Variational Bayesian Feature Selection

Research Report RR-03-087

Fabio Valente, Christian Wellekens

October 1, 2003

Contents

Chapter 1

Features Selection

1.1 Introduction

In a pattern recognition task, a central role is played by feature selection. Selecting most important features has a double goal: reducing computational charges and improving recognition rate (eliminating the most noisy features).

Feature selection algorithms belongs to two big families: filters and wrappers that need class labels. In many applications, class labels are not available (e.g. unlabeled speech recognition database) and other kind or algorithms must be used.

In this report we first consider feature saliency model proposed in [1] and [2] for mixture based clustering and then propose a bayesian framework base on variational learning that enables parameter learning and model learning.

Chapter 2

Mixture Model with Feature Selection

In this chapter we consider the model proposed in [1]. This model permits to learn simultaneously parameters and features saliency in a mixture model.

2.1 Mixture Model

Let's consider the following Gaussian Mixture Model:

$$p(y) = \sum_{j=1}^{K} \alpha_j p(y|\theta_j) = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} p(y_l|\theta_{jl}) \tag{2.1}$$

where the factorization is possible if we assume diagonal covariance matrix. The model consists of K different components, each component is multivariate gaussian, vector dimension is D, θ_{jl} is the model of the $l-th$ vector component in the $j-th$ mixture component.

Feature selection problem consists in determining how relevant is the $l-th$ component. The relevance is considered as the capacity of discriminate between different mixture components; assuming $q(y_l|\lambda_l)$ the distribution of the $l-th$ feature regardless the mixture component it belongs to. If the $l-th$ feature is irrelevant, it's intuitively reasonable to assume $p(y_l|\theta_{jl}) = q(y_l|\theta_l)$ i.e. the feature distribution in the $j-th$ mixture component is identical to the total distribution. Introducing a binary variable ϕ_l that indicates if the feature is relevant ($\phi_l = 1$) or not ($\phi_l = 0$), it's possible to write the model like:

$$p(y) = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} (p(y_l|\theta_{jl}))^{\phi_l} (q(y_l|\lambda_l))^{1-\phi_l} \tag{2.2}$$

Obviously in real data problems instead of determining the value of ϕ as a binary variable, it's more interesting to determine the "saliency" of a certain feature. For this reason it's possible to consider ϕ as an hidden variable and define $\rho_l = P(\phi_l = 1)$. It can be demonstrated that using this assumption the

model can be written as:

$$p(y) = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} (\rho_l p(y_l|\theta_{jl}) + (1-\rho_l) q(y_l|\lambda_l)) \tag{2.3}$$

This formulation can be seen as a Hierarchical Mixture Model with two hidden variables that indicates the mixture components and the feature saliency.

2.2 Learning Parameters

Given an unlabeled training set $Y = (y_1, ..., y_N)$ with $y_i = (y_{i1}, ..., y_{iD})$, the problem is estimation of parameters of model 2.3 given by $\Theta = \{\alpha_j, \theta_{jl}, \lambda_l, \rho_l\}$. In [1] parameters learning is based on a Maximum Likelihood criterion:

$$\hat{\Theta} = argmax\ log\, p(Y|\Theta) = argmax \sum_{i=1}^{N} log \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} (\rho_l p(y_l|\theta_{jl}) + (1-\rho_l) q(y_l|\lambda_l)) \tag{2.4}$$

Because of the fact that the model uses hidden variables an Expectation-Maximization algorithm must be used.

The E-step consists consists in the following estimations:

$$a_{ijl} = P(\phi_l = 1, y_{il}|z_i = j) = \rho_l p(y_{il}|\theta_{jl}) \tag{2.5}$$

$$b_{ijl} = P(\phi_l = 0, y_{il}|z_i = j) = (1-\rho_l) q(y_{il}|\lambda_l) \tag{2.6}$$

$$c_{ijl} = P(y_{il}|z_i = j) = a_{ijl} + b_{ijl} \tag{2.7}$$

$$w_{ij} = P(z_i = j|y_i) = \frac{\alpha_j \prod_l c_{ijl}}{\sum_j \alpha_j \prod_l c_{ijl}} \tag{2.8}$$

$$u_{ijl} = P(\phi_l = 1, z_i = j|y_i) = \frac{a_{ijl}}{c_{ijl}} w_{ij} \tag{2.9}$$

$$v_{ijl} = P(\phi_l = 0, z_i = j|y_i) = w_{ij} - u_{ijl} \tag{2.10}$$

$$\sum_j v_{ijl} = P(\phi_l = 0|y_i) = 1 - \sum_j u_{ijl} \tag{2.11}$$

The M-step consists in the following parameters estimations:

$$\hat{\alpha_j} = \frac{\sum_i w_{ij}}{\sum_{ij} w_{ij}} \tag{2.12}$$

$$\mu(\hat{\theta}_{jl}) = \frac{\sum_i u_{ijl} y_{il}}{\sum_i u_{ijl}} \tag{2.13}$$

$$\sigma(\hat{\theta}_{jl}) = \frac{\sum_i u_{ijl} (y_{il} - \mu(\hat{\theta}_{jl}))^2}{\sum_i u_{ijl}} \tag{2.14}$$

$$\mu(\hat{\lambda}_l) = \frac{\sum_i (\sum_j v_{ijl}) y_{il}}{\sum_{ij} v_{ijl}} \tag{2.15}$$

$$\sigma(\hat{\lambda}_l) = \frac{\sum_i (\sum_j v_{ijl}) (y_{il} - \mu(\hat{\lambda}_l))^2}{\sum_{ij} v_{ijl}} \tag{2.16}$$

$$\hat{\rho_l} = \frac{\sum_{ij} u_{ijl}}{\sum_{ij} u_{ijl} + \sum_{ij} v_{ijl}} = \frac{\sum_{ij} u_{ijl}}{n} \tag{2.17}$$

2.3 Model Selection

A serious problem in using GMM is that the number of mixture must be given. If it does not match the "real" cluster number, parameter estimation may suffer of many problems like overfitting or local maxima problems. A possible model selection criterion is the Minimum Message Length (MML)

It's possible to obtain a criterion for joint parameters estimation and model selection. In [1] it's obtained the following criterion:

$$\hat{\Theta} = argmin\{-log\, p(Y|\Theta) + \frac{1}{2}(K + D + KDR + DS) log\, n + \frac{R}{2}\sum_{j=1}^{K}\sum_{l=1}^{D} log(\alpha_j \rho_l) + \frac{S}{2}\sum_{l=1}^{D} log\,(1-\rho_l)\} \tag{2.18}$$

where R and S are number of parameters in θ_{jl} and λ_l. In the proposed model $R = S = 2$. Criterion 2.18 can be rewritten as:

$$\hat{\Theta} = argmax\{log\, p(Y|\Theta) - \frac{RD}{2}\sum_{l=1}^{K} log\,\alpha_j - \frac{S}{2}\sum_{l=1}^{D} log\,(1-\rho_l) - \frac{RK}{2}\sum_{l=1}^{D} log\,\rho_l\} \tag{2.19}$$

which can be seen as a MAP estimate defining the following improper priors on α_j and ρ_l:

$$p(\alpha_1, ..., \alpha_K) \propto \prod_{l=1}^{K} \alpha_j^{-RD/2} \tag{2.20}$$

$$p(\rho_l) \propto \rho_l^{-RK/2}(1-\rho_l)^{-S/2} \tag{2.21}$$

In the EM algorithm the update formula for α_j and ρ_l becomes:

$$\hat{\alpha_j} = \frac{max(\sum_i w_{ij} - RD/2, 0)}{\sum_j max(\sum_i w_{ij} - RD/2, 0)} \tag{2.22}$$

$$\hat{\rho_l} = \frac{max(\sum_{ij} u_{ijl} - KR/2, 0)}{max(\sum_{ij} u_{ijl} - KR/2, 0) + max(\sum_{ij} v_{ijl} - S/2, 0)} \tag{2.23}$$

2.3.1 Pruning Model

An important point in using update formula 2.22 and 2.23 is the possibility of pruning parameters. In fact if the initial model is initialized with a huge number of gaussians, MML learning should detect the correct number of clusters pruning extra gaussians.

Naturally when α_j or ρ_l go to zero, parameters number changes because parameters relative to components j or feature l can be discarded; it means that criterion 2.18 must be modified in order to consider the right number of parameters.

$$\hat{\Theta} = argmin\{-log\, p(Y|\Theta) + \frac{1}{2}(K^{'} + D_3 + K^{'} D_1 R + D_2 S) log\, n + R/2 \sum_{j=1}^{K^{'}}\sum_{l=1}^{D_1} log(\alpha_j \rho_l) + S/2 \sum_{l=1}^{D_2} log\,(1-\rho_l)\} \tag{2.24}$$

where $K^{'}$ denotes the current number of mixture components, D_1 are features for which $\rho_l > 0$, D_2 are features for which $\rho_l < 1$, and D_3 are features for which $0 < \rho_l < 1$.

It's now possible to modify equations 2.22 and 2.23 as follows:

$$\hat{\alpha_j} = \frac{max(\sum_i w_{ij} - RD_1/2, 0)}{\sum_j max(\sum_i w_{ij} - RD_1/2, 0)} \tag{2.25}$$

$$\hat{\rho_l} = \frac{max(\sum_{ij} u_{ijl} - K^{'}R/2, 0)}{max(\sum_{ij} u_{ijl} - K^{'}R/2, 0) + max(\sum_{ij} v_{ijl} - S/2, 0)} \tag{2.26}$$

2.3.2 Component-wise EM

A problem outlined in [1] is the very hard pruning behavior in first EM step; in fact if the initial component number is too big the penalty term can be too hard and *all* components are pruned out. To avoid this behavior, authors propose a modification to the EM algorithm: mixture components are updated in turns instead of in parallel. They refer to this modified EM algorithm as Component Wise EM (CWEM) (see [1] for details).

Chapter 3

Variational Bayesian Feature Selection

Another efficient way for doing model selection consists in using *Variational Bayesian* (VB) techniques. First application to gaussian mixture models was proposed in [3].

Using VB learning offers a way for optimizing jointly parameters and model; furthermore another key feature of VB learning is that it naturally prunes extra degree of freedom. For an exhaustive review see [4]

3.1 Variational Bayesian Learning

Let's consider the model 2.3:

$$p(y|\Theta) = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} (\rho_{l0} p(y_l|\theta_{jl}) + \rho_{l1} q(y_l|\lambda_l)) \tag{3.1}$$

where $\Theta = \{\alpha_j, \theta_{jl}, \lambda_l, \rho_l\}$.

Let's explicit gaussian distributions:

$$p(y_l|\theta_{jl}) = \frac{1}{\sqrt{2\pi}\sigma_{jl}} exp(-\frac{(y_l - \mu_{jl})^2}{2\sigma_{jl}^2}) \tag{3.2}$$

$$q(y_l|\lambda_l) = \frac{1}{\sqrt{2\pi}\sigma_l} exp(-\frac{(y_l - \mu_l)^2}{2\sigma_l^2}) \tag{3.3}$$

and their prior distributions:

$$p(\Theta) = p(\alpha)p(\rho_l) \prod_{jl} p(\sigma_{jl})p(\mu_{jl}|\sigma_{jl})p(\sigma_l)p(\mu_l|\sigma_l) \tag{3.4}$$

where

$$p(\alpha) = Dir(\lambda_0) \tag{3.5}$$

$$p(\rho) = Dir(\tau_0) \tag{3.6}$$

$$p(\sigma_{jl}) = \Gamma(b_0, c_0) \tag{3.7}$$
$$p(\mu_{jl}|\sigma_{jl}) = N(\mu|m_0, \beta_0\sigma_{jl}) \tag{3.8}$$
$$p(\sigma_l) = \Gamma(b_0, c_0) \tag{3.9}$$
$$p(\mu_l|\sigma_l) = N(\mu|m_0, \beta_0\sigma_l) \tag{3.10}$$

where Γ is a gamma distribution and N is a normal distribution.
To learn optimal variational bayesian posterior EM algorithm can be applied.

3.1.1 E-step

Following formula implements the E-step:

$$a_{ijl} = \tilde{\rho}_{l0}\bar{\sigma}_{jl}^{1/2} exp(-\frac{1}{2}(y-\mu_{jl})^T\bar{\sigma}_{jl}(y-\mu_{jl}))exp(-\frac{1}{2\beta_{jl}}) \tag{3.11}$$
$$b_{ijl} = \tilde{\rho}_{l1}\bar{\sigma}_{l}^{1/2} exp(-\frac{1}{2}(y-\mu_{l})^T\bar{\sigma}_{l}(y-\mu_{l}))exp(-\frac{1}{2\beta_{l}}) \tag{3.12}$$

where

$$log\ \tilde{\rho}_{l0} = \Psi(\tau_0) - \Psi(\tau_0+\tau_1) \tag{3.13}$$
$$log\ \tilde{\rho}_{l1} = \Psi(\tau_1) - \Psi(\tau_0+\tau_1) \tag{3.14}$$
$$log\ \bar{\sigma}_{jl} = \Psi(b_{jl}/2) - log\ c_{jl} + log2 \tag{3.15}$$
$$log\ \bar{\sigma}_{l} = \Psi(b_{l}/2) - log\ c_{l} + log2 \tag{3.16}$$
$$\bar{\sigma}_{jl} = b_{jl}/c_{jl} \tag{3.17}$$
$$\bar{\sigma}_{l} = b_{l}/c_{l} \tag{3.18}$$

where Ψ is the digamma function.

$$c_{ijl} = a_{ijl} + b_{ijl} \tag{3.19}$$
$$w_{ij} = \frac{\tilde{\alpha}_j \prod_l c_{ijl}}{\sum_j \tilde{\alpha}_j \prod_l c_{ijl}} \tag{3.20}$$
$$u_{ijl} = \frac{a_{ijl}}{c_{ijl}} w_{ij} \tag{3.21}$$
$$v_{ijl} = w_{ij} - u_{ijl} \tag{3.22}$$
$$\sum_j v_{ijl} = 1 - \sum_j u_{ijl} \tag{3.23}$$

where

$$log\ \tilde{\alpha}_j = \Psi(\lambda_j) - \Psi(\sum_j \lambda_k) \tag{3.24}$$

3.1.2 M-step

Model parameters can be estimated using formula 2.12-2.17. Optimal parameter posterior can be estimated because prior distributions belongs to conjugate family.

Let's define

$$N_{alpha-j} = \sum_i w_{ij} \tag{3.25}$$

$$N_{rho-l0} = \sum_{ij} u_{ijl} \tag{3.26}$$

$$N_{rho-l1} = \sum_{ij} v_{ijl} \tag{3.27}$$

$$N_{jl} = \sum_i u_{ijl} \tag{3.28}$$

$$N_l = \sum_{ij} v_{ijl} \tag{3.29}$$

It's now possible to estimate posterior distributions:

$$q(\alpha) = Dir(\lambda) \tag{3.30}$$

$$q(\rho_l) = Dir(\tau_l) \tag{3.31}$$

$$\tag{3.32}$$

with

$$\lambda_j = \lambda_0 + N_{alpha-j} \tag{3.33}$$

$$\tau_{l0} = \tau_0 + N_{rho-l0} \quad \tau_{l1} = \tau_0 + N_{rho-l1} \tag{3.34}$$

$$\tag{3.35}$$

For means distributions:

$$q(\mu_{jl}|\sigma_{jl}) = N(m_{jl}|\beta_{jl}\sigma_{jl}) \tag{3.36}$$

$$q(\mu_l|\sigma_l) = N(m_l|\beta_l\sigma_l) \tag{3.37}$$

where

$$m_{jl} = \frac{N_{jl}\,\mu_{ijl} + \beta_0 m_0}{N_{jl} + \beta_0} \tag{3.38}$$

$$m_l = \frac{N_l\,\mu_l + \beta_0 m_0}{N_l + \beta_0} \tag{3.39}$$

$$\beta_{jl} = N_{jl} + \beta_0 \tag{3.40}$$

$$\beta_l = N_l + \beta_0 \tag{3.41}$$

For precision distributions:

$$q(\sigma_{jl}) = \Gamma(b_{jl}, c_{jl}) \tag{3.42}$$

$$q(\sigma_l) = \Gamma(b_l, c_l) \tag{3.43}$$

where

$$b_{jl} = N_{jl}\,\sigma_{jl} + \frac{N_{jl}\,\beta_0\,(\mu_{jl} - m_0)^2}{N_{jl} + \beta_0} + b_0 \tag{3.44}$$

$$b_l = N_l\,\sigma_l + \frac{N_l\,\beta_0\,(\mu_l - m_0)^2}{N_l + \beta_0} + b_0 \tag{3.45}$$

$$a_{jl} = N_{jl} + a_0 \tag{3.46}$$

$$a_l = N_l + a_0 \tag{3.47}$$

3.1.3 Model Selection

Variational bayesian learning offers another important possibility: Variational bound represent a criterion for model selection. In general, given an observation set Y hidden variables X, model parameters Θ variational bound can be expressed as (see [4] for details):

$$M = \int q(\Theta|Y)\, q(X|Y)\, log \frac{P(Y,X|\Theta)}{P(X|\Theta)} d\Theta dX \quad - D(q(\Theta|Y)||p(\Theta)) \tag{3.48}$$

where $p(\Theta)$ is parameters prior distribution, $q(\Theta|Y)$ and $q(X|Y)$ are respectively variational parameters posterior and variational hidden variables posterior.

Expression 3.48 can be rewritten as:

$$\begin{aligned} M \quad = \quad & \int q(\Theta|Y)q(X|Y)\, logP(Y|X,\Theta) d\Theta dX + \int q(\Theta|Y)q(X|Y)\, logP(X|\Theta) d\Theta dX + \\ - \quad & \int q(\Theta|Y)q(X|Y) log\, q(X|Y) d\Theta dX \end{aligned} \tag{3.49}$$

It's possible to compute in close form the three terms that compose 3.49.
For the first term:

$$\begin{aligned} & \int q(\Theta|Y)q(X|Y)\, logP(Y|X,\Theta) d\Theta dX = \\ & \int q(\Theta|Y) \prod_n q(X_n|Y) \sum_n log\, P(Y_n|X_n,\Theta) d\Theta dX = \\ & \sum_n \int q(\Theta|Y)q(X_n|Y)\, log\, P(Y_n|X_n,\Theta) d\Theta dX = \end{aligned} \tag{3.50}$$

Let's now write the hidden variables distribution as:

$$q(X_n|Y) = q(g_n, \phi_{n1}, ..., \phi_{nl}) = q(g_n)q(\phi_{n1}|g_n)...q(\phi_{nl}|g_n) \tag{3.51}$$

$$\begin{aligned} & \sum_n \sum_{hidden} \int q(\Theta|Y)\, q(g_n, \phi_{n1}, ..., \phi_{nl})\, log \prod_l P(Y_{nl}|X_{nl},\Theta) = \\ \sum_n \sum_{hidden} & q(g_n)q(\phi_{n1}|g_n)...q(\phi_{nl}|g_n) \sum_l \int q(\Theta|Y)\, log\, P(Y_{nl}|g_n,\phi_{nl},\Theta) = \\ & \sum_n \sum_{hidden} q(g_n)q(\phi_{n1}|g_n)...q(\phi_{nl}|g_n) \sum_l \tilde{log}\, P(Y_{nl}|g_n,\phi_{nl}) = \\ & \sum_n \sum_{gaussian} q(g_n) \sum_l \sum_{\phi_l} q(\phi_{nl}|g_n) \tilde{log}\, P(Y_{nl}|g_n,\phi_{nl}) = \\ \sum_n \sum_{gaussian} & q(g_n) \sum_l (q(\phi_{nl}=0|g_n)\, \tilde{log}\, P(Y_{nl}|g_n,\phi_{nl}=0) + \\ & + q(\phi_{nl}=1|g_n)\, \tilde{log}\, P(Y_{nl}|g_n,\phi_{nl}=1) = \\ & \sum_n \sum_{gaussian} w_{ij} \sum_l (exp(a_{ijl})\, \frac{a_{ijl}}{c_{ijl}} + exp(b_{ijl})\, \frac{b_{ijl}}{c_{ijl}}) \end{aligned}$$

For the second term of 3.49:

$$\int q(\Theta|Y)q(X|Y)\,log P(X|\Theta)d\Theta dX =$$

$$\int q(\Theta|Y)\prod_n q(X_n|Y)\,log\prod_n P(X_n|\Theta)dX\,d\Theta =$$

$$\sum_n \int q(X_n|Y)\int q(\Theta|Y)\,log\,P(X_n|\Theta)dX\,d\Theta = \sum_n \sum_{hidden} q(X_n|Y)\int q(\Theta|Y)log\,P(X_n|\Theta)d\Theta =$$

$$\sum_n \sum_{hidden} q(g_n)q(\phi_{n1}|g_n)...q(\phi_{nl}|g_n)\int q(\Theta|Y)\,log\,\{P(g_n)P(\phi_{n1}|g_n)...P(\phi_{nl}|g_n)\}d\Theta =$$

$$\sum_n \sum_{hidden} q(g_n)q(\phi_{n1}|g_n)...q(\phi_{nl}|g_n)\,\{\tilde{log}\,P(g_n) + \tilde{log}\,P(\phi_{n1}|g_n) + ... + \tilde{log}\,P(\phi_{nl}|g_n)\} =$$

$$\sum_n \sum_{gaussian} q(g_n)[\tilde{log}\,P(g_n) + \sum_l \{q(\phi_{nl}=0)\tilde{log}\,P(\phi_{nl}=0|g_n) + q(\phi_{nl}=1)\tilde{log}\,P(\phi_{nl}=1|g_n)\}] =$$

$$\sum_n \sum_{gaussian} w_{nj}[\tilde{\alpha_j} + \sum_l \{\frac{a_{jnl}}{c_{jnl}}\tilde{\rho_{0jl}} + \frac{b_{jnl}}{c_{jnl}}\tilde{\rho_{1jl}}\}]$$

For the third term of 3.49:

$$\int q(\Theta|Y)q(X|Y)log\,q(X|Y)d\Theta dX =$$

$$\int q(\Theta|Y)\prod_n q(X_n|Y)\,log\prod_n q(X_n|Y)d\Theta dX =$$

$$\int \prod_n q(X_n|Y)\sum_n log\,q(X_n|Y)dX =$$

$$\sum_n \sum_{hidden} q(g_n)q(\phi_{n1}|g_n)...q(\phi_{nl}|g_n)\,log\,[q(g_n)q(\phi_{n1}|g_n)...q(\phi_{nl}|g_n)] =$$

$$\sum_n \sum_{gaussian} q(g_n)[log\,q(g_n) + \sum_l \{P(\phi_{nl}=0|g_n)log\,P(\phi_{nl}=0|g_n) + P(\phi_{nl}=1|g_n)log\,P(\phi_{nl}=1|g_n)\}] =$$

$$\sum_n \sum_{gaussian} w_{nj}[log\,w_{nj} + \sum_l \{\frac{a_{jnl}}{c_{jnl}}log\,\frac{a_{jnl}}{c_{jnl}} + \frac{b_{jnl}}{c_{jnl}}log\,\frac{b_{jnl}}{c_{jnl}}\}]$$

Chapter 4

Supervised learning and HMM

4.1 Supervised learning

The approach we considered so far assumes that observation labels are not available. Anyway this is not the case if the training is a supervised training. In this case the model we have previously used can still be used to infer the feature saliency; if labels are provided we don't need to learn model structure because components number is known. Because of the fact we know the gaussian to which the training element belongs to, there is no need for weighting gaussian components.

Given a training set $Y = \{y_1, ..., y_n\}$ and a given function $g(.)$ that assign a class to each training element we can write likelihood like:

$$log\ P(Y|\Theta) = \sum_i log\ p(y_i|g(y_i)) \ = \sum_i log \prod_l (\rho_l p(y_{il}|\theta_{g(y_i)}) + (1-\rho_l)p(y_{il}|\lambda)) \tag{4.1}$$

where $\theta_{g(y_i)}$ indicates parameters of class $g(y_i)$, and λ indicates parameters independent by classes.

In other words the only hidden variable in this model is now feature saliency. EM formula can be simply deduced by formula in section 2.2 assuming that hidden variables z_j are actually known.

4.2 Hidden Markov Models

The most popular model in speech recognition is HMM. State pdf are generally assumed to be gaussian distributions or mixture of gaussian distributions. In this case another hidden set of variables represented by state sequence must be considered.

The feature selection framework can be used also in modeling the state pdf in the same way as previously defined. For parameter learning classical Baum-Welch algorithm can be used. If model selection must be done HMM can be learned using VB learning (see [4]).

Chapter 5

Experiments

5.1 Synthetic data

To test GMM/feature saliency methods, we generated 1000 vectors of dimension 5 using a 3 component GMM with following mean vectors:

$$\begin{aligned} mean1 &= [0, 0, 0, 0, 1] \\ mean2 &= [-1, 0, -1, -1, 1] \\ mean3 &= [1, 0, 1, -1, 1] \end{aligned}$$

and diagonal covariance matrix. GMM weighs are respectively 0.3 0.4 and 0.3.

We can notice that feature one and three can discriminate between three gaussians, feature 4 cannot discriminate between $m2$ and $m3$, and features two and five cannot discriminate at all. In an ideal experiments we should have $\phi_1 = \phi_3 = 1$,$\phi_2 = \phi_5 = 0$ and $0 \leq \phi_4 \leq 1$.

Instead of using features unable to discriminate between models, simply noisy could be added to three different gaussians. We run experiments even in noisy environment having same results.

5.1.1 Feature saliency and component number

In this experiment we run GMM with feature saliency as described in section 2.1 changing the gaussian component number from 1 to 10. Feature saliency is represented in figure 5.1.

The three discriminant features get $\phi = 1$ for all initial component values. On the other side feature saliency for features two and five shows huge variation depending on the component number. Here comes the need of finding a technique that can determine the correct number of components.

5.1.2 MML versus Variational Bayesian

MML method and VB method were compared on the same task. Concerning feature saliency both methods were successful assigning $\phi_1 = \phi_3 = \phi_4 = 1$ and $\phi_2 = \phi_5 = 0$. On the other side MML have much more difficult to identify the correct cluster number. Both techniques actually suffer from local minima problems; while using VB with different initialization we were always able to

detect the correct cluster dimension, MML often gets stuck in local minima that generate some extra components. Furthermore VB converges always faster than MML (as it was already outlined in [4]).

Figure 5.2 represents the variational bound function of the iteration number: convergence is achieved after few iterations. Figure 5.3 represents number of surviving components function of iteration number: here again after few iterations convergence is achieved and correct number of cluster is found (3).

Figure 5.4 represent log-likelihood function of iteration for MML/EM technique: in this case convergence is slower, but the most important problem is shown in figure 5.5 where component number is function of iteration; more than 400 iteration were needed to converge.

Obviously those results are shown with a single initialization but using different initialization we verified the same problem.

5.1.3 KL distance between clusters

After running feature selection, we have all features sorted on the base of their "saliency". A simple way to see the measure of relevant and irrelevant features is the KL distance between different cluster components.

KL distance between two gaussian distributions $q(x) = N(x; \mu_q, \Sigma_q^{-1})$ and $p(x) = N(x; \mu_p, \Sigma_p)$ is given by:

$$KL(q||p) = 0.5\, log(\frac{|\Sigma_p|}{|\Sigma_q|}) + 0.5\, Tr(\Sigma_p^{-1}\Sigma_q) + 0.5(\mu_q - \mu_p)^T \Sigma_p^{-1}(\mu_q - \mu_p) - \frac{d}{2} \quad (5.1)$$

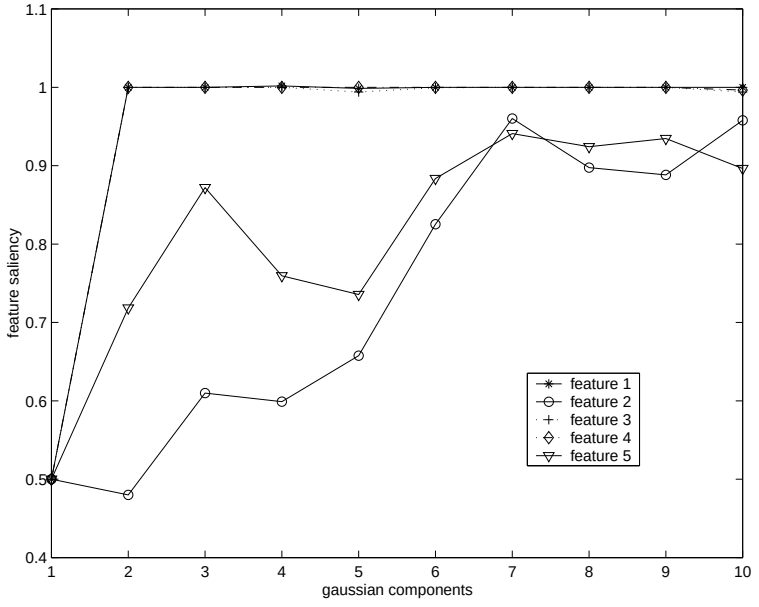

Figure 5.1: Feature saliency vs. component number; it's not possible to distinguish between features one, three and four because they have almost the same value (1)

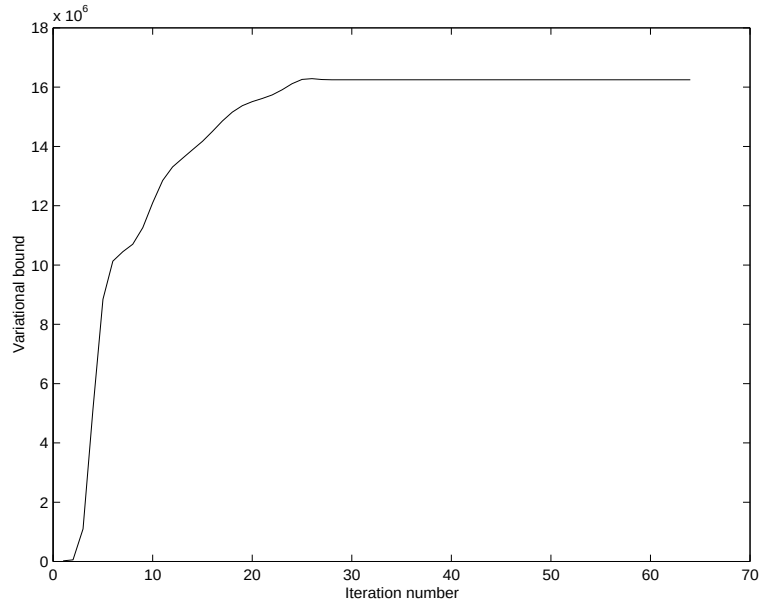

Figure 5.2: Variational bound vs. number of iteration

In our case each feature distribution is actually represented as a gaussian mixture with two components, the first one cluster dependent and the second one cluster independent. KL distance can be estimated numerically, or eventually an average pair-wise KL distance can be used (introduced in [5]); given two GMM $q(x) = \sum_i \alpha_i N(x; \mu_{iq}, \Sigma_{iq}^{-1})$ and $p(x) = \sum_j \beta_j N(x; \mu_{jp}, \Sigma_{jp})$ the average pair-wise KL distance can be written as:

$$PKL(q||p) = \sum_i \sum_j \alpha_i KL(q_i||p_j) + \alpha_i log \frac{\alpha_i}{\beta_j} \tag{5.2}$$

We can so define a pair-wise KL divergence:

$$D = \sum_i \sum_{j \neq i} (PKL(p_i||p_j) + PKL(p_j||p_i)) \tag{5.3}$$

Features are progressively removed on the base of their saliency and expression 5.3 is computed using the new feature subset. Result is shown in figure 5.6.

In Figure 5.6, it's evident that removing the less relevant features (i.e. features that cannot discriminate between gaussians) leaves unchanged the pair-wise KL distance between cluster components; when relevant features are removed suddenly the KL distance decrease.

5.1.4 Observation entropy

The second criterion is based on data entropy; let's define as in [1]:

$$w_{ij} \propto \hat{\alpha}_j \; p(y_i|\Theta_j) \;\; v_{ij} \propto \hat{\alpha}_j \; p(y_i|\Theta_U) \tag{5.4}$$

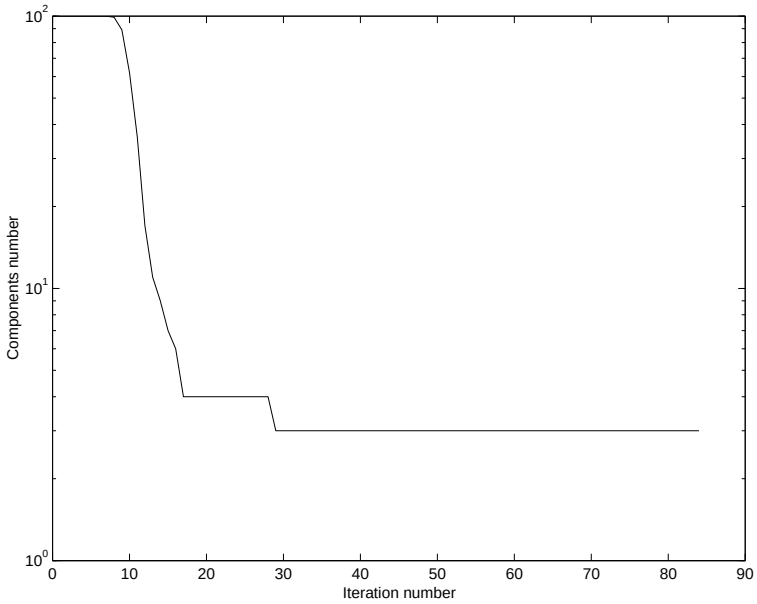

Figure 5.3: Number of components vs. number of iteration for variational bayesian learning; Y axis is logarithmic scale

w_{ij} measure the probability of the observation y_i to belong to cluster j and not to the common data distribution whose probability is given by v_{ij}. In the case of only relevant features, it should be $w_{ij} = 1$; a simple way to measure how good the feature subset is, consists in entropy $H(w_{ij}) = 1/n \sum_i \sum_j w_{ij} log(w_{ij})$.

5.2 Speech Recognition

In this section we describe experiments we run on speech data obtained by the TIMIT database.

A huge number of front end techniques have been proposed with high redundancy between them because based almost on the same principle. To study the efficiency of algorithms previously proposed we tried to determine feature saliency of following feature set: 12 MFCC+12 Δ+12 $\Delta\Delta$,12 PLP+12 Δ+12 $\Delta\Delta$. Data from the TIMIT database are processed in order to obtain a 75 component features vector.

Feature saliency algorithms (MML and VB) have been applied to the feature set with respectively a training set of dimension 2k,20k and 200k, in order to study the robustness w.r.t. the amount of data.

Figures 5.7,5.8,5.9, shows feature saliency obtained by the MML feature selection algorithm with respectively 2k,20k and 200k training observation; Figures 5.10,5.11,5.12, shows feature saliency obtained by the MML feature selection algorithm with respectively 2k,20k and 200k training observation;

Of course we tested the efficacity of the selected features, using respectively the first 24 and 39 more robust features.

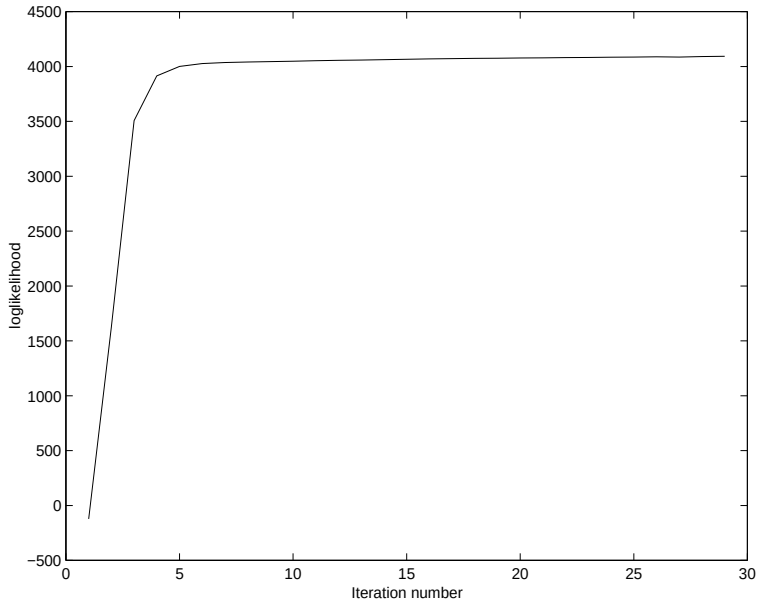

Figure 5.4: Loglikelihood vs. number of iteration

data amount	2k	20K	200K
MML learning (24 features)	N/A	59.3 %	60.17 %
VB learning (24 features)	57.23 %	60.6%	61.8%
MML learning (39 features)	N/A	64.6 %	64.6 %
VB learning (39 features)	64.4 %	64.6%	65.7%

Table 5.1: Recognition rate

5.3 Conclusion

After running experiments on synthetic and speech data we can conclude that using VB learning has many advantages compared to MML learning. First of all, VB converges faster than MML, saving computational time. Then VB seem to determine features in a more robust way; it comes from the fact that recognition rate coming from VB selected features is always higher to recognition rate coming from MML selected features. This is probably due to the fact that the clustering is pretty different for the two approach. Furthermore VB learning benefits form regularization effects coming from prior distributions. This can be seen when very few training data are used: MML based method prunes almost all components, while VB based method achieved a "regularized" solution probably thanks to priors.

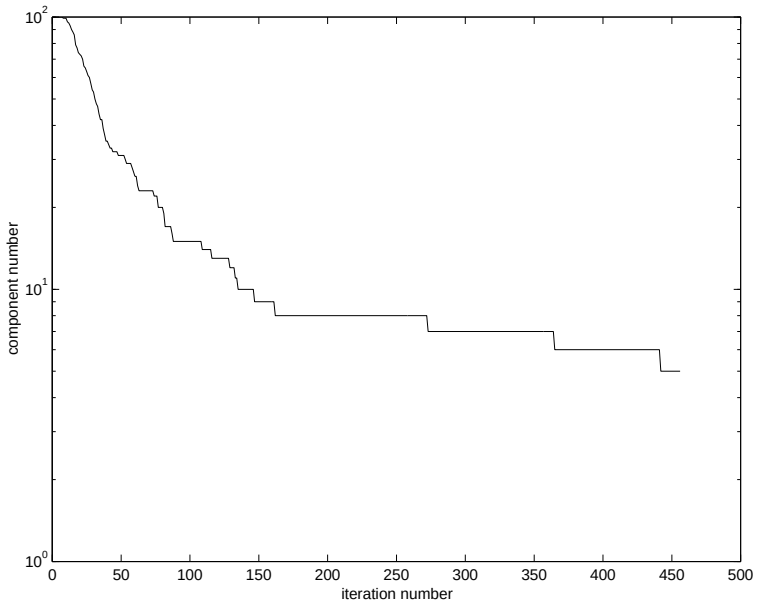

Figure 5.5: Number of components vs. number of iteration for MML/EM algorithm

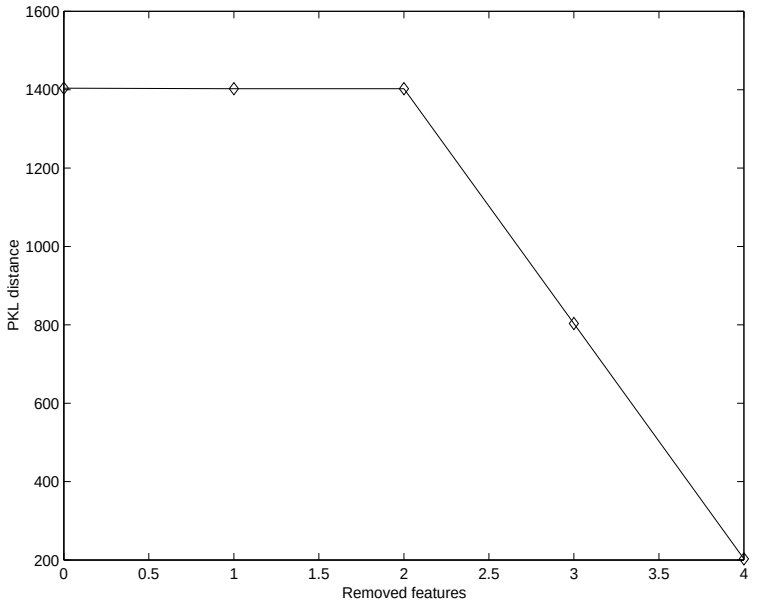

Figure 5.6: PKL distance between clusters obtained progressively removing features on the base of their saliency

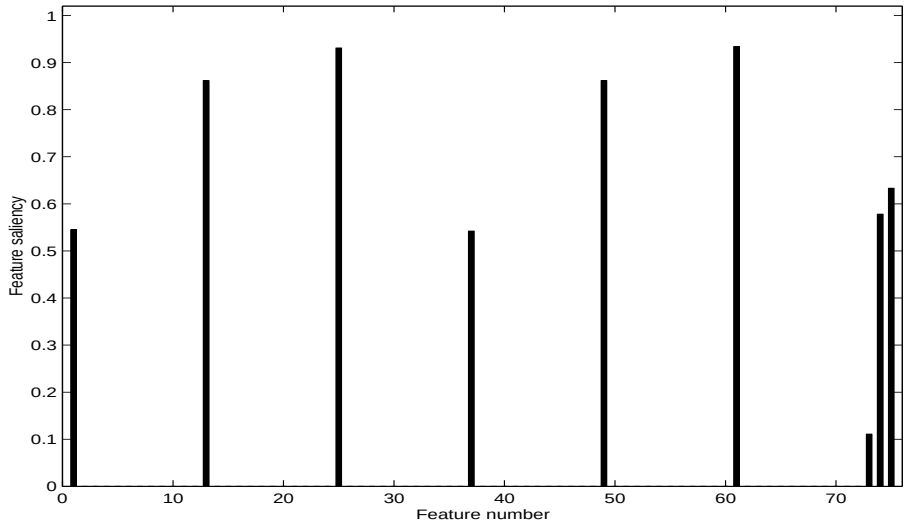

Figure 5.7: Feature saliency with MML algorithm (inferred components 2) with 2000 training vectors

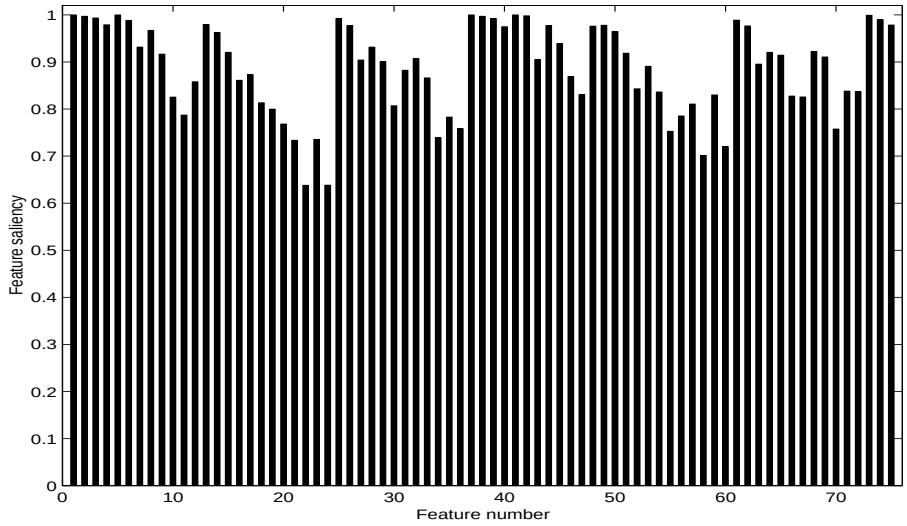

Figure 5.8: Feature saliency with MML algorithm (inferred components 68) with 20000 training vectors

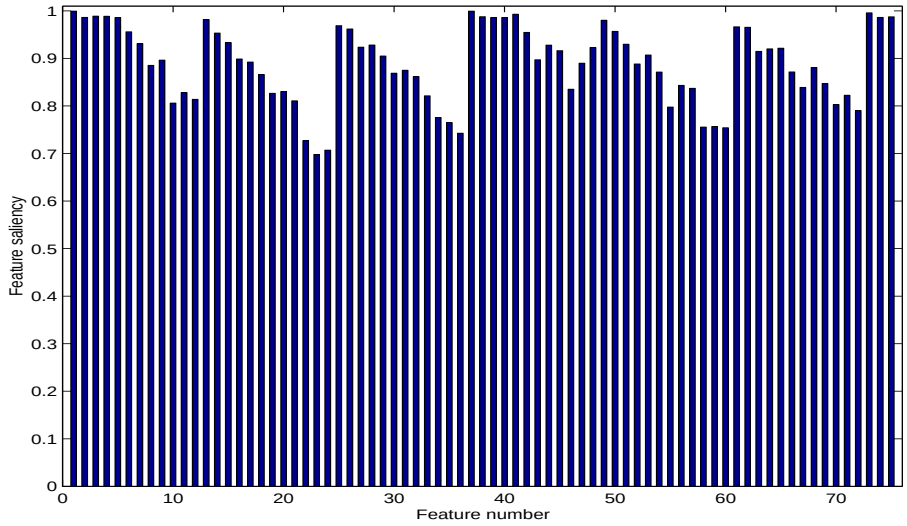

Figure 5.9: Feature saliency with MML algorithm (inferred components 100) with 200000 training vectors

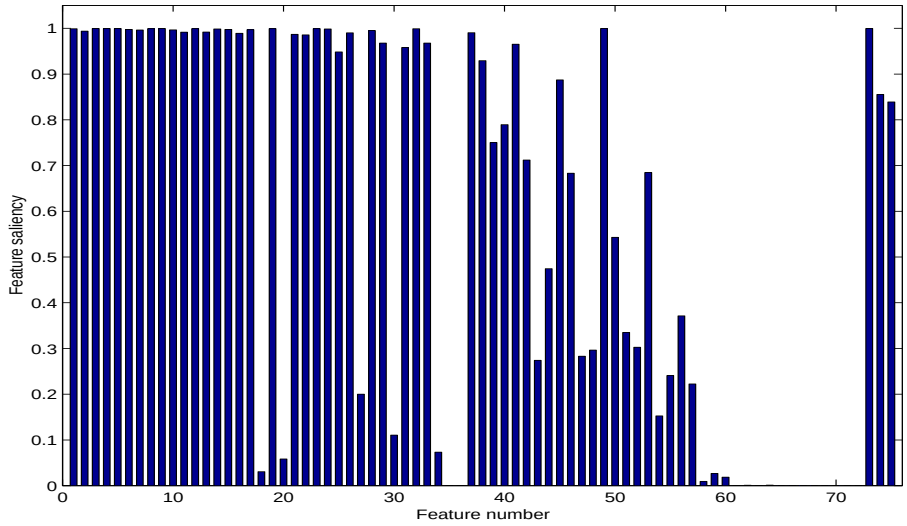

Figure 5.10: Feature saliency with VB algorithm (inferred components 48) with 2000 training vectors

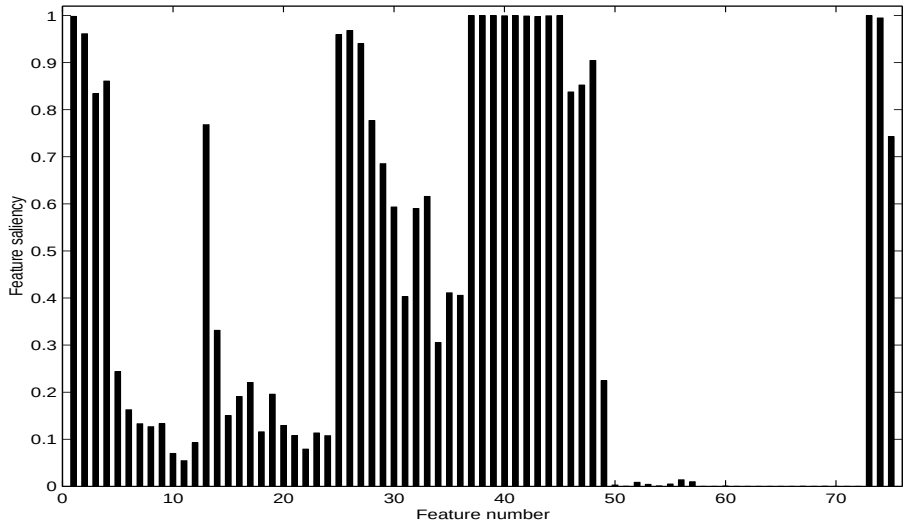

Figure 5.11: Feature saliency with VB algorithm (inferred components 50) with 20000 training vectors

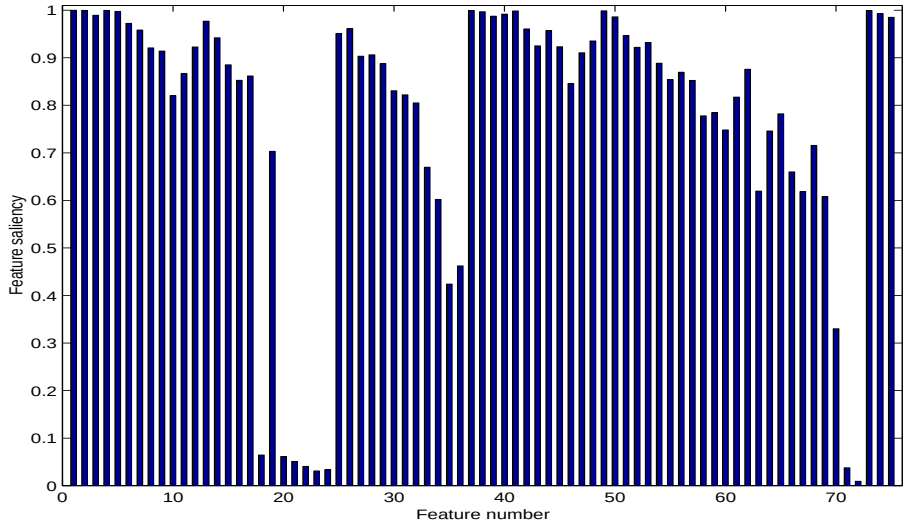

Figure 5.12: Feature saliency with VB algorithm (inferred components 75) with 200000 training vectors

Bibliography

[1] Figueriredo M. A. Law M. H., Jain A. K. Features selection in mixture based clustering. *NIPS*, 2002.

[2] Jain A. Law M., Figueredo M. Feature saliency in unsupervised learning. Technical report, Michingan state university, 2002.

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

[4] Valente F.; Wellekens C. Variational bayesian learning for gaussian mixture models and hidden markov models. Technical Report RR-03-079, Institut Eurecom, 2003.

[5] Omar M.K.;Chen K.;Hasegawa-Johnson M.;Brandman Y. An evaluation of using mutual information for selection of acoustic features representation of phonemes for speech recognition. 2002.