition, 2001.

[42] J.D. Ferguson. Hidden markov analysis: an introduction. In *Hidden Markov Models for Speech.* Institute for Defense Analyses,Princeton NJ, 1980.

[43] R.P. Feynman. *Statistical Mechanics: A set of Lectures.* Perseus Reading, MA, 1972.

[44] B.J. Frey, L. Deng, A. Acero, and T. Kristjansson. Algonquin: Iterating laplace's method to remove multiple types o f acoustic distortion for robust speech recognition. *Eurospeech*, 2001.

[45] J.L. Gauvain, L. Lamel, and G. Adda. The limsi broadcast news transcription system. *Speech Communication*, 37(1-2):89–108, 2002.

[46] J.L. Gauvain and C. Lee. Maximum a-posteriori estimation for multivariate gaussian mixture obs ervations of markov chains. *IEEE Transactions SAP*, 2:291–298, 1994.

[47] J.L. Gauvain and C.H. Lee. Improved acoustic modeling with bayesian learning. *ICASSP-92*, 1992.

[48] Z. Ghaharamani and G.E. Hinton. Switching stae-space models. Technical Report CRG-TR-96-3, Toronto: Department of Computer Science, University of Toronto, 1996.

[49] Z. Ghaharamani and M.I. Jordan. Factorial hidden markov models. *Machine Learning*, 29:245–273, 1997.

[50] A. Ghahramani and H. Attias. Online variational bayesian learning. *Slides presented at NIPS*, 2000.

[51] Z. Ghahramani and M.J. Beal. Variational inference for bayesian mixtures of factor analyzer. *Advances in Neural Information Processing System*, 12, 2000.

[52] Z. Ghahramani and M.J. Beal. Propagation algorithms for variational bayesian learning. *Advances in Neural Information Processing Systems*, 13, 2001.

[53] Z. Ghahramani and G.E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12(4), 2000.

[54] H. Gish and H. Schmidt. Text independent speaker identification. *IEEE Signal Processing Magazine*, 1994.

[55] H. Gish, M.H. Siu, and R. Rohlicek. Segregation of speakers for speech recognition and speaker identification. *Proceedings of ICASSP'91*, 1991.

[56] S.F. Gull. Bayesian inductive inference and maximum entropy. *Maximum Entropy and Bayesian Methods in Science and Engineering vol 1: Foundations*, pages 53–74, 1988.

[57] G.E. Hinton and D.van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Proceedings of the 6th Annual Workshop on Computational learning theory.* ACM Press, 1993.

[58] X.D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov Models for Speech Recognition.* Edinburgh, U.K., Edimburgh University Press, 1990.

[59] T. Jaakkola. *Variational methods for inference and estimation in graphical models.* PhD thesis, Massachusetts Institute of Technology, Cambridge MA, 1997.

[60] T. S. Jaakkola and M. I. Jordan. Recursive algorithms for approximating probabilities in graphical models. *Advances in Neural Information Processing Systems*, 9, 1997.

[61] T. S. Jaakkola and M. I. Jordan. Variational probabilistic inference and the qmr-dt network. *Journal of Artificial Intelligence Research*, 10:291–322, 1999.

[62] T. Jebara. *Discriminative, generative and imitative learning.* PhD thesis, Media Laboratory, MIT, December 2001.

[63] H. Jeffreys. *Theory of Probability.* Oxford Univ. Press, 1939.

[64] F. Jelinek. Continuous speech recognition by statistical methods. *Proc. of the IEEE*, 64(4):532–556, 1976.

[65] F. Jelinek. *Statistical Methods for Speech Recognition.* Cambridge, MA, MIT Press, 1998.

[66] J. L. W. V. Jensen. Sur les fonctions convexes et les inegalité entre les valeurs moyennes. *Acta Math.*, 30:175–193, 1906.

[67] H. Jin, F. Kubala, and R. Schwartz. Automatic speaker clustering. *Proceedings of the DARPA speech recognition workshop*, 1997.

[68] T. Jitsuhiro and S. Nakamura. Increasing the mixture components of non-uniform hmm structures based on a variational bayesian approach. *Proceedings of of 8th International Conference on Spoken Language Processing (ICSLP)*, 2003.

[69] T. Jitsuhiro and S. Nakamura. Variational bayesian approach for automatic generation of hmm topologies. *Proc.of ASRU2003*, 2003.

[70] S. Johnson and P. Woodland. Speaker clustering using direct maximization of the mllr-adapted likelihood. *Proc of ICSLP '98*, pages 1775–1778, 1998.

[71] S. Johnson and P. Woodland. Speaker clustering using direct maximization of the mllr-adapted likelihood. *Proceedings of ICSLP'98*, 1998.

[72] M.I. Jordan. *Learning in Graphical Models.* Kluwer Academic Publishers, 1998.

[73] M.I. Jordan, Z. Ghahramani, T. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[74] M.I. Jordan, Z. Ghahramani, and L.K. Saul. Hidden markov decision tree. *Advances in Neural Information Processing*, 9, 1997.

[75] B.H. Juang, W. Hou, and C.H. Lee. Minimum classification error rate methods for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 5:pp 257–265, May 1997.

[76] S. Kajarekar, A. Ferrer, K. Venkataraman, E. Sonmez, E. Shriberg, H. Stolcke, H. Bratt, and R.R. Gadde. Speaker recognition using prosodic and lexical features. *Procedeeings IEEE ASR*, 2003.

[77] R.E. Kass and A.E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90:773–775, 1995.

[78] T. Kemp, M. Schmidt, M. Westphal, and W. Waibel. Strategies for automatic segmentation of audio data. *Proceedings of Int. Conf. on Acoustic Speech and Signal Processing (ICASSP)*, 2000.

[79] T. Kristjansson. *Speech Recognition in Adverse Environments: A Probabilistic Ap proach.* PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, April 2002.

[80] T. Kristjansson and B.J. Frey. Accounting for uncertainty in observations: A new paradigm for robust automatic speech recognition. *IEEE International Conference on Acoustics, Speech and Signal P rocessing (ICASSP)*, 2002.

[81] F. Kubala. The 1996 bbn byblos hub-4 transcription system. *Proceedings of the speech recognition workshop*, 1997.

[82] O.-W. Kwon, K.-L. Chan, and T.-W. Lee. Speech feature analysis using variational bayesian pca. *IEEE Signal Processing Letters*, 2003.

[83] I. Lapidot. Som as likelihood estimator for speaker clustering. *Proc. of EUROSPEECH-2003*, pages 3001–3004, 2003.

[84] I. Lapidot and H. Guterman. Resolution limitation in speakers clustering and segmentation problems. *Proceedings of ODYSSEY-2001*, pages 169–174, 2001.

[85] M. H. Law, A. K. Jain, and M. A. Figueriredo. Features selection in mixture based clustering. *NIPS*, 2002.

[86] L.J. Lee, H. Attias, and L. Deng. Variational inference and learning for segmental switc hing state space models of hidden speech dynamics. *Proceedings of the IEEE 2003 International Conference on Acoustics, Speech, and Signal Processing*, 2003.

[87] L.J. Lee, H. Attias, and L. Deng. A multimodal variational approach to learning and infe rence in switching state space models. *Proceedings of the 2004 IEEE Internationa l Conference on Acoustics, Speech, and Signal Processing*, 2004.

[88] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantizer design. *IEEE Transaction on Communication COM-28(1)*, pages 84–95, 1980.

[89] D. Liu and F. Kubala. Online speaker clustering. *Proc of ICASSP 2004, Montreal*, 2004.

[90] J.F. Lopez and D.P.W. Ellis. Using acoustic condition clustering to improve acousitc change detection on broadcast news. *Proceedings of Int. Conf. on Spoken Language and Processing*, 2000.

[91] D. J. C. MacKay. Bayesian faq. *http://www.inference.phy.cam.ac.uk/mackay/Bayes-FAQ.html.*

[92] D. J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[93] D. J. C. MacKay. Probable networks and plausible predictions-a review of practical bayesian methods for supervised neural networks. *Network:Comput. Neural Syst.*, 6:469–505, 1995.

[94] D. J. C. MacKay. Ensemble learning for hidden Markov models. `http://www.inference.phy.cam.ac.uk/mackay/abstracts/ensemblePaper.html`, 1997.

[95] D. J. C. MacKay. Choice of basis for laplace approximation. *Machine Learning*, 33(1), 1998.

[96] D. J. C. MacKay. A problem with variational free energy minimization. `http://www.inference.phy.cam.ac.uk/mackay/abstracts/minima.html`, 2001.

[97] D. J. C. MacKay and L.C. Peto. A hirarchical dirichlet language model. *Natural Language Engineering*, 1(3):1–19, 1995.

[98] G. J. McLachlan and D. Peel. *Finite mixture models.* John Wiley & Sons 1st ed., 2001.

[99] S. Meignier, J. F. Bonastre, and S. Igounet. E-hmm approach for learning and adapting sound models for speaker indexing. *A Speaker Odyssey 2001, Crete*, June 2001.

[100] S. Meignier, J.-F. Bonastre, and I. Magrin-Chagnolleau. Speaker utterances tying among speaker segmented audio documents using hierarchical classification: Towards speaker indexing of audio databases. *Proc of ICSLP 2002 Denver USA*, 2002.

[101] Daniel Moraru. *segmentation en locuteurs de documents audios et audio-visuels : application a la recherche d'information multimedia.* PhD thesis, CLIPS IMAG, 2004.

[102] K. Mori and S. Nakagawa. Speaker change detection and speaker clustering using vq distortion for broadcast news speech recognition. *Procedings of Int. Conf. on Pattern Recognition (ICPR)*, 2002.

[103] R. Neal and G. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models.* Kluwer, 1998.

[104] R.M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

[105] M. Nishida and T. Kawahara. Unsupervised speaker indexing using speaker model selection based on bayesian information criterion. *Proceedings of ICASSP*, 2003.

[106] Y. Normandin. *Hidden Markov Models, Maximum Mutual Information Estimation and the Speech Recognition Problem.* PhD thesis, Dept. of Elect. Eng., McGill University, Montreal, 1991.

[107] J. O. Olsen. Separation of speaker in audio data. *Proceedings of the European Conference on Speech Technology*, pages 355–358, 1995.

[108] W. Penny. Kullback-liebler divergences of normal,gaussian, dirichlet and wishart densities. Technical report, Wellcome Department of Cognitive Neurology, 2001.

[109] L.M. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition.* Englewood Cliffs NJ, Prentice Hall, 1993.

[110] C.E. Rasmussen. The infinite gaussian mixture model. *Advances in Neural Information Processing Systems*, 12:554–560, 2000.

[111] C.E. Rasmussen and Z. Ghaharamani. Bayesian monte carlo. *Advances in Neural Information Processing Systems*, 14, 2003.

[112] D.A. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Communication*, 17:91–108, August 1995.

[113] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10, 2000.

[114] C. J. Van Rijsbergen. *Information Retrieval.* London, Butter worths, 1979.

[115] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society, Series B (Methodological)*, 49, 1987.

[116] R. Rockafellar. *Convex Analysis.* Princeton University Press, 1972.

[117] W. Roush. Bayesian machine learning in 10 emerging technologies that will change our world. *MIT's magazine of innovation technology review*, February 2004.

[118] M. Sato. Online model selection based on the variational bayes. *Neural Computation*, 13(7):1649–1681, 2001.

[119] L.K. Saul and M.I. Jordan. Exploiting tractable substructures in intractable networks. *Advances in Neural Information Processing Systems*, 8, 1996.

[120] G. Schwartz. Estimation of the dimension of a model. *Annals of Statistics*, 6, 1978.

[121] M. Seeger. Bayesian model selection for support vector machines, gaussian processes and other kernel classifiers. *Neural Information Processing Systems*, 12, 2000.

[122] J. Shao. An asymptiotic theory for linear model selection. *Statistica Sinica*, 7:221–264, 1997.

[123] M. Siegler, A. Jain, U. Raj, and R.M. Stern. Automatic segmentation, classification and clusering of broadcast news. *DARPA Speech Recognition Workshop*, pages 97–99, 1997.

[124] A. Solomonoff, A. Mielke, M. Schmidt, and H. Gish. Clustering speakers buy their voices. *Proceedings of ICASSP'98*, 1998.

[125] P. Somervuo. Speech modeling using variational bayesian mixture of gaussians. *Proceedings of 7th International Conference on Spoken Language Processing (ICSLP)*, 2002.

[126] P. Somervuo. Comparison of ml, map, and vb based acoustic models in large vocabu lary speech recognition. *Proc of 8th International Conference on Spoken Language Process ing (ICSLP)*, 2004.

[127] A. Tritschler and R. Gopinath. Improved speaker segmentation and segments clustering using the bayesian information criterion. *Proceedings of EUROSPEECH'99*, pages 679–682, 1999.

[128] F. Valente and C. Wellekens. Variational bayesian gmm for speech recognition. *Proceedings of 8th european conference on speech communication and technology (EUROSPEECH)*, 2003.

[129] F. Valente and C. Wellekens. Regroupement bayesien variationnel des locuteurs. *Proceedings of 25èmes Journées d'étude de la parole (JEP)*, 2004.

[130] F. Valente and C. Wellekens. Scoring unknown speaker clustering : Vb vs. bic. *Proceedings of the 8th Biennial Conference of International Conference on Spoken Language Proces sing(ICSLP)*, 2004.

[131] F. Valente and C. Wellekens. Variational bayesian adaptation for speaker clustering. *Proceedings of the 30th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004.

[132] F. Valente and C. Wellekens. Variational bayesian feature selection for gaussian mixture models. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[133] F. Valente and C. Wellekens. Variational bayesian speaker clustering. *Proceedings of Odyssey'2004, The speaker and language recognition workshop*, 2004.

[134] F. Valente and C. Wellekens. Variational bayesian feature saliency for audio type classification. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[135] A. Vandecastseye and J.P. Martens. A fast accurate and stream-based speaker segmentation and clustering algorithm. *Proceedings of the European Conf. on Speech Communication and Technology*, 2003.

[136] A.J. Viterbi. Error bounds for convolutional codes and an asymptotical optimum decoding algorithm. *IEEE Trans. on Information Theory*, 13(2):260–269, 1967.

[137] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda. Application of variational bayesian approach to speech recognition. *Proceedings of NIPS15*, 2002.

[138] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda. Application of variational bayesian estimation and clustering to acoustic model adaptation. *Proceedings of ICASSP'03*, 2003.

[139] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda. Variational bayesian estimation and clustering for speech recognition. *IEEE transaction on Speech and Audio Processing*, 12:pp. 365–38, 2004.

[140] S. Watanabe and A. Nakamura. Robustness of acoustic model topology det ermined by vbec (variational bayesian estimation and clustering for speech recognition) for different speech data sets. *Proc. Workshop on statistical modeling approarch for speech recognition - Beyond HMM*, pages 55–60, 2004.

[141] S. Watanabe, A. Sako, and A. Nakamura. Automatic determination of acoustic model topology using variational bayesian estimation and clustering. *Proc. ICASSP'04*, pages 813–816, 2004.

[142] S. Waterhouse, D.J.C. MacKay, and T. Robinson. Bayesian methods for mixtures of experts. In *Advances in Neural Information Processing Systems 8*. MIT Press, 1996.

[143] P. Woodland, M. Gales, D. Pye, and S. Young. The development of the 1996 htk broadcast new transcription system. *Proceedings of the speech recognition workshop*, 1997.

[144] S. Young and al. *The HTK book*. 2002.

[145] B. Zhou and J.H.L. Hansen. Unsupervised audio stram segmentation and clustering via the bayesian information criterion. *Proceedings of the Int. Conf. on Spoken Language Processing (ICSLP)*, 2000.

[146] P. Zolfaghari, S. Watanabe, A. Nakamura, and S. Katagiri. Bayesian modelling of the speech spectrum using mixture of gaussians. *Proceedings of the 30th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004.

[147] G. Zoubin. Tutorial on bayesian methods for machine learning. *ICML*, 2004.