Variational Bayesian Learning for Gaussian Mixture Models and Hidden Markov Models

Technical Report RR-03-079

Fabio Valente, Christian Wellekens

July 3, 2003

Contents

1 Variational Bayesian Learning **5**
- 1.1 Introduction . . . 5
- 1.2 Variational Bayesian Learning, Maximum Likelihood and EM algorithm . . . 5
  - 1.2.1 EM as special case of VB Learning . . . 6
- 1.3 Variational bayesian learning . . . 6
  - 1.3.1 Another point of view . . . 8
  - 1.3.2 Variational Bayesian Learning and MAP . . . 8

2 Gaussian Mixture Models **11**
- 2.1 Introduction . . . 11
- 2.2 Gaussian Mixture Models and EM algorithm . . . 11
- 2.3 Variational Bayesian Gaussian Mixture Models I . . . 12
- 2.4 Variational Bayesian Gaussian Mixture Models II . . . 14

3 Experiments **17**
- 3.1 Experiments on synthetic data . . . 17
- 3.2 Conclusion . . . 20

4 Variational bayesian HMM **23**
- 4.1 Discrete density HMM . . . 23
  - 4.1.1 Optimization of Q_A . . . 24
  - 4.1.2 Optimization of Q_B,Q_π . . . 25
  - 4.1.3 Optimization of $Q_S(S)$ . . . 25
  - 4.1.4 Another point of view . . . 26
- 4.2 Variational bayesian HMM with GMM . . . 27

Chapter 1

Variational Bayesian Learning

1.1 Introduction

This report aims at applying variational bayesian learning to two class of well-known models: gaussian mixture models and hidden Markov models. Variational bayesian (VB) learning is a relatively new approach to model learning that offers many advantages respect to classical ML and MAP learning that can be seen as a particular case of VB learning.

1.2 Variational Bayesian Learning, Maximum Likelihood and EM algorithm

Let's consider a probabilistic model with some observed variables Y, some hidden variables X and model parameters Θ. The goal of the learning process is to find the optimal value of Θ given some instances of Y.

The learning process is typically done maximizing the model likelihood $p(Y|\Theta)$. Let's define

$$p(Y|\Theta) = \frac{p(Y,X|\Theta)}{p(X|Y,\Theta)} \tag{1.1}$$

the log-likelihood $L(\Theta) = logp(Y|\Theta)$can then be expressed like:

$$L(\Theta) = log\, p(Y,X|\Theta) - log\, p(X|Y,\Theta) \tag{1.2}$$

The variational approach suppose that the true hidden variables probability $p(X)$ can be approximated by the distribution Q_λ (where λ are the distribution parameters). So integrating out H we can write:

$$L(\Theta) = \int Q_\lambda(H)\ log\, p(X,Y|\Theta)dX - \int Q_\lambda(H)\ log\, p(X|Y,\Theta)dX \tag{1.3}$$

After a few manipulation we can write 1.3 as:

$$L(\Theta) = F(\lambda, \Theta) + D(Q||p) \tag{1.4}$$

where

$$F(\lambda, \Theta) = \int Q_\lambda(X) \, log \, \frac{p(X, Y|\Theta)}{Q_\lambda(X)} \tag{1.5}$$

$$D(Q||p) = \int Q_\lambda(X) \, log \, \frac{Q_\lambda(X)}{p(X|Y, \Theta)} \tag{1.6}$$

The KL-divergence $D(Q||p)$ represent the distance between the approximated hidden variables distribution function and the real hidden variable distribution $p(X|Y, \Theta)$.

It can be proved that $(Q||p) \geq 0$ in other words $L(\Theta) \geq F(\lambda, \Theta)$ with equality if $Q_\lambda = p(X|Y, \Theta)$.

The variational learning consists in using the lower bound $F(\lambda, \Theta)$ as function to optimize instead of the log-likelihood. An EM like algorithm with an approximated E step can be used:

- *Approximated E-step*: fix model parameters at Θ_{t-1}, and update variational parameters λ to maximize $F(\lambda, \Theta)$.
- *M-step*: fix variational parameters at λ_t, and update model parameters Θ to maximize $F(\lambda, \Theta)$.

1.2.1 EM as special case of VB Learning

When $Q_\lambda(H) = p(X|Y, \Theta)$ the learning algorithm becomes the classical EM algorithm for ML training because $D(Q||p) = 0$ and the strict lower bound $F(\lambda, \Theta)$ is equal to the log-likelihood. In this case the approximated E-step becomes an exact E-step because the expectation is done w.r.t. the true distribution and not the approximated one.

1.3 Variational bayesian learning

An important use of variational learning is the approximation of posterior densities in bayesian learning.

Let's define $p(Y)$ the likelihood of a probabilistic model after the model's parameters Θ has been integrated out. Let's write:

$$p(Y) = \frac{p(Y, \Theta)}{p(\Theta|Y)} \tag{1.7}$$

and so:

$$log \, p(Y) = log \, p(Y, \Theta) - log p(\Theta|Y) \tag{1.8}$$

Again we suppose that the real posterior distribution $p(\Theta|Y)$ can be approximated with a distribution $q(\Theta|Y)$ that we will call *variational posterior* and so integrating out model parameters Θ, we obtain:

$$log\, p(Y) = \int q(\Theta|Y) log\, p(Y,\Theta) d\Theta - \int q(\Theta|Y) log\, p(\Theta|Y) \tag{1.9}$$

Multiplying the probability $p(\Theta|Y)$ top and bottom by $q(\Theta|Y)$ and rearranging gives:

$$log\, p(Y) = F(\Theta) + D(q(\Theta|Y)||p(\Theta|Y)) \tag{1.10}$$

where

$$F(\Theta) = \int q(\Theta|Y)\, log\, \frac{p(Y,\Theta)}{p(\Theta|Y)} d\Theta \tag{1.11}$$

As before, $D(q(\Theta|Y)||p(\Theta|Y))$ is the KL-divergence between the approximate posterior distribution and the true posterior distribution. Because of the fact $D(Q|p) \geq 0$ with equality when $Q = p$ we obtain

$$log\, p(Y) \geq F(\Theta) \tag{1.12}$$

Maximizing $F(\Theta)$ means make the approximate posterior as close as possible to the true posterior. Using Bayes rule $p(Y,\Theta) = p(Y|\Theta)p(\Theta)$ we can write:

$$F(\Theta) = \int q(\Theta|Y)\, log\, p(Y|\Theta) d\Theta - D(q(\Theta|Y)||p(\Theta)) \tag{1.13}$$

The KL-divergence in this case is the divergence between the approximating posterior and the prior. This means that the quantity $D(q(\Theta|Y)||p(\Theta))$ penalizes more complex models.

The bayesian framework can be extended to models with hidden variables X. In this case $F(\Theta)$ can be written:

$$F(\Theta, X) = \int q(\Theta, X|Y)\, log\, \frac{p(Y,X,\Theta)}{q(\Theta, X|Y)} d\Theta\, dX \tag{1.14}$$

Using the following factorization : $q(\Theta, X|Y) = q(\Theta|Y)q(X|\Theta)$ and $p(Y,X,\Theta) = p(Y,X|\Theta)p(\Theta)$ we can write:

$$F(\Theta, X) = \int q(\Theta|Y)q(X|Y) log\, \frac{p(X,Y|\Theta)}{q(X|Y)} d\Theta dX - D(q(\Theta|Y)||p(\Theta)) \tag{1.15}$$

The first term is the average likelihood while the second term is the KL-divergence between the approximating posterior and the prior. It can be demonstrated that when $N \to \infty$ the KL-divergence reduces to $(|\Theta_0|/2) log N$ which is linear in the number of parameters $|\Theta_0|$ i.e. eq. 1.13 corresponds to the Bayesian Information Criterion.

The optimization of 1.13 can be done using a free-form optimization and an EM-like algorithm as described in [1].

- *E step*: compute the posterior over the hidden nodes solving $\partial F(\Theta, X)/\partial q(X) = 0$ to obtain:

$$q(X) \propto e^{<log\, p(X,Y|\Theta)>_\Theta} \tag{1.16}$$

The notation $< . >_\Theta$ means that the average is taken w.r.t $q(\Theta)$.

- *M step*: compute the posterior distribution over the parameters solving $\partial F(\Theta, X)/\partial q(\Theta) = 0$ to obtain:

$$q(\Theta) \propto e^{<\log p(X,Y|\Theta)>_X} p(\Theta) \tag{1.17}$$

The notation $< . >_X$ means that the average is taken w.r.t $q(X)$.

It's important to notice that if we choose $p(\Theta)$ such that $q(\Theta) \propto f(\Theta)p(\Theta)$ belongs to the same family i.e. $p(\Theta)$ is conjugate to $f(\Theta)$, the learning procedures will simply consists in updating hyperparameters, transforming the prior parameters into posterior parameters.

1.3.1 Another point of view

Another way of looking the optimization process was proposed by McKay and Neal (see [2]). The first assumption is that there is no difference between latent variables and model parameters: both of them corresponds to unobserved stochastic variables and can be treated on the same level. The second assumption is considering the approximated posterior factorizable:

$$Q(Z) = \prod_i Q_i(Z_i) \tag{1.18}$$

where Z is the ensemble of model parameters and latent variables.

So we can maximize in the variational sense of the term with respect to $Q_i(Z_i)$ keeping all others $Q_j(Z_j)$ with $j \neq i$ fixed.

So we can rewrite 1.13 as:

$$\log Q_i(Z_i) =< logP(Y, Z) >_{j \neq i} + const \tag{1.19}$$

where $< . >_t$ denotes an expectation with respect to the distribution $Q_t(X_t)$. So taking exponentials of both sides and normalizing we obtain:

$$Q_i(Z_i) = \frac{exp < logP(Y, Z) >_{j \neq i}}{\sum_{Z_i} exp < logP(Y, Z) >_{j \neq i}} \tag{1.20}$$

Those equations are coupled equations since the solution for Q_i depends on expectations with respect to the other factors $Q_{i \neq j}$.

So the optimization process consists in initializing each of the Q_i and then iteratively update each factor at time using values computed during previous iterations.

1.3.2 Variational Bayesian Learning and MAP

Classical MAP learning can be interpreted as a special case of variational learning. To obtain the MAP formulation it's enough to set $Q_\lambda(\theta|Y) = \delta(\theta - \theta^{'})$. Free energy maximization becomes:

$$\begin{aligned} max_{Q(\theta)} F(\theta) &= max_{\theta'} \int \delta(\theta - \theta^{'}) log[p(Y|\theta)p(\theta)] d\theta \\ &= max_{\theta'} log[p(Y|\theta^{'})p(\theta^{'}))] \end{aligned} \tag{1.21}$$

The term $\int Q_\lambda(\Theta) log(Q_\lambda(\Theta) dX$ is constant and doesn't play any role in the maximization, for this reason has been dropped. Generalization to hidden variables case is easygoing.

The variational bayesian approach generalizes the MAP because it carries information about the uncertainty in Θ even if it's only an approximation. Furthermore classical MAP doesn't allow any model selection and doesn't bring any information on the model quality.

Chapter 2

Gaussian Mixture Models

2.1 Introduction

One of the most popular model widely used in speech recognition is the mixture of gaussians. In this chapter we will give an overview of current works for learning variational bayesian GMM (based on [1] and [3]) after giving a fast review of classical EM algorithm.

As general notation we will denote the pdf. associated with a M components guassians mixture as:

$$p(y|\Theta) = \sum_{i=1}^{M} \pi_i p_i(y|\Theta_i) = \sum_{i=1}^{M} \pi_i N(\mu_i, \Sigma_i) \tag{2.1}$$

where $\sum_{i=1}^{M} \pi_i = 1$ and each p_i is a gaussian density function parameterized by $\Theta_i = \{\mu_i, \Sigma_i\}$. Let's denote with $S = \{s_t\}_{t=1}^{N}$ the hidden variables that indicate the component that generated the t^{th} observation y_t.

2.2 Gaussian Mixture Models and EM algorithm

A current approach for finding optimal parameters for the GMM consists in the application of the EM algorithm as described in the previous chapter.

In fact in the case of a GMM the probability of hidden variables is trivial: given an observation y_t, the probability of the hidden variable s_t can be expressed as:

$$p(s_t = s|y_t) = \frac{\pi_s p_s(y_t|\Theta_s)}{\sum_{i=1}^{M} \pi_i p_i(y|\Theta_i)} \tag{2.2}$$

So it's possible to compute the exact E step in which the expectation is done w.r.t. the true hidden variable distributions. Applying EM algorithm, well known update formulas for GMM parameters can be obtained.

$$\pi_s = \frac{1}{N} \sum_{t=1}^{N} p(s|y_t, \Theta) \tag{2.3}$$

$$\mu_s = \frac{\sum_{t=1}^{N} y_t p(s|y_t,\Theta)}{\sum_{t=1}^{N} p(s|y_t,\Theta)} \tag{2.4}$$

$$\Sigma_s = \frac{\sum_{t=1}^{N} p(s|y_t,\Theta)(y_t-\mu_s)(y_t-\mu_s)^T}{\sum_{t=1}^{N} p(s|y_t,\Theta)} \tag{2.5}$$

2.3 Variational Bayesian Gaussian Mixture Models I

An interesting application of variational bayesian learning is the GMM training. Let's specify again the model:

$$p(y|\Theta) = \sum_{s=1}^{M} \pi_i p_i(y|\Theta_i) = \sum_{s=1}^{M} p(y|s_n,\Theta)p(s_n = s|\Theta) \tag{2.6}$$

where $p(y|s_n,\Theta) = N(\mu_s,\Sigma_s)$ and $p(s_n = s|\Theta) = \pi_s$. Let's define conjugate priors on parameters $\Theta = (\mu_s,\pi_s,\Sigma_s)$. We choose the following conjugate priors for parameters distributions:

$$p(\{\pi_s\}) = D(\lambda^0) \tag{2.7}$$

$$p(\mu_s|\Sigma_s) = N(\rho^0,\beta^0\Sigma_s) \tag{2.8}$$

$$p(\Sigma_s) = W(\nu^0,\Phi^0) \tag{2.9}$$

where $D(.)$ denotes a Dirichlet distribution, $N(.)$ a normal distribution and $W(.)$ a Wishart distribution.

Let's define $q(\Theta)$ the approximation of parameters posterior and let's assume that it can be factorized like $q(\Theta) = q(\{\pi_s\})\prod_s q(\mu_s,\Sigma_s)$.

Now it's possible to apply the EM-like algorithm defined in [1]. In the E-step we obtain:

$$\gamma_s^n = q(s_n = s|y_n) \propto \tilde{\pi_s}\tilde{\Sigma_s}^{1/2} e^{-(y_n-\rho_s)^T\bar{\Sigma}_s(y_n-\rho_s)/2}\, e^{-d/2\beta_s} \tag{2.10}$$

where

$$log\tilde{\pi_s} =< log\pi_s >= \psi(\lambda_s) - \psi(\sum_{s'}\lambda_{s'}) \tag{2.11}$$

$$log\tilde{\Sigma_s} =< log|\Sigma_s| >= \sum_{i=1}^{d}\psi((\nu_s+1-i)/2) - log|\Phi_s| + dlog2 \tag{2.12}$$

$$\bar{\Gamma_s} = a_s B_s^{-1} \tag{2.13}$$

This expression is equivalent to classical ML accumulator where parameters have been integrated out. $\psi(s)$ is the digamma function defined as $d\,log\Gamma(x)/dx$.

Then in the M-step is possible to update parameters of the gaussian mixtures as well as new parameters posteriors that will have the same form of parameters priors because of the choice we have done (conjugate priors).

For the gaussian mixture parameters we will have:

$$\overline{\pi}_s = \frac{1}{N}\sum_{n=1}^{N}\gamma_s^n, \quad \overline{\mu}_s = \frac{1}{\overline{N}_s}\sum_{n=1}^{N}\gamma_s^n y_n, \quad \overline{\Sigma}_s = \frac{1}{\overline{N}_s}\gamma_s^n(y_n-\overline{\mu}_s)(y_n-\overline{\mu}_s)^T \tag{2.14}$$

where $\overline{N}_s = N\overline{\pi}_s$. Parameters posteriors will have the same form of priors:

$$q(\pi_s) = D(\lambda_s), \quad q(\mu_s|\lambda_s) = N(\rho_s, \beta_s\Gamma_s), \quad q(\Gamma_s) = W(\nu_s, \Phi_s) \tag{2.15}$$

Posteriors parameters can be updated following this rules:

$$\lambda_s = \overline{N}_s + \lambda^0, \quad \rho_s = (\overline{N}_s\overline{\mu}_s + \beta^0\rho^0)/(\overline{N}_s + \beta^0), \quad \beta_s = \overline{N}_s + \beta^0$$
$$\nu_s = \overline{N}_s + \nu_0, \quad \Phi_s = \overline{N}_s\overline{\Sigma}_s + \overline{N}_s\beta^0(\overline{\mu}_s - \rho^0)(\overline{\mu}_s - \rho^0)^T/(\overline{N}_s + \beta^0) + \Phi^0 \tag{2.16}$$

We can express the approximate marginal likelihood 1.15:

$$F = \int q(\pi_s)log\frac{p(\pi_s)}{q(\pi_s)}d\pi + \sum_{s=1}^{m}\int q(\Gamma_s)log\frac{p(\Gamma_s)}{q(\Gamma_s)}d\Gamma_s +$$
$$+\sum_{s=1}^{m}\int q(\mu_s|\Gamma_s)log\frac{p(\mu_s|\Gamma_s)}{q(\mu_s|\Gamma_s)}d\mu_s + \sum_{s=1}^{m}\sum_{n=1}^{N}q(s_n)\int q(\pi)log\frac{p(s_n|\pi)}{q(s_n)}d\pi +$$
$$+\sum_{s=1}^{m}\sum_{n=1}^{N}q(s_n)\int\int q(\Gamma_s)q(\mu_s|\Gamma_s)logp(y_n|\Gamma_s,\mu_s,s_n)d\Gamma_s d\mu_s \tag{2.17}$$

First three terms are KL-distance and the last two are the average likelihood.

$$F = -KL_{Dir}(\lambda,\lambda_0) - \sum_{s=1}^{m}KL_W(a_s, B_s; a_0, B_0) +$$
$$-\sum_{s=1}^{m}KL_N(m_s, B_s/(\beta_s a_s); m_0, B_s/(\beta_0 a_s)) + \sum_{s=1}^{m}Lik(s) \tag{2.18}$$

where

$$Lik(s) = \overline{N}_s log\tilde{\pi}_s - \sum_{n=1}^{N}\gamma_s^n log\gamma_s^n + +\frac{\overline{N}_s}{2}(-dlog2 + log\tilde{\Gamma}_s - Tr(\overline{\Gamma}_s)\overline{\Sigma}_s - d/\beta_s) \tag{2.19}$$

KL distances between prior and posterior can be interpreted as a model penality term and the free energy F can be used to select the best model. The KL penalty reduces to BIC penalization term when $N \to \infty$ i.e.:

$$BIC(m) = \sum_{n=1}^{N}log\, p(y_n|\hat{\Theta}) - \frac{N_m}{2}log\, N \tag{2.20}$$

where N_m is the number of parameters in a model of size m. For a GMM $N_m = m(1 + d + d(d+1)/2)$.

2.4 Variational Bayesian Gaussian Mixture Models II

A different version of variational bayesian gaussian mixtures has been proposed by Bishop [3]. Hyperparameters are here optimized using a 'type 2' maximum likelihood technique in which the values of the hyperparameters are chosen to optimize the marginal likelihood of the observed data in which the model parameters have been integrated out.

The mixture model is re-interpreted using the latent variables s_{in} with $s_{in} \epsilon 0, 1$ where $i = 1, .., M \sum_{i=1}^{M} s_{in} = 1$. The latent variables represent the gaussian that generated the y_n samples. Let D be the data set, we can write:

$$P(D|\mu, \Sigma, s) = \prod_{n=1}^{N} \prod_{i=1}^{M} N(y_n|\mu_i, \Sigma_i)^{s_{in}} \tag{2.21}$$

The latent variables s_{in} are a discrete distribution that depends on mixing coefficients π_i

$$P(s|\pi) = \prod_{i=1}^{M} \prod_{n=1}^{N} \pi_i^{s_{in}} \tag{2.22}$$

Then we introduce conjugate priors on means and inverse covariance matrix (we don't need conjugate priors on weights because anyway they will be integrated out).

$$P(\mu) = \prod_{i=1}^{M} N(\mu_i|0, \beta I) \tag{2.23}$$

$$P(\Sigma) = \prod_{i}^{M} W(T_i|\nu, V) \tag{2.24}$$

where β is a fixed parameter, I is the unit matrix, and W is a Wishart distribution with parameters ν and V. It is possible to write:

$$P(D, \mu, \Sigma, s|\pi) = P(D|\mu, \Sigma, s)P(s|\pi)P(\mu)P(\Sigma) \tag{2.25}$$

Now variational bayesian learning can be used to evaluate hyperparameters. Mixing coefficients π will be optimized in a separate M step. So introducing an approximating distribution $Q(\Theta)$ it is possible to write:

$$log P(D|\pi) = log \int P(D, \Theta|\pi) d\Theta = log \int Q(\Theta) \frac{P(D, \Theta|\pi)}{Q(\Theta)} d\Theta \geq \int Q(\Theta) log \frac{P(D, \Theta|\pi)}{Q(\Theta)} d\Theta \tag{2.26}$$

Making the assumption that approximated priors are independent, it is possible to write:

$$Q(\mu, \Sigma, s) = Q_\mu(\mu) Q_\Sigma(\Sigma) Q_s(s) \tag{2.27}$$

Applying method described in 1.3.1, we obtain:

$$Q_s(s) = \prod_{n=1}^{N}\prod_{i=1}^{M} p_{in}^{s_{in}} \tag{2.28}$$

$$Q_\mu(\mu) = \prod_{i=1}^{M} N(\mu_i|m_\mu^{(i)}, \Sigma_\mu^{(i)}) \tag{2.29}$$

$$Q_\Sigma(\Sigma) = \prod_{i=1}^{M} W(\Sigma_i|\nu_\Sigma^{(i)}, V_T^{(i)}) \tag{2.30}$$

where we have defined

$$p_{in} = \frac{\tilde{p}_{in}}{\sum_{j=1}^{(M)} \tilde{p}_{in}} \tag{2.31}$$

$$\tilde{p}_{in} = exp(\frac{< log|\Sigma_i| >}{2} + log\pi_i - \frac{1}{2}Tr\{< \Sigma_i > (y_n y_n^T - < \mu_i > y_n^T - y_n < \mu_i >^T + < \mu_i \mu_i^T >)\}) \tag{2.32}$$

$$\Sigma_\mu^{(i)} = \beta I + < \Sigma_i > \sum_{n=1}^{N} < s_{in} > \tag{2.33}$$

$$m_\mu^{(i)} = \Sigma_\mu^{(i)-1} < \Sigma_i > \sum_{n=1}^{N} y_n < s_{in} > \tag{2.34}$$

$$\nu_\Sigma^{(1)} = \nu + \sum_{n=1}^{N} < s_{in} > \tag{2.35}$$

$$V_\Sigma^{(i)} = V + \sum_{n=1}^{N} y_n y_n^T < s_{in} > - \sum_{n=1}^{N} y_n < s_{in} >< \mu_i^T > + - < \mu_i > \sum_{n=1}^{N} y_n^T < s_{in} > + < \mu_i \mu_i^T > \sum_{n=1}^{N} < s_{in} > \tag{2.36}$$

and assuming that the expected values are:

$$< s_{in} >= p_{in} \tag{2.37}$$

$$< \mu_i >= m_\mu^{(i)} \tag{2.38}$$

$$< \mu_i \mu_i^T >= (\Sigma_\mu^{(i)})^{-1} + m_\mu^{(i)} (m_\mu^{(i)})^T \tag{2.39}$$

$$< \Sigma_i >= \nu_\Sigma^{(i)} (V_\Sigma^{(i)})^{-1} \tag{2.40}$$

$$< log|\Sigma_i| >= \sum_{s=1}^{d} \psi((\nu_\Sigma^{(i)} + 1 - s)/2) + d\, log 2 - log|V_\Sigma^{(i)}| \tag{2.41}$$

Once a form on the approximated priors Q_μ Q_Σ Q_s is assumed, it is possible to compute the lower bound on the marginal likelihood:

$$L = < logP(D|\mu, \Sigma, s) > + < logP(s) > + < logP(\mu) > + < logP(\Sigma) > +$$

$$- < logQ_s(s) > - < logQ_\mu(\mu) > - < logQ_\Sigma(\Sigma) > \tag{2.42}$$

where

$$< logP(D|\mu, \Sigma, s) >= \sum_{i=1}^{M}\sum_{n=1}^{N} < s_{in} > \frac{1}{2} < log|\Sigma_i| > -\frac{d}{2}log(2\pi)+$$

$$-\frac{1}{2}Tr(< \Sigma_i > (y_n y_n^T - y_n < \mu_i^T > - < \mu_i > y_n^T + < \mu_i \mu_i^T >)) \tag{2.43}$$

$$< logP(s) >= \sum_{i=1}^{M}\sum_{n=1}^{N} < s_{in} > log\pi_i \tag{2.44}$$

$$< logP(\mu) >= 2\frac{Md}{2}log(\frac{\beta}{2\pi}) - \frac{\beta}{2}\sum_{i=1}^{M} < \mu_i^T \mu_i > \tag{2.45}$$

$$< logP(\Sigma) >= M\{-\frac{\nu d}{2}log2 - \frac{d(d-1)}{4}log\pi - \sum_{s=1}^{d} log\gamma(\frac{\nu+1-s}{2}) + \frac{\nu}{2}log|V|\}$$

$$+\frac{\nu-d-1}{2}\sum_{i=1}^{M} < log|\Sigma| > -\frac{1}{2}Tr\left(V\sum_{i=1}^{M} < \Sigma_i >\right) \tag{2.46}$$

$$< logQ_s(s) >= \sum_{i=1}^{M}\sum_{n=1}^{N} < s_{in} > log < s_{in} > \tag{2.47}$$

$$< logQ_{\mu(\mu)} >= \sum_{i=1}^{M}\left(-\frac{d}{2}(1+log(2\pi)) + \frac{1}{2}log|\Sigma_\mu^{(i)}|\right) \tag{2.48}$$

$$< logQ_\Sigma(\Sigma) >= \sum_{i=1}^{M}(\frac{-\nu_\Sigma^{(i)} d}{2} + \frac{1}{2}log|\Sigma_\mu^{(i)}| + \sum_{s=1}^{d} log\Gamma\left(\frac{\nu_\Sigma^{(i)}+1-s}{2}\right) +$$

$$\frac{\nu_\Sigma^{(i)}}{2}log|V_\Sigma^{(i)}|) + \frac{\nu_\Sigma^{(i)}-d-1}{2} < log|\Sigma_i| > -\frac{1}{2}Tr(V_\Sigma^{(i)} < \Sigma_i >) \tag{2.49}$$

Once the bound is calculated we have an approximation of $logP(D|\pi)$. It is now possible then to re-estimate new values of π using the so called 'type 2' maximum likelihood technique. Deriving the lower bound and setting the results to zero, it is possible to compute new values of π.

$$\pi_i = \frac{1}{n}\sum_{n=1}^{N} p_{in} \tag{2.50}$$

So the variational bayesian learning will consists in iteratively updating variational parameters and mixing coefficients.

Chapter 3

Experiments

3.1 Experiments on synthetic data

In order to study the efficacity of those three approach, we realized some simple experiments on synthetic data i.e. data generated from a known gaussian mixture.

We used a simple 3 gaussian mixture with mean [2,2] [-2,-2] [5.5 5], diagonal covariance matrix and 0.3 0.5 0.2 as mixing coefficients; 5000 samples were generated following this distribution.

First of all we tried to find the correct dimension of the data set using the BIC criterion. Figure 3.1 shows the plot of $BIC(m)$ function of gaussian mixtures; the maximum is for m=3 that is the effective number of gaussian.

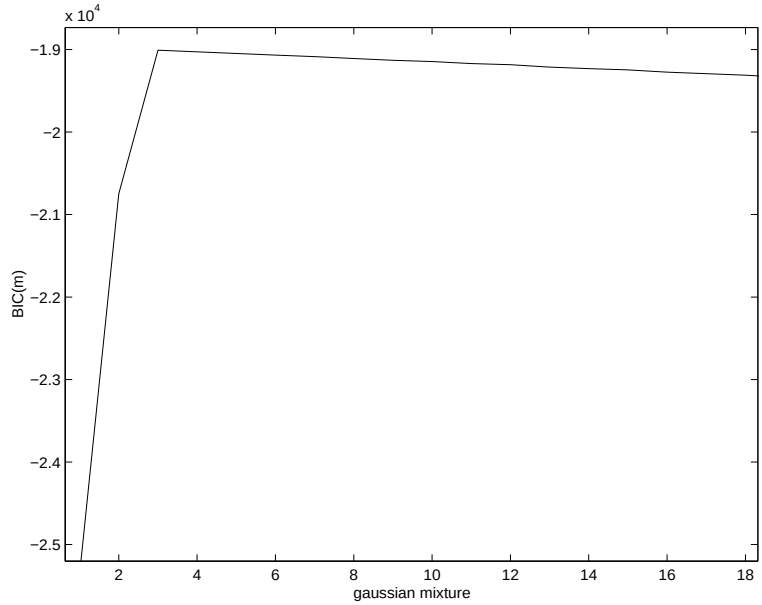

Figure 3.1: BIC(m) function of gaussian mixtures

Then we tried the variational bayesian technique described in citeattias

(VBGMM1) where we used the negative free energy to determine the best model.

We noticed that results of the VBGMM1 strongly depends on the choice of the covariance matrix priors. In fact the value of Φ_0 determines the gaussians components that dominates over the others. There's a simple way to see this problem: in classical EM when very few vectors are attributed to a gaussian component, the covariance matrix relative to this component becomes more and more singular and in the limit case of just one vector for component it becomes a zero matrix. In VBGMM, because of priors, it cannot happen that the covariance matrix becomes singular and when just one vector is attributed to a gaussian component, the weight relative to this component go to zero. This prevents from overfitting.

On the other side, the value of Φ_0 is the main cause of another phenomenon typical of variational bayesian learning: the self-pruning i.e. degree of freedom that are not used are pruned out. For this reason changing the value of Φ_0 can determine the number of gaussian components that will be pruned.

To corroborate those ideas we run experiments with different ρ values where $\Phi = \rho\, d\, I$ and I is a unit matrix, d is the vector dimension.

Figure 3.2 shows the plot of the negative free energy function of the gaussian mixtures with $\rho = 1$.

Figure 3.3 shows the plot of the non-negative components function of the gaussian mixtures with $\rho = 1$.

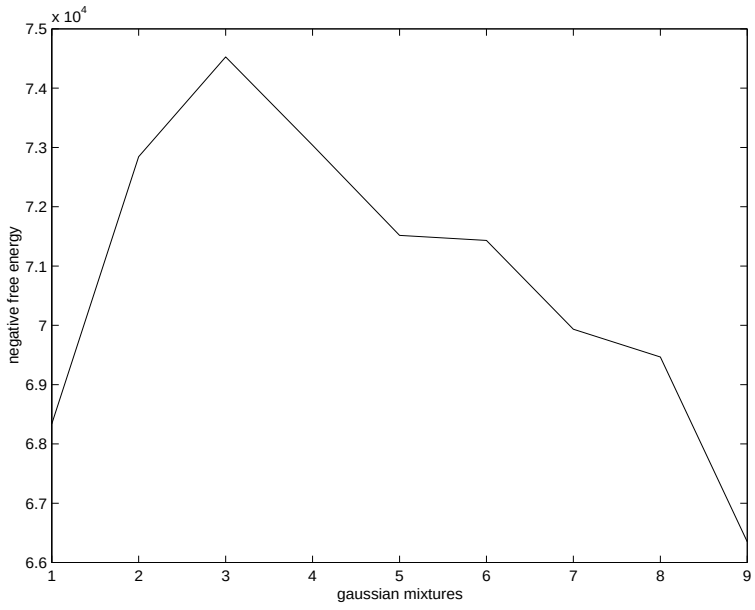

Figure 3.2: Negative free energy function of gaussian mixtures for $\rho = 1$ (VBGMM1)

From figure 3.2, the best model is the model with 3 gaussians that corresponds to model used to generate data. Figure 3.3 shows that all different gaussian components have non-zero coefficients at the end of the VBGMM training.

Now consider the same experiment with $\rho = 1000$.

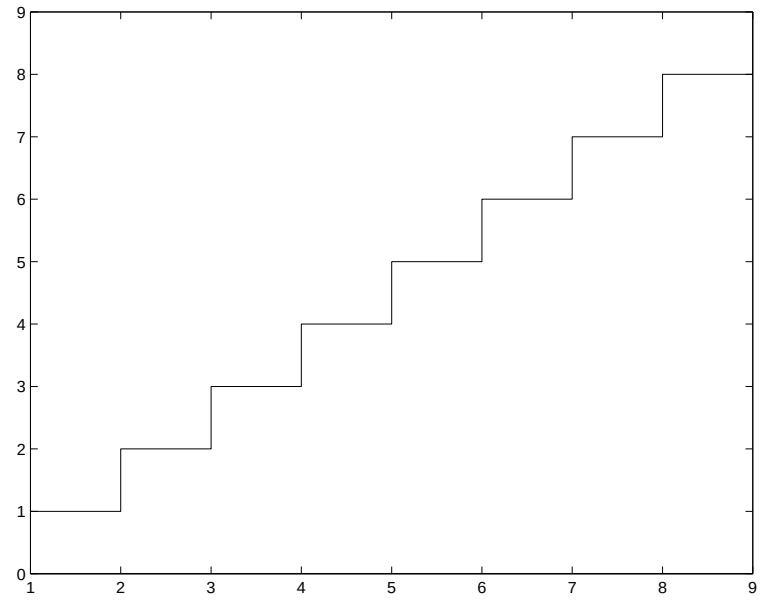

Figure 3.3: Non-zero components function of gaussian mixtures for $\rho = 1$ (VBGMM1)

Figure 3.4 shows the negative free energy: its value increases progressively when the components number goes from 1 to 3 and then is constant. Looking at the number of non zero coefficients in figure 3.5, we can notice that even when the initial number of gaussian components is bigger then 3, the model prunes freedom degree that are considered useless recovering the original number of components. For this reason all models with more than 3 gaussians have the same negative free energy i.e. they have the same number of non zero gaussian.

Now let's compare the original values with the recovered one: original weights are 0.2 0.3 and 0.5; recovered weights are 0.204 0.506 and 0.29; original means are [2,2] [-2,-2] and [5.5 5]; recovered means are [2.01,2.02] [-1.96,-1.99] [5.5 4.9]; original diagonal covariance matrix are [1 0.5] [1 1] [1 1]; recovered diagonal covariance matrix are [0.99 0.50] [1.07 0.98] [1.03 0.96].

Increasing ρ to the value 1000 data are clustered in two clusters. For this reason the choice of the ρ value is an important issue depending on the application.

Now let's consider the other approach we described as VBGMM2. The philosophy of this approach is slightly different. In fact the idea is to initialize the model with a high number of gaussian and let the model pruning itself. Again priors on the covariance matrix (here V) play a fundamental role in the gaussian number that will survive at the end of the training. Figure 3.6 plots gaussian number at the end of the training session function of covariance matrix priors.

We can notice that using priors for V between 1 and 100 permits the correct evaluation of the original gaussian components while lower values allows more components to survive after the pruning.

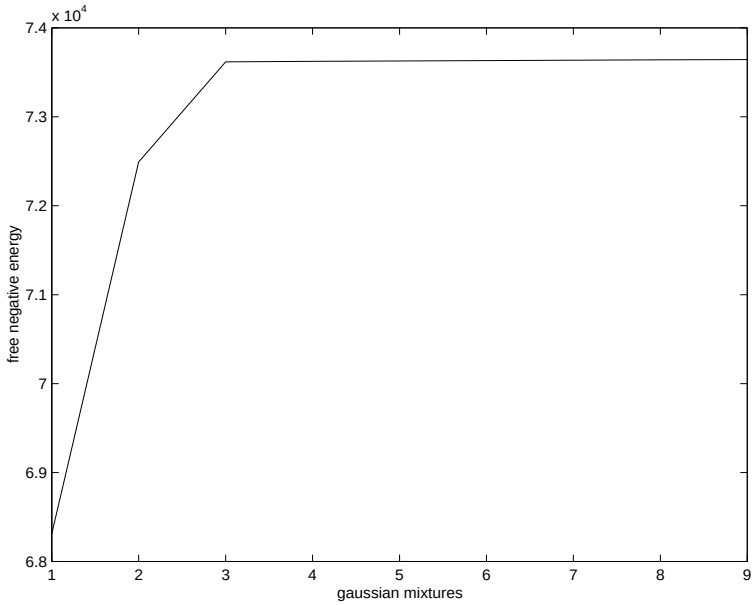

Figure 3.4: Negative free energy function of gaussian mixtures for $\rho = 100$ (VBGMM1)

3.2 Conclusion

Experiments show that Variational Bayesian technique can help to fit model to data better than classical techniques. Anyway we must notice that those approach are very sensible to prior choice (even if we try to use non-informative priors): changing priors final result seems to change considerably. Furthermore, as it was notice in [4], the model self pruning has no theorical evidence: we don't have a theorical reason for saying that the model self pruning works correctly but practical evidence.

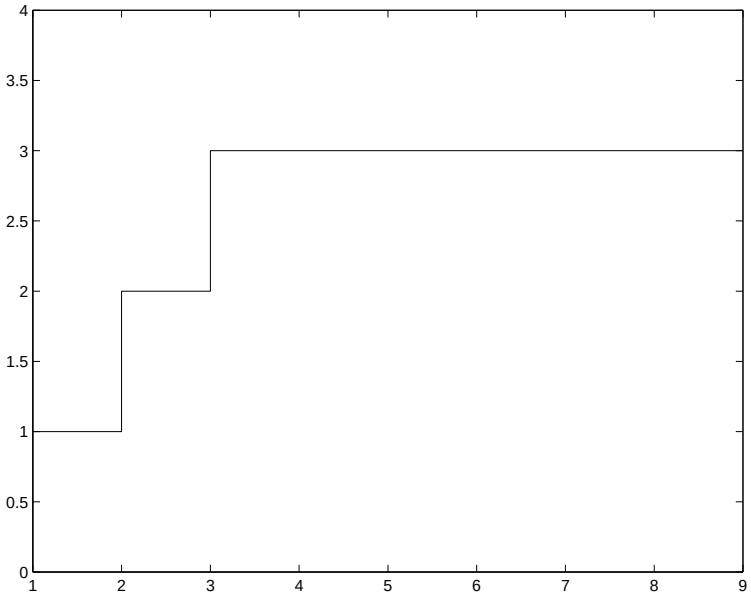

Figure 3.5: Non-zero components function of gaussian mixtures for $\rho = 100$ (VBGMM1)

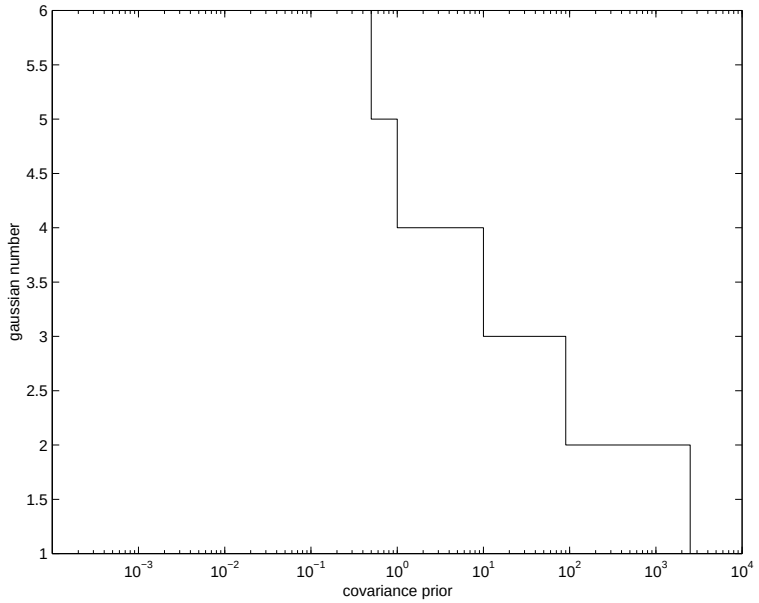

Figure 3.6: gaussian number function of covariance prior (VBGMM2)

Chapter 4

Variational bayesian HMM

It's also possible to train an HMM using variational bayesian learning. The first formulation of the problem come from McKay in [5] who described the training of a discrete density HMM using Dirichlet distributions as priors. In next section we give details about this solution.

4.1 Discrete density HMM

Let's consider the following notations:

- $S = \{s_1, s_2, ..., s_T\}$: hidden state sequence. $(s \epsilon 1...M\ )$
- $X = \{x_1, x_s, ...x_T\}$: observed sequence $(x \epsilon 1...M)$
- $A = \{a_{ij}\}$, $a_{ij} = P(s_{t+1} = j | s_t = i)$: state transition probability matrix.
- $B = \{b_{im}\}$, $P(x_t = m | s_t = i)$ emission probabilities
- $\pi = \{\pi_i\}$, $\pi_i = P(s_1 = i)$ initial state distribution
- $\theta = \{A, B, \pi\}$: model's parameters.
- $U = \{u^{(A)}, u^{(B)}, u^{(\pi)}\}$: hyperparameters that define prior over θ

Given parameters θ, it's possible to write:

$$P(X, S|\theta) = [\prod_{t=1}^{T-1} a_{s_t s_{t+1}}][\prod_{t=1}^{T} b_{s_t x_t}]\pi_{s1} \tag{4.1}$$

and for the posterior probability of the hidden variables S, it's possible to write:

$$P(S|X, \Theta) = \frac{1}{P(X|\Theta)}[\prod_{t=1}^{T-1} a_{s_t s_{t+1}}][\prod_{t=1}^{T} b_{s_t x_t}]\pi_{s1} \tag{4.2}$$

where $P(X|\Theta)$ is the normalization constant. McKay assumes prior probabilities over $\{A, B, \pi\}$ like product of Dirichlet distribution i.e.

$$P(A|u^{(A)}) = \prod_i Dirichlet(\{a_{i1}...a_{iI}\}; u^{(A)}) \tag{4.3}$$

The ensemble learning consists in approximating the distribution of hidden variables S and parameters Θ by an ensemble distribution $Q(S,\Theta)$. We assume that this distribution can be separable such that:

$$Q(S,\Theta) = Q_S(S)Q_A(A)Q_B(B)Q_\pi(\pi) \tag{4.4}$$

The optimization task consist in minimizing the free energy $F(Q(S,\Theta))$:

$$F(Q) = -\int_A \int_B \int_\pi \sum_S Q(S,\Theta) log[\frac{P(X,S,\Theta)|U}{Q(S,\Theta)}] \tag{4.5}$$

The strategy proposed by McKay consists in iteratively optimize each $Q_.$ while keeping others constant.

The log-probability can be written:

$$\begin{aligned} log\, P(X,S,\Theta|U) = & \sum_{i,j\neq 1}^{I} (u_j^{(A)} - 1) log\, a_{ij} + \sum_{i=1}^{I}\sum_{m=1}^{M} (u_m^B - 1) log\, b_{im} \\ + \sum_{i=1}^{I} (u_i^{(\pi)} - 1) log\, \pi_i + & \sum_{t=1}^{T-1} log\, a_{s_t s_{t+1}} + \sum_{t=1}^{T} log\, b_{s_t x_t} + log\, \pi_{s1} + const. \end{aligned} \tag{4.6}$$

Now let's consider the optimization of 4.6.

4.1.1 Optimization of Q_A

Let's assume that distributions Q_B, Q_π, Q_S are fixed, and let consider the free energy as a functional of Q_A:

$$\begin{aligned} F(Q_A) = & -\int_A Q_A(A)[-\sum_{i,j\neq 1}^{I} (u_j^{(A)} - 1) log\, a_{ij} + \sum_S Q_S(S) \sum_{t=1}^{T-1} log\, a_{s_t s_{t+1}} \\ & - log\, Q_A(A)] + const. \end{aligned} \tag{4.7}$$

taking inspiration from the Baum-Welch algorithm, it's possible to write:

$$w_{ij}^{(t)} = \sum_S (S)\delta(s_t = i, s_{t+1} = j), \tag{4.8}$$

and so, it's possible to write 4.7:

$$F(Q_A) = \int_A Q_A\, log\, [\frac{Q_A(A)}{\prod_{i,j} a_{ij}^{\lceil} W_{ij} - 1]}] + const \tag{4.9}$$

where

$$W_{ij} = \sum_{t=1}^{T-1} w_{ij}^{(t)} + u_j^{(A)} \tag{4.10}$$

Using Gibbs's inequality, the expression $\int_x Q(x) log \frac{Q(x)}{P(x)}$ is minimized with respect to $Q(x)$ by $Q(x) = \frac{1}{Z}$ where Z is the normalizing constant.

So the distribution Q_A that minimize 4.9 is given by a product of Dirichlet distributions:

$$Q_A(A) = \prod_i Dirichlet(a_{ij}{}_{j=1}^{I}; W_{ij}{}_{j=1}^{I}) \tag{4.11}$$

4.1.2 Optimization of Q_B,Q_π

Applying the same procedure we can obtain analogous formula for Q_B, Q_π with

$$w_{im}^{(t)} = \sum_S (S)\delta(s_t = i, x_t = m) \tag{4.12}$$

$$w_i^{\pi} = \sum_S Q_S(S)\delta(s_1 = i) \tag{4.13}$$

4.1.3 Optimization of $Q_S(S)$

Now let's fix $\{Q_A, Q_B, Q_\pi\}$ and let's consider the free energy just function of the hidden variables S.

$$\begin{aligned} F(Q_S(S)) = & -\sum_S Q_S(S)[\int_A Q_A(A) \sum_{t=1}^{T-1} log\, a_{s_t s_{t+1}} + \int_B Q_B(B) \sum_{t=1}^{T} log\, b_{s_t x_t} \\ & + \int_\pi Q_\pi(\pi) log\, \pi_{\pi_{s_1}} - log\, Q_S(S)] + const \end{aligned} \tag{4.14}$$

Let's define

$$a_{ij}^{\star} = exp[\int_A Q_A(A) log\, a_{ij}] \tag{4.15}$$

$$b_{ik}^{\star} = exp[\int_B Q_B(B) log\, b_{ik}] \tag{4.16}$$

$$\pi_i^{\star} = exp[\int_\pi Q_{\pi_i}(\pi_i)] \tag{4.17}$$

and let's rewrite 4.14 as

$$F(Q_S(S)) = \sum_S Q_S(S) log\, [\frac{Q_S(S)}{[\prod_{t=1}^{T-1} a_{s_t s_{t+1}}^{\star}][\prod_{t=1}^{T} b_{s_t x_t}^{\star}]\, \pi_{s1}^{\star}}] + const. \tag{4.18}$$

The optimal $Q_S(S)$ distribution is given by:

$$Q_S(S) = \frac{1}{Z_S}[\prod_{t=1}^{T-1} a_{s_t s_{t+1}}^{\star}][\prod_{t=1}^{T} b_{s_t x_t}^{\star}]\, \pi_{s1}^{\star} \tag{4.19}$$

Equation 4.19 has the same form of 4.2. So it's possible to compute the relevant properties of $Q_S(S)$ distribution i.e. quantities w using a forward-backward algorithm using $a^\star, b^\star, \pi^\star$. To compute those values, the following formula can be used:

$$\int_p Dirichlet(p; u) log\, p_i = \psi(u_i) - \psi(u) \tag{4.20}$$

where

$$\psi(x) = \frac{\delta}{\delta x} log\, \Gamma(x) \quad u = \sum_j u_j \tag{4.21}$$

4.1.4 Another point of view

It's interesting to notice that the same update formula can be obtained using the EM-like algorithm described in section 1.3. In fact applying formula 1.16 we have:

$$\begin{aligned} q(S) &\propto e^{<log\, p(S,X|\Theta)>_\Theta} \\ &= e^{\int_A Q_A(A) \sum_{t=1}^{T-1} log\, a_{s_t s_{t+1}} + \int_B Q_B(B) \sum_{t=1}^{T} log\, b_{s_t x_t} + \int_\pi Q_\pi(\pi) log\, \pi_{s_1}} \end{aligned} \tag{4.22}$$

and defining $a^\star, b^\star, \pi^\star$ as in the previous section we can write

$$q(S) \propto [\prod_{t=1}^{T-1} a^\star_{s_t s_{t+1}}][\prod_{t=1}^{T} b^\star_{s_t x_t}]\, \pi^\star_{s1} \tag{4.23}$$

For distribution parameters, let's optimize its posterior distribution $Q(A), Q(B), Q(\pi)$. Applying formula 1.17 we have:

$$Q(A)Q(B)Q(\pi) \propto e^{<log\, p(S,X|\Theta)>_S} p(A)p(B)p(\pi) \tag{4.24}$$

Knowing that Dirichlet distribution has conjugate priors i.e. the posterior probability has the same form of the prior probability, $Q(A), Q(B), Q(\pi)$ will be Dirichlet distributions.

$$Q(A)Q(B)Q(\pi) \propto e^{\sum_S Q_{(S)}[\sum_t log\, a_{s_t s_{t+1}} + \sum_t log\, b_{s_t x_t} + log \pi_{s1}]} p(A)p(B)p(\pi) \tag{4.25}$$

Defining $w_{ij}^{(t)}, w_{im}^{(t)}, w_i^\pi$ and W_{ij}, W_{im}, W_ias in the previous section we obtain:

$$Q(A)Q(B)Q(\pi) \propto \prod_{ij} a_{ij}^{[\sum w_{ij}]} \prod_{im} b_{im}^{[\sum w_{im}-1]} \prod_i \pi^{[\sum w_i - 1]}\; p(A)p(B)p(\pi) \tag{4.26}$$

It's easy to obtain formulas for $Q(A), Q(B), Q(\pi)$:

$$Q_A(A) = \prod_i Dirichlet(a, W_{ij}) \tag{4.27}$$

$$Q_B(B) = \prod_i Dirichlet(b, W_{im}) \tag{4.28}$$

$$Q_\pi(\pi) = \prod_i Dirichlet(\pi, W_i) \tag{4.29}$$

4.2 Variational bayesian HMM with GMM

In this section we extend the variational bayesian HMM framework to continuous density emission probabilities using variational bayesian GMM described in previous sections.

Let's keep the same notations of previous section for HMM parameters but for the emission probabilities. In fact now we will consider continuous emission probabilities modeled by a GMM.

Let's assume that $b_i = \sum_m c_{im} N(\mu_{im}, \Sigma_{im})$ is the pdf of emission in state i. In the variational bayesian framework let's define priors on this quantities as:

$$p(\{c_{im}\}) = D(\lambda^0) \tag{4.30}$$

$$p(\mu_{im}|\Sigma_{im}) = N(\rho^0, \beta^0 \Sigma_{im}) \tag{4.31}$$

$$p(\Sigma_{im}) = W(\nu^0, \Phi^0) \tag{4.32}$$

In GMM/HMM there are two different hidden variables sequences: the hidden state sequence S and the hidden emitting gaussians M and we will approximate their probabilities with a distribution $Q(S, M)$. It's possible to write:

$$P(X, S, M|\Theta) = [\prod_{t=1}^{T-1} a_{s_t s_{t+1}}][\prod_{t=1}^{T} b_{s_t m_t x_t}]\pi_{s1} \tag{4.33}$$

where

$$b_{s_t m_t x_t} = c_{s_t m_t} N(\mu_{s_t m}, \Sigma_{s_t m}) \tag{4.34}$$

is the probability of the emission of the m-th gaussian in the s-th state. and now let's try to apply the EM-like algorithm. In the M step we should compute:

$$q(S, M) \propto e^{<log\, p(S,M,X|\Theta)>_\Theta} \tag{4.35}$$

where

$$\begin{aligned} < log\, p(S, M, X|\Theta) >_\Theta >= & \int_A Q_A \sum_{t=1}^{T-1} log\, a_{s_t s_{t+1}} + \int_{Q_C} \int_{Q_\mu} \int_{Q_\Sigma} \sum_{t=1}^{T} log\, b_{s_t m_t} + \int_\pi Q_\pi log\, \pi_{s1} \\ = & \sum_{t=1}^{T-1} \int_A Q_A log\, a_{s_t s_{t+1}} + \sum_{t=1}^{T} \int_{Q_C} \int_{Q_\mu} \int_{Q_\Sigma} log\, b_{s_t m_t} + \sum_{t=1}^{T} \int_\pi Q_\pi log\, \pi_{s1} \end{aligned} \tag{4.36}$$

Defining now

$$a^\star_{ij} = exp[\int_A Q_A log\, a_{ij}] \tag{4.37}$$

$$b^\star_{imx} = exp[\int_{Q_C} \int_{Q_\mu} \int_{Q_\Sigma} log\, b_{imx}] \tag{4.38}$$

$$\pi^\star_i = exp[\int_\pi Q_\pi log\, \pi_i] \tag{4.39}$$

we can rewrite 4.35 as:

$$Q(S,M) \propto [\prod_{t=1}^{T-1} a^{\star}_{s_t s_{t+1}}][\prod_{t=1}^{T} b^{\star}_{s_t m_t x_t}]\, \pi^{\star}_{s1} \tag{4.40}$$

Computation of $a^{\star}_{ij}$ and $\pi^{\star}_{i}$ can be done using formula 4.20, while computation of $b^{\star}_{imx}$ can be done using results from 2.10.

To obtain the probability $Q(S)$, it's enough to marginalize $Q(S,M)$ with respect to M obtaining:

$$\begin{aligned} Q(S) = \sum_{M} Q(S,M) &\propto \sum_{M} [\prod_{t=1}^{T-1} a^{\star}_{s_t s_{t+1}}][\prod_{t=1}^{T} b^{\star}_{s_t m_t x_t}]\, \pi^{\star}_{s1} \\ &= [\prod_{t=1}^{T-1} a^{\star}_{s_t s_{t+1}}][\prod_{t=1}^{T} \sum_{m_t} b^{\star}_{s_t m_t x_t}]\, \pi^{\star}_{s1} \end{aligned} \tag{4.41}$$

It's interesting to notice that $Q(S)$ is not the distribution we obtain marginalizing $P(S,M,X|\Theta)$ and then computing the posterior $Q(S)$. In fact someone may suppose that $Q(S)$ could be obtained computing $exp(< log \sum_M P(S,M,X|\Theta) >$) because $P(S) = \sum_M P(S,M)$. The problem in doing it is that $Q(S)$ and $Q(S,M)$ are not exact probabilities but just approximating function, and so Q(S) is not the probability density that comes from the marginalization of P(S,M). In other words it's different to integrate the logarithm of a sum than integrating the sum of a logarithm. Important statistics of the $Q(S,M)$ distribution can be found using the forward-backward algorithm or the Viterbi algorithm.

Now let's consider the M-step: updating formula for transition probabilities and initial state probabilities are almost the same because $Q(S,M)$ is marginalized w.r.t. M while update formulas for gaussian mixtures differs. We can write the M-step like:

$$q(A)q(\pi)q(\mu)q(\Sigma)q(c) = e^{<logP(S,M,X|\Theta)>_{Q(S,M)}} p(A)p(\pi)p(\mu)p(\Sigma)p(c) \tag{4.42}$$

where

$$\begin{aligned} < logP(S,M,X|\Theta) >_{Q(S,M)} &= \sum_{S,M} Q(S,M)[\sum_{t} log\, a_{s_t s_{t+1}} + \sum_{t} log\, b_{s_t m_t x_t} + log\, \pi_1] \\ &= \sum_{S} Q(S) \sum_{t} log\, a_{s_t s_{t+1}} + \sum_{S,M} Q(S,M) \sum_{t} log\, b_{s_t m_t x_t} + \sum_{S} Q(S) log\, \pi_1 \end{aligned} \tag{4.43}$$

Update formulas for a_{ij} and π are analogous to updating formula in the previous section with the only difference that $Q(S)$ comes from the marginalization of $Q(S,M)$.

Let's consider now updating formula just for gaussians parameters: the problem is analogous to simple guassian mixtures with the difference that in here we have to consider the state probability togheter with the component probability. We can write:

$$\sum_{S,M} Q(S,M) \sum_t log\, b_{s_t m_t x_t} = \sum_t \sum_{s_t,m_t} Q(S)\delta(s_t,m_t) log\, b_{s_t m_t x_t} = \sum_t Q(s_t,m_t) log\, b_{s_t m_t x_t} \tag{4.44}$$

where Q_{s_t,m_t} is the approximated probability for being in state s at time t and for emission of gaussian m. This time we can write $Q(s_t,m_t) = Q(s_t)Q(m_t|s_t) = Q(s_t)\gamma_m / \sum \gamma_m$. This is the same expression computed in 2.3 using $Q(s_t,m_t)$ instead of simply $Q(m)$. It means that update formulas for gaussian parameters can be obtains as in section 2.3 using $N_{s,m} = \sum_{s,m} Q(s_t,m_t)$ instead of $N_s = \sum \gamma_s$.

Bibliography

[1] Attias H. A variational bayesian framework for graphical models. *Advances in Neural Information Processing Systems*, 12, 2000.

[2] Winn J. Bishop C.M. Structured variational distribution in vibes. *Proceedings Artificial Intelligence and Statistics,Key West, Florida*, 2003.

[3] Bishop C.M. Corduneau A. Variational bayesian model selection for mixture distributions. *In T. Richardson and T. Jaakkola (Eds.), Proc. 8th Int. Conf. on Artificial Intelligence and Statistics, Morgan Kaufmann.*, pages 27–34.

[4] MacKay D.J.C. Local minima, symmetry breaking and model pruning in variational free energy minimization.

[5] MacKay D.J.C. Ensemble learning for hidden markov models.