Variational Bayes Latent Variable Models And Mixture Extensions

Master Thesis (Lyngby 2004)
Supervised by *Ole Winther and Lars Kai Hansen*

Handed 21st June by:

Slimane Bazaou s974608

IMM

2004 – 48

Technical University of Denmark
Building 321
2800 Lyngby

Preface

This master thesis serves as documentation for the final assignment in the requirements to achieve the Master degree of Science in Engineering. The work has been carried out at the Intelligent Signal Processing group at the Institute of Informatics and Mathematical Modelling, Technical University of Denmark. I wish to thank Professor Lars Kai Hansen and Associate Professor Ole Winther, for supervising, guidance and inspiring me through out my thesis.

Kgs. Lyngby, Juni 21st, 2004

Slimane Bazaou, s974608

Abstract

This thesis is concerned with the problem of applying an *approximate Bayesian learning* technique referred to as *variational Bayes* to different *Gaussian latent variable models and their mixture extensions.*

I will try to give a smooth transition between the different models used in this thesis, starting from the single (multivariate) Gaussian model to the more complex *Linear Factor* and its mixture extension *Mixture of Factor Analyzers* model, where either/both the *hidden* dimensionality and the hidden number of components (in the case of mixtures) are unknown.

One of the aims of this thesis is to investigate how the Bayesian framework infers the wanted parameters, e.g. number of components in a mixture model, given a model, and how it succeeds in solving the different problems related to overfitting. I also investigate which one of these models that perform best for a range of tasks. Throughout the report I will try to discuss the performance of the Bayesian techniques, mainly by comparing it to the standard *Maximum Likelihood* approach.

Each of the models discussed in the thesis are applied to one or more of the following problems: *Density estimation*, *Classification*, *Signal separation* and *Image compression.* Both synthetic and real data are tested.

Keywords: Graphical Models, Linear latent variable models, Mixture Models, Maximum Likelihood, Bayesian Inference, Variational Bayes.

Contents

List of Figures

Nomenclature

Below follows the most used symbols and abbreviations.

EM Expectation maximization.

VB Variational Bayesian.

FA Factor analysis.

$\textsc{Tr}[\mathbf{X}]$ Trace operator, i.e, $\sum_j \; diag_j[\mathbf{X}]$,see DIAG.

MLFA Maximum likelihood factor. analysis.

VBFA Variational Bayesian factor analysis.

ARD Automatic relevance determination.

KL Kullback-Leibler divergence.

IID Independently and Identically Distributed

$\textsc{diag}[\mathbf{X}]$ Diagonal operator, i.e, $diag_j[\mathbf{X}] = x_{j,j}$.

PCA Principal component analysis.

PPCA Probabilistic Principal component analysis.

AIC Akaike's information criterion.

$\langle\cdot\rangle$ Expectation operator also denoted by $\mathcal{E}[\cdot]$.

$p(\cdot)$ Probability distribution.

$\mathcal{Q}(\cdot)$ Variational posterior distribution.

d_X Data dimension.

N Number of samples.

$\mathbf{X}$ Data matrix of dimension $[d_X \times N]$.

$\mathbf{S}$ Hidden states $[d_S \times N]$.

$\mathbf{A}$ Factor loading matrix of dimension $[d_X \times d_S]$

$\mathcal{F}(\cdot)$ Lower bound on the log marginal likelihood.

$\mathcal{L}_{inc}(\mathbf{\Theta})$ Incomplete data log likelihood.

$\mathcal{L}_c(\mathbf{\Theta})$ Complete data log likelihood.

CHAPTER 1

Introduction

The ever increasing amount of production and consumption of information in those recent decades, demand robust and reliable analyzing and processing techniques, to make it even possible to work with, and to extract some, for the user, useful data. Text search in the internet is just one example. In *Machine learning* models are developed to make these task possible. When constructing such models at least two different learning problems can be used: *supervised* and *unsupervised.*

Supervised Learning is the case where the data set $\mathcal{D}$ consists of pairs of *patterns* $\mathbf{x}_i$ and *targets* (e.g., labels for classification) $\boldsymbol{t}_i$ represented as:

$$\mathcal{D} = \{(\mathbf{x}_1, \boldsymbol{t}_1), \ldots, (\mathbf{x}_N, \boldsymbol{t}_N)\}$$

known as the *training* set. In the case where the targets are discrete classes, we are dealing with a *classification* problem. If, otherwise, the target are real valued, the problem is then referred to as a *regression* problem. This will not be discussed further in this report, for a comprehensive and detailed book see [5].

Unsupervised Learning on the other hand is more diverse and difficult to define. This type of learning mostly deals with discovering clusters or other structures in the input data, without having any knowledge of class labels for the data and obvious criteria to guide the search. A convenient way of dealing with many forms of unsupervised learning in a probabilistic way is through density estimation. One of the most popular density estimation methods is the *Gaussian Mixture Model* (GMM) (section 5.1). Promising alternatives to GMMs are the *Latent Variable Models* and their Mixtures extension. Examples of these models are *Probabilistic Principal Component analysis* (PPCA) and *Factor Analysis* (FA) (section 4.3), and their mixture extension *Mixture of* PCA and *Mixture of* FA (section 5.2). The advantage of these latent variable models is that they are capable of representing the covariance structure with less parameters by choosing the dimension of a subspace in a suitable way. This is explained in section 4.2. An empirical evaluation on a large number of data sets shows that mixtures of latent variable models almost always outperform GMMs (5.1).

What we are really interested in, is a model which not only learns the training set but also *generalizes*[1] well on unseen examples. In a probabilistic way, we assume that there is some unknown probability density function $p(\mathbf{x})$ (or $p(\mathbf{x}, \boldsymbol{t})$ in the supervised case), from which all examples are drawn independently, and of which the training set is a sample. Learning then involves extracting that information from the training data, which is characteristic for the density $p(\mathbf{x})$, while avoiding two extreme situations. The

[1] In probabilistic modelling, we will define generalization of model $\mathcal{M}$ by the probability it assigns to a new previously unseen validation data-set.

first one occurs when the model learns the inherent noise in the *finite* training set, this is commonly referred to as *overfitting*. The other extreme situation appears where the model is too simple or not flexible enough, this is known as *underfitting*. It can be seen that the learning has to deal with a kind of dilemma between the bias and variance, and its believed that the best model is the one that balances between these two problems [5]. In *maximum likelihood* (ML) learning, also referred to in many literatures as *conventional learning*, the choice of the model complexity[2] requires the use of methods based on, for example, *cross-validation* techniques, briefly explained in section 3.1. To choose an appropriate model, these techniques are computationally expensive, wasteful of data, and give noisy estimate for the optimal number of components [7]. An appropriate learning technique, which efficiently uses the data set and returns the posterior distribution over the model complexity, is the *Bayesian* treatment. Techniques of solving the Bayesian problem can be characterized as follows [14] (part IV):

- **Exact Methods** which computes the required quantity directly. However, due to their computational complexity, they are still rarely used in Machine Learning. There exist however exceptions such as exact inference in Bayesian networks with the *junction tree algorithm* which is widely used [15].

- **Approximate Methods** which can be subdivided into

  - **Deterministic approximation** this subgroup includes: *Maximum Likelihood* (section 3.1), *Laplace*'s method , and *Variational methods* (section 3.3)
  - **Monte Carlo methods** are mainly used when the deterministic approximation, such as Laplace's method, does not work. Some methods to implement Monte Carlo are: *important sampling*, *rejection sampling*, the *Metropolis* methods and *Gibbs sampling* [14]. One of the disadvantages of these methods is that they require saving the whole posterior distribution, and they are usually too slow. An implementation of fully Bayesian technique using *Markov Chain Monte Carlo* (MCMC) for a mixture model can be found in [2].

Neither Monte Carlo methods, nor the exact methods will be discussed further in this thesis. The focus is concentrated on *Variational Bayesian* (VB) *techniques* and *Maximum Likelihood*. Maximum Likelihood, though, does not return the posterior distribution over the model, but can be extended by a *regularization* factor (proportional to the prior)to penalize the large or complex models.

Throughout this thesis a comparison of the performance of ML and the VB is discussed. The discussion will in many cases be illustrated graphically.

Roadmap

So far I have given a very short description of some learning techniques, which are going to be used in the thesis. The general flow goes from the simple model to more complex model, and from Maximum Likelihood to Variational Bayes. But first some theory on both techniques are presented.

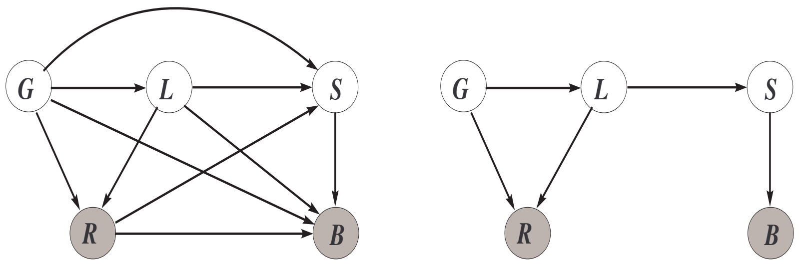
[2]by the complexity I mean the number of components and/or the number of dimensions depending on the used model

Structure of the course of the thesis looks like:

- **Chapter 2:** describes *Bayesian Networks*, relevant notations and how they can be used in learning models.
- **Chapter 3:** deals with learning in Bayesian networks. The two learning methods ML and Bayesian approach are discussed. The last section of this chapter introduces the Bayesian approximation method called *Variational Bayes*.
- **Chapter 4:** introduce and links several models used in machine learning. All these models can be included in the framework of *Latent variable models*. A detailed derivation of ML and VB for these model is given.
- **Chapter 5:** introduces the mixture extension to linear latent variable models, and gives a general idea of how these models are related to each other
- **Chapter 6:** focuses on model order selection. VB and BIC penalized ML models are tested, on real and synthetic data.

CHAPTER 2

Bayesian Networks



(a) A Bayesian Net representation of full version of joint distribution (equation (2.1)), for the Lie Detector problem

(b) A simplified Bayesian Net representation of figure 2.1(a) (Equation (2.2)).

Figure 2.1: A Bayesian networks corresponding to the probabilistic model used in a simplified Lie Detector problem [46]. The graphical model represents a conditional decomposition of the joint probability in equation (2.1). The circular nodes represent random variables, with some distribution over them. The shaded circles are the *observed* variables, while the unshaded are *hidden* ones. The arcs (edges) goes from the *parent/parents* to the *child/children*, and represent the probabilistic connection between two variables. The lack of arcs encode conditional independencies. (a) Represent the *model* of the full version of the joint distribution, corresponding to equation (2.1). (b) The unnecessary arcs (dependencies) are removed using expert knowledge and definition (2.1); the resulting graph corresponds to equation (2.2).

In real world problems, we might be faced with a problem involving a large number of variables, hundreds or thousands. For example, a digital color image contains many millions of measurements, which discourage representing or manipulating with the *joint* distribution of all these variables. However, we can assume that, of all possible direct dependencies between variables, only a fraction are needed in most interesting problem domains. The dependencies and independencies between variables can be represented graphically, in the form of *Probabilistic Graphical Models*.

2.1 Graphical Models

"Graphical models are a marriage between probability theory and graph theory. They provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering ...
a complex system is built by combining simpler parts. Probability theory provides the glue whereby the parts are combined, ensuring that the system as a whole is consistent...

Many of the classical multivariate probabilistic systems studied in fields such as statistics, systems engineering, information theory, pattern recognition and statistical mechanics are special cases of the general graphical model formalism – examples include mixture models, factor analysis, hidden Markov models, Kalman filters and Ising models. The graphical model framework provides a way to view all of these systems as instances of a common underlying formalism.... "

Michael I. Jordan , Learning in Graphical Models
http://www.ai.mit.edu/~murphyk/Bayes/bnintro.html

The probabilistic graphical models are not only a tool of visualizing the relationship between variables but, by exploiting the conditional independence relationships, also provide a backbone upon which, it has been possible to derive efficient algorithms such as *massage-propagating* algorithms [46] for updating the uncertain beliefs of the machine [17]. In the graph models the nodes represent variables, and arcs (edges) represent dependencies. There are two types of nodes, *circles* and *rectangles*: circles denote *random variables*, (with distribution over them) while rectangles correspond to *deterministic*[1] parameters (i.e., fixed although unknown variables [9]), with no distribution over them. The circles in their turn are subdivided into two groups: *observed* shaded circles, and *hidden* unshaded circles.
There are also two kinds of arcs: *directed* (marked with arrows) and *undirected*.
Graphs with the former type of arcs are called *directed graphs*, where the basic graph model is the *Bayesian Networks* also called *Belief Nets*, which is most popular in *Artificial Intelligence*. The arcs are taken to represent the conditional relationships between the variables corresponding to the parent and the child.
As their name implies *undirected graphs* are graphs with undirected arcs. These graphs are sometimes called *Markov network*, and are used in image processing and statistical physics. There are also graphs involving both types of arcs, and this type of graph is called *Mixed graphical models*. For more detail about graphical models and their extensions see e.g. the tutorials by David Heckerman [9], and Buntine [28].

2.2 Bayesian Networks

Bayesian Networks is the type of graph we are dealing with in this thesis. The directed arcs in this type of graphs are used exclusively to form a directed acyclic graphs (DAG). Graphical models are based on a trivial yet important notion of independence, which is worth to state here.

Definition 2.1 $\mathbf{x}$ *is independent of* $\mathbf{s}$ *given* Θ *if* $p(\mathbf{x}, \mathbf{s}|\Theta) = p(\mathbf{x}|\Theta)p(\mathbf{s}|\Theta)$ *whenever* $p(\Theta) \neq 0$ *for all* $\mathbf{x}$, $\mathbf{s}$ *and* Θ*,*

The ability of the graph models to represent the conditional decomposition of the joint distribution, which is an extension to definition (2.1), together with the important notations of *hidden* and *observed* variables are illustrated in the following simplified Lie Detector example adapted from J. Winn [46]

[1] It is just like how the frequentist looks at the parameters they want to infer when using Maximum Likelihood, we return to that shortly.

Example 2.1 *Consider building a simplified Lie Detector machine, where the aim is to determine whether a suspect is guilty of a crime. The binary variable G is true if the suspect is guilty. The yes/no response of the suspect, of whether he committed the crime is recorded in R and let L represent whether the suspect is lying. The principle behind lie detectors is that most people will become stressed when attempting to deceive another person and that this can be detected by examining their physiological reaction. Let S be whether the suspect is stressed and B be some biophysical measurements (heart rate, respiratory rate, skin resistance etc.). Since the suspect has no interest in facing jail, he will try to hide his guilt (G), by lying (L). The true value of (L) and (G) are only known by the suspect himself, and hidden from the detectives, these variables are therefore referred to as* hidden variables. *To cheat the detectives, the suspect try to hide symptoms of stress (S) while lying. (S) is therefor hidden too. Unfortunately (for the suspect) stress can be measured, and the result can be* seen *in the (B). (B) is therefore an* observed variable.
To express the joint probability distribution over these five variables we can use the chain rule (figure 2.1.a).

$$P(G, L, R, S, B|\mathcal{M}) = \tag{2.1}$$
$$P(G|\mathcal{M})P(L|G, \mathcal{M})P(R|G, L, \mathcal{M})P(S|G, L, R, \mathcal{M})P(B|G, L, R, S, \mathcal{M}),$$

where $\mathcal{M}$ is the conditioning context, for instance the expert's knowledge and/or the choice of the Architecture *of the graphical model, (the architecture is referred to some times as* model*). From the expert's knowledge it can be assumed that the suspect's stress levels depend only on whether he is lying and so $P(S|G, L, R, \mathcal{M})$ can be simplified to $P(S|L, \mathcal{M})$. Similarly, it is assumed that the bio−physical measurements depend only on whether the suspect is stressed, so $P(B|G, L, R, S, \mathcal{M})$ reduces to $P(B|S, \mathcal{M})$. Now we can rewrite our joint distribution as a product of factors each involving only a small subset of the variables (figure 2.1.b):*

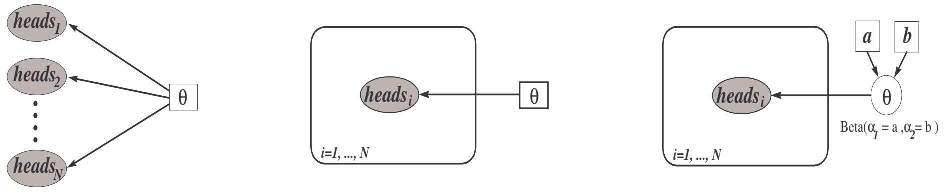
$$P(G, L, R, S, B|\mathcal{M}) = P(G|\mathcal{M})P(L|G, \mathcal{M})P(R|G, L, \mathcal{M})P(S|L, \mathcal{M})P(B|S, \mathcal{M}). \tag{2.2}$$

In this example, we have exploited conditional independencies between variables in the model, to factorize the joint distribution. In equation (2.2) some expert's knowledge has been used to reduce the dependencies. Each variable is then writing conditioned on its *parents*, where parents($\mathbf{x}$) (parents of x) in the Bayesian networks is the set of variables with direct arc into x.
A Bayesian network is a compact representation of a full joint probability distribution. A general way of writing this equation is

$$p(\mathbf{X}|\mathcal{M}) = \prod_{\mathbf{x}\in\mathbf{X}} p(\mathbf{x}|parents(x), \mathcal{M}). \tag{2.3}$$

Bayesian networks representation of equations (2.1) and 2.2 are illustrated in Figure 2.1(a) and 2.1(b), where *observed variables* are shown as shaded nodes. In figure 2.1(b) the lack of possible arcs between variables (nodes), encodes conditional independencies.

Another important extension which allows us to use graphical models to represent and reason about the task of *learning*, is the notion of a *plate* or *replicated node*. This will be explained, too, using an example (this time taken from Buntine [28]).



(a) The coin model with repeated group of $heads_i$ and deterministic parameter θ.

(b) The coin model with plate and deterministic parameter θ.

(c) The same model as (b) with deterministic *hyperparameters* $\{a, b\}$.

Figure 2.2: Bayesian Net of a simple unsupervised learning problem containing uncertainty (see example 2.2), where we model tossing a biased coin with physical probability of heads equal to θ. The observed variables are shown as shaded circles, the deterministic[3] variables are shown as *rectangles* and stochastic hidden variables are the non-shaded circles. When assuming $heads_i$ to be (IID) the repeated group of ($heads_i$, $i = 1, \cdots, N$) in (a) can be replaced by a single node in (b), with a *plate* (rounded box) around it. (b) This type of Bayesian nets models where the parameters θ are deterministic (*rectangle*), is referred to as ML models (see 3.1). (c) θ is a stochastic variable (*circle*), the parameters $\{a, b\}$ governing its distribution are referred to as *hyperparameters*, they are deterministic in this case.

Example 2.2 *Consider a very simple unsupervised learning problem containing uncertainty, where we have a biased coin with an unknown bias θ for heads (and $(1-\theta)$ for tail). that is, the long-run frequency of getting heads for this coin on a fair toss is θ. The coin is tossed N times and each time the binary variable $heads_i$ is recorded (0/1 for heads/tail respectively and $i = 1, \cdots, N$) . The graphical model can be seen in figure 2.2. Assume that all the data of size N, is generated by sampling from the* Binomial distribution *with* parameter θ *(i.e N_{heads} $\sim$Bin(N,θ), where N_{heads} is the number of heads in N trials). Nodes $heads_i$ in figure 2.2 are represented as* circles*, their value is given at each trial, $heads_i$ are therefore* observed variables *and they are represented as* shaded *circles. θ is the parameter to infer, when we assume that its value is (although unknown) as in the case of Maximum Likelihood inference, the parameter is represented as a* rectangle*, see figure 2.2(a) and 2.2(b). When θ is assumed to be a* stochastic *variable with a distribution over it (i.e.* prior distribution $p(\theta)$*), the node is represented as a* circle*, since its value is hidden the circle is* unshaded *see figure 2.2(c). The prior could be in this case* $p(\theta|a, b) \sim \text{Beta}(\alpha_1 = a,\ \alpha_2 = b)$ (Appendix *B). Since a and b are parameters for the distribution of another parameter θ, we referred to them as* hyperparameters.
The joint distribution of everything in figure 2.2(a) can be writing as:

$$\begin{aligned} p(\theta, heads_1, \cdots, heads_N) &= p(\theta)p(heads_1|\theta)p(heads_2|heads_1, \theta) \\ &\quad \cdots p(heads_N|heads_1, \cdots, heads_{N-1}, \theta) \end{aligned} \tag{2.4}$$

*With the assumption that $heads_i$ are independently and identically distributed (*IID*), the repeated group of $heads_i$ nodes in figure 2.2(a) can be replaced by a single node, with a box around it in figure 2.2(b). This box is referred to as a* plate*. Equation (2.4) can* *be rewriting as*

$$\begin{aligned} p(\theta, heads_1, \cdots, heads_N) &= p(\theta)p(heads_1|\theta)p(heads_2|\theta)\cdots p(heads_N|\theta) \\ p(\theta, heads_1, \cdots, heads_N) &= p(\theta)\prod_i^N p(heads_i|\theta) \end{aligned} \tag{2.5}$$

The result from equation (2.5) is what we will get if we try to write down the joint distribution of figure 2.2(b).

The *plate* introduced in example (2.2) (figure 2.2(b)) implies that

- The enclosed subgraph is duplicated N times.
- The enclosed variables are indexed.
- Any exterior-interior links are duplicated

When dealing with *plates* in Bayesian networks, one should express the joint probability ignoring the plates, and take a product to index the variable inside it, as in equation (2.5). The next question is how to learn such models.

CHAPTER 3

Inference in a Bayesian Network

Once the Bayesian network is constructed for a certain problem involving unknown parameters (variables). Logically '*the*' task is to find the true value of these parameters. However, in almost all practical cases, finding these values involves some impossible computations, for instance, example (2.2) the task was to find the *true head's* physical probability θ:

$$\theta_{true} = \lim_{N\to\infty} \frac{N_{heads}}{N}.$$

This is not possible to compute since it requires infinite number of trials. Then given the limited amount of observations (*training* data) at hand, an *single* estimate to the true parameters (*maximum likelihood* way) or a distribution over them (*Bayesian technic*) should be *inferred*.

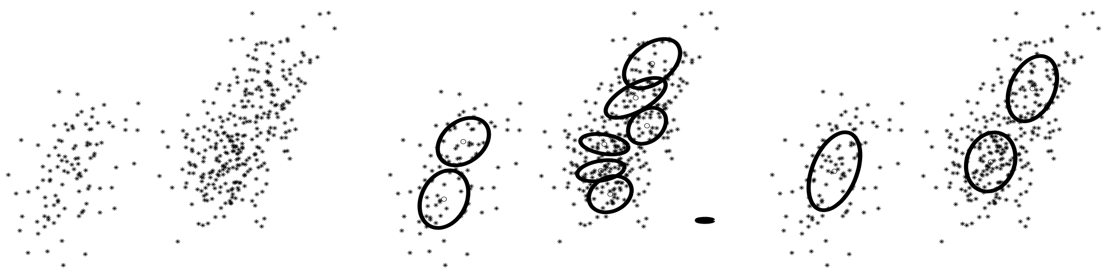
In the frequentist school, a single parameter (set of parameters) $\mathbf{\Theta}^* = \{\theta_j^*\}_{j=1}^K$ which maximizes the fit to the data, is to be found. The fit in this case is measured by the likelihood function $p(\mathbf{x}|\theta)$. Due to the monotonicity of the *logarithm*, as well as its many advantages, such as concavity and algebraic convenience, the log likelihood $\mathcal{L}(\theta) = \ln p(\mathbf{x}|\theta)$ is taken instead, as a measure of the fit.

The other school (Bayesian school) has a different view of what it means to learn from data, in which probability is used to represent uncertainty about the relationship being learned. In a Bayesian sense, learning a model involves calculating the posterior probability density over the possible model parameter values. In both methods the inference can be divided into two levels:

Model fitting: Here we assume that one of the models $\mathcal{M}$, which we invented is true. By model in Bayesian nets we mean the number of parameters (nodes in the graph), whether they are stochastic variables (circles) or deterministic variables (rectangles), and the dependencies between those parameters (the presence or lack of the arcs). The model is then fitted to the data. Typically a model includes free parameters $\mathbf{\Theta}$. Fitting the model to the data involves inferring what values those parameters should take, given the data. This level is repeated for each model.

Model selection: Here we compare the different models in the light of the data. Using some measure, a "best" model is chosen.

The two different schools solve the above levels differently. This is what we are going to discuss in the rest of this chapter.



(a) The observed data (b) A possible model $\mathcal{M}_1$. (c) Another possible model $\mathcal{M}_2$.

Figure 3.1: This figure shows an unsupervised problem of density estimation using mixture of gaussian models. (a) is the observed data, generated by a mixture of 4 gaussians. (b) is a possible model with a mixture of 9 gaussians. (c) is an other possible model, this time with a mixture of only 3 gaussians. There is no doubt that the model in (b) will have higher score, than the one in (c), in form of higher likelihood, i.e., $\mathcal{L}_{inc}(\mathbf{\Theta}_{\mathcal{M}_1}|\mathcal{M}_1) > \mathcal{L}_{inc}(\mathbf{\Theta}_{\mathcal{M}_2}|\mathcal{M}_2)$, ML tends to prefer the more complex models that over-fit to data, i.e., it does not penalize complex models. This is in fact one of the major problems of ML. A regularization factor can though be add to the likelihood function to take into account the complexity of the model, while training.

3.1 Maximum likelihood

As mentioned earlier, a frequentist way of learning a model, where the complexity is taken into account, can be done in two levels: Model fitting and model comparison.

Model fitting: The frequentists assume that the parameters are deterministic variables and try to infer them by maximizing the (log-) likelihood function.
Given a certain model $\mathcal{M}$ to be true, a general expression of the log-likelihood function of parameters $\mathbf{\Theta} = \{\theta_j\}_{j=1}^{K}$ can be written as:

$$\begin{aligned} \mathcal{L}(\mathbf{\Theta}|\mathcal{M}) &= \ln p(\mathbf{X}|\mathbf{\Theta}, \mathcal{M}) \\ &= \ln \prod_i p(\mathbf{x}_i|\mathbf{\Theta}, \mathcal{M}) \\ &= \sum_i \ln p(\mathbf{x}_i|\mathbf{\Theta}, \mathcal{M}), \end{aligned} \tag{3.1}$$

where in the second line of the above equation, we used the assumption implied by the plate that the examples are IID, which allow us to factorize the probability $p(\mathbf{X}|\mathbf{\Theta}, \mathcal{M})$. The estimate $\mathbf{\Theta}^*_{\mathcal{M}}$ is then found by

$$\mathbf{\Theta}^*_{\mathcal{M}} = \underset{\mathbf{\Theta}}{\text{argmax}}\ \mathcal{L}(\mathbf{\Theta}|\mathcal{M}). \tag{3.2}$$

This is intuitively appealing since it corresponds to values for $\mathbf{\Theta}$ which describe the data well.
Since we are going to deal with *Latent Variable Models* (section 4.2), it is convenient to assume that we also have *hidden states* (or *missing data*[1]) $\mathbf{s}$ that help in modelling the observed data $\mathbf{X}$. The *hidden states* can, for example, be discrete component labels

[1] *hidden state* or *missing data* in Bayesian nets, are the hidden variables (unshaded circles) that are inclosed with the observed data in a plate. Hidden state can either be continuous or discrete variables.

which represent a sort of imaginary class labels for the observed data. In fact this is a way to define *mixture models*, as we will see in section 5. The *factors* in a *Factor Analysis Model* (section 4.3) are also an example of *hidden states*. The log-likelihood of $\mathbf{\Theta}$ is then obtained by marginalizing over the hidden states $\mathbf{S}$. Since $\mathcal{L}(\mathbf{\Theta}|\mathcal{M})$ in equation (3.1), does not include these *missing data*, this quantity is usually referred to as *incomplete log-likelihood*, and from now on is written as $\mathcal{L}_{inc}(\mathbf{\Theta}|\mathcal{M})$.
Including the marginalization over the *missing variables* $\mathbf{S}$, equation (3.1) can be rewritten as:

$$\mathcal{L}_{inc}(\mathbf{\Theta}|\mathcal{M}) = \ln p(\mathbf{X}|\mathbf{\Theta}, \mathcal{M}) = \sum_i \ln \int p(\mathbf{x}_i, \mathbf{s}|\mathbf{\Theta}, \mathcal{M}) \, d\mathbf{s}. \tag{3.3}$$

An intuitive approach to maximize $\mathcal{L}_{inc}(\mathbf{\Theta}|\mathcal{M})$ is to take the partial derivative to each of its parameters θ_j, and try to solve for zero.
Since the integral (or sum) over $\mathbf{s}$ is required to obtain the marginal probability of the data, and due to the fact that the log of the integral can potentially couple all of the parameters of the model, maximizing (3.3) directly is often difficult. Furthermore for models with many hidden variables, the integral (or sum) over $\mathbf{s}$ can be intractable. This problem will be solved later (see section 3.3.2) when dealing with *variational techniques*. A major problem of maximum likelihood is that, it does not take the complexity of the model into account. This problem will lead the maximum likelihood to choose more complex models, this is illustrated in figure 3.1, for a hypothetical example of density estimation using mixture of gaussian models (more about this type of models will be discussed later in this thesis.). Where the model in figure 3.1(b) will certainly have higher score, than the one in figure 3.1(c), in form of higher likelihood, i.e., $\mathcal{L}_{inc}(\mathbf{\Theta}_{\mathcal{M}_1}|\mathcal{M}_1) > \mathcal{L}_{inc}(\mathbf{\Theta}_{\mathcal{M}_2}|\mathcal{M}_2)$.
To prevent ML in choosing the more complex models, a penalty term $\mathcal{C}_p(\mathbf{\Theta})$ [35] which grows as the complexity of the model grows, should be subtracted from the likelihood function. The new score function $\mathcal{C}(\mathbf{\Theta}|\mathcal{M})$ becomes

$$\mathcal{C}(\mathbf{\Theta}|\mathcal{M}) \quad = \quad \mathcal{L}_{inc}(\mathbf{\Theta}|\mathcal{M}) - \gamma \, \mathcal{C}_p(\mathbf{\Theta}|\mathcal{M}), \tag{3.4}$$

where γ is a regularization parameter, which controls the influence of the penalty term on the total score function.
In words equation (3.4) states that, a model which provides a good fit to the training data will give high likelihood $\mathcal{L}_{inc}(\mathbf{\Theta}|\mathcal{M})$, while one which is very simple will give a small value for $\mathcal{C}_p(\mathbf{\Theta}|\mathcal{M})$. The learning then becomes a trade off between maximizing the fit to data and minimizing the complexity of the model. A well known example of regularization is the *weight decay*:

$$\mathcal{C}_p(\mathbf{\Theta}|\mathcal{M}) = \frac{1}{2}||\mathbf{\Theta}||^2, \tag{3.5}$$

which are applied to numerous linear and non-linear model e.g., *Neural networks* [5]. This regularizer favors a smoother mapping by penalizing large parameter values which are often an indication of overfitting. Note that there is only one regularization term γ for the whole model. A more complex example of regularization, which is motivated by the framework of *automatic relevance determination* (ARD)(section 4.3), is used in this thesis to control the dimension of the latent space in FA and PPCA (see section 4.3):

$$\frac{1}{2}\sum_{j=1}^{d} \gamma_j \, ||\mathcal{A}_j||^2, \tag{3.6}$$

where $\mathcal{A}_j$ ($\mathbf{A} = \{\mathcal{A}_j\}_{j=1}^{d}$) are the columns of the *factor loadings* in FA model, and d is the dimension of the observed variable $\mathbf{X}$. Note here that there is one regularizer for each column $\{\gamma_j\}_{j=1}^{d}$, where each of them tries to cancel its corresponding column if it is unnecessary to interpret the information in the data.

Model selection: The choice of a good model depends on how well that model performs on *wide range* of unseen data (i.e., how well the model generalizes) [5]. The inclusion of a regularizer, such as weight decay or ARD in (3.6) simplifies the task of model selection, since we can now choose a rather complex model and rely on the penalty term to control the effective complexity. The problem here is that the appropriate choice of the regularization parameters γ in (3.4) cannot be found by increasing $\mathcal{C}(\mathbf{\Theta}|\mathcal{M})$ using *training data* since this will of course result in a optimal choice of zeros. This means that we need to keep a part of data set (*test* set) (unseen during training) to compare the performance of the model. This procedure can itself lead to overfitting to the test set. The performance of the selected model should then be confirmed by measuring its performance on a third independent data called *validation* set. This becomes unwieldy for more complex regularizer with several regularization terms such as $\{\gamma_j\}_{j=1}^{d}$ in (3.6). The need of keeping aside part of, in the most cases, limited data set highlights another disadvantage of ML methods. An attempt to remedy this problem, techniques based on *cross-validation* are used [5], but they are computationally expensive, and are wasteful of data [7].

The frequentist has developed various criteria, often in context of linear models, to estimate the value of the generalization performance of trained models without the use of validation data. One of these criteria is *Akaike Information Criterion* (AIC) (3.8). Such criteria take the general form of the prediction error (PE) [5], which consists of the sum of two terms

$$\text{PE} = \text{trainning error} + \text{complexity term}, \tag{3.7}$$

where the complexity term represents a penalty term which grows as the number of free parameters in the model grows. Note that this is not exactly the same as the regularization in ARD (3.6) or weight decay (3.5), where the complexity was expressed by the values taken by the parameters, i.e., we could still have large number of parameters and low penalty if those parameters take on values close to zero.
Another quantity which arises from the Bayesian approach, is *the Bayesian information criterion* (BIC)[2] (see section 3.2.2), which can also be expressed as (3.7). AIC and BIC are defined as:

$$\begin{aligned} \text{AIC}(\mathbf{\Theta}_{\text{ML}}|\mathcal{M}_j) &= \mathcal{L}_{inc}(\mathbf{\Theta}_{\text{ML}}|\mathcal{M}_j) - |\mathcal{M}_j| & (3.8)\\ \text{BIC}(\mathbf{\Theta}_{\text{ML}}|\mathcal{M}_j) &= \mathcal{L}_{inc}(\mathbf{\Theta}_{\text{ML}}|\mathcal{M}_j) - \frac{|\mathcal{M}_j|}{2}\ln N, & (3.9)\end{aligned}$$

[2]BIC is the same as the negative of Minimum Description Length (MDL) BIC = −MDL

where $|\mathcal{M}_j|$ is the number of free parameters to be estimated in the model, and N is size of training set. The fact that BIC does not depend on the distribution over $\mathbf{\Theta}$, makes it suitable for use in ML where $\mathbf{\Theta}$ is deterministic.
Following equation (3.7), a certain model is better than another one, if it has a higher AIC (or BIC) value (i.e., lower prediction error PE). Both AIC and BIC have solid theoretical foundations: *Kullback-Leibler* distance in information theory (for AIC), and integrated likelihood in Bayesian theory (for BIC) (section 3.2.2). If the complexity of the true model does not increase with the size of the data set, BIC is the preferred criterion, otherwise AIC is preferred [30]. In the chapter 6, BIC will be compered to a Bayesian approach. Note that using BIC, *the good thing is*: all the data is now used to infer what we need, namely the parameter values, and the complexity control is just a part of training. But *the bad thing is*: even when using BIC or AIC it is still necessary to run the training for different models to choose the best one.

3.2 Bayesian Learning

"Bayesian inference is an approach to statistics in which all forms of uncertainty are expressed in terms of probability"

Radford M. Neal, Philosophy of Bayesian Inference
http://www.cs.toronto.edu/∼radford/res-bayes-ex.html

If you are still not convinced, here comes another one:
"I like Bayesian methods, because I know what my assumptions are, and I know what my approximations are, and I obtain error bars with a well-defined meaning, and I can marginalize over nuisance variables in order to obtain predictive distributions. It is all well-defined, mechanical and beautiful."

D.MacKay, http://www.cs.toronto.edu/∼mackay/Bayes_FAQ.html

In the previous section we discussed maximum likelihood, which attempt to find a single set of values for the parameters. By contrast, the Bayesian approach has another way of interpreting learning from data, where a probability distribution function over the parameter space is used to represent the relative degrees of belief in different values for the parameters.
In this section we consider the application of Bayesian inference techniques to Bayesian nets (see section 2.2). We will see that the regularization discussed in the last section (see equation (3.4)) can be given a natural interpretation in the Bayesian framework (3.11), and that in Bayesian methods the values of regularization terms (such as γ_j in (3.6)) can be selected using only training data, without the need to use separate training and validation data. We will also show that Bayesian model comparison embodies *Occam's razor* [12], the old principle that states a preference for simple models.
To be consistent with *levels* of learning, stated before, we will describe Bayesian learning as a 2 level learning, using the steps of *evidence framework* [10].

Level 1: Model fitting:
Assuming that one model $\mathcal{M}_m$ is true, we infer what the model's parameters might be

given the data. In the absence of any data, our belief is expressed by the prior distribution $p(\mathbf{\Theta}|\mathcal{M}_m)$. Once we observe the data $\mathbf{X}$ we can compute the *posterior probability* of the parameters $\mathbf{\Theta}$ using Bayes' rule. The process of computing the posterior $p(\mathbf{\Theta}|\mathbf{X}, \mathcal{M}_m)$ is termed *Bayesian inference*.

$$p(\mathbf{\Theta}|\mathbf{X}, \mathcal{M}_m) = \frac{\overbrace{p(\mathbf{X}|\mathbf{\Theta}, \mathcal{M}_m)}^{\text{Likelihood}}\overbrace{p(\mathbf{\Theta}|\mathcal{M}_m)}^{\text{Prior}}}{\underbrace{p(\mathbf{X}|\mathcal{M}_m)}_{\text{Evidence}}}. \tag{3.10}$$

This relationship has far reaching consequences for *Machine Learning*, and it is a prescription on how to systematically update our guess of a problem given the data observed. In words equation (3.10) states that our understanding of $\mathbf{\Theta}$ after seeing data $\mathbf{X}$ (captured by the posterior distribution over the parameters $p(\mathbf{\Theta}|\mathbf{X}, \mathcal{M}_m)$) is the previous knowledge of $\mathbf{\Theta}$ (the prior $p(\mathbf{\Theta}|\mathcal{M}_m)$) modified by how likely the observation $\mathbf{X}$ is under that previous model (likelihood $p(\mathbf{X}|\mathbf{\Theta}, \mathcal{M}_m)$). Thus the parameters that seemed plausible before, but failed in performing a well match, will now be seen as being much less likely, while the probability for values of the parameters that do fit the data well will increase.
The denominator $p(\mathbf{X}|\mathcal{M}_m)$ is a normalizing term called *marginal likelihood* or *evidence*, and ensures that the posterior behaves as a probability. Note that, with or without this term, the mean value and the parameter that maximizes the resulting distribution will still be the same. This term is sometimes omitted, but only in this level, since the evidence is crucial in the next level (model selection).

Including the hyperparameter in equation (3.10), we can interpret the cost function in (3.4) entirely in a probabilistic way, as follows:

$$\begin{aligned}
\mathcal{C}(\mathbf{\Theta}|\mathcal{M}_m) = \ln\ p(\mathbf{\Theta}|\mathbf{X}, \gamma, \mathcal{M}_m) &= \ln\ [\frac{p(\mathbf{X}|\mathbf{\Theta}, \gamma, \mathcal{M}_m)p(\mathbf{\Theta}|\gamma, \mathcal{M}_m)}{p(\mathbf{X}|\gamma, \mathcal{M}_m)}] \\
&= \ln\ p(\mathbf{X}|\mathbf{\Theta}, \gamma, \mathcal{M}_m) + \ln\ p(\mathbf{\Theta}|\gamma, \mathcal{M}_m) - \mathcal{Z} \\
&= \mathcal{L}_{inc}(\mathbf{\Theta}|\gamma, \mathcal{M}_m) - \ [-\ln\ p(\mathbf{\Theta}|\gamma, \mathcal{M}_m)] - \mathcal{Z}, \quad (3.11)
\end{aligned}$$

where $\mathcal{Z}$ is the log evidence, and is regarded here as constant (wrt. $\mathbf{\Theta}$). The weight decay regularizer in (3.5) then corresponds directly to the prior distribution for the parameters

$$\begin{aligned}
p(\mathbf{\Theta}|\gamma, \mathcal{M}_m) &= \mathcal{N}(\mathbf{\Theta}; 0, 1/\gamma) \quad (3.12) \\
&\propto \exp(-\frac{\gamma||\mathbf{\Theta}||^2}{2}) \\
\Rightarrow\ -\ln\ p(\mathbf{\Theta}|\gamma, \mathcal{M}_m) &\propto \frac{\gamma||\mathbf{\Theta}||^2}{2}.
\end{aligned}$$

Thus, maximizing the regularized cost function in (3.4) is similar to taking the *most probable* parameter value $\mathbf{\Theta}^* = \mathbf{\Theta}_{MP}$ that maximizes the posterior distribution $p(\mathbf{\Theta}|\mathbf{X}, \gamma, \mathcal{M}_m)$. This is a nice probabilistic interpretation of the cost function. This is in fact what is referred to as *maximum a priori* estimator (MAP), and $\mathbf{\Theta}_{MP} = \mathbf{\Theta}_{\text{MAP}}$, lets write the MAP for $\mathbf{\Theta}$, corresponding to weight decay:

$$\begin{aligned}
\boldsymbol{\Theta}^* = \boldsymbol{\Theta}_{\mathrm{MAP}} &= \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\ \mathcal{C}(\boldsymbol{\Theta}|\mathcal{M}_m) \\
&= \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\ \ln p(\boldsymbol{\Theta}|\mathbf{X}, \gamma, \mathcal{M}_m) \\
&= \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\ \mathcal{L}_{inc}(\boldsymbol{\Theta}|\gamma, \mathcal{M}_m) - \frac{\gamma||\boldsymbol{\Theta}||^2}{2}. \qquad (3.13)
\end{aligned}$$

This can easily be extended to the more complex case of the ARD used in this thesis, simply by assuming that the parameters $\boldsymbol{\Theta} = \{\theta_j\}_{j=1}^d$ are IID and that each parameter is a vector equal to a column in the factor loading matrix, i.e., ($\boldsymbol{\Theta} = \mathbf{A} = \{\mathcal{A}_j\}_{j=1}^d$), and where each one of those vectors has a precision (inverse of the variance) $\boldsymbol{\gamma} = \{\gamma_j\}_{j=1}^d$. Using a diagonal[3] gaussian distribution as a prior for the parameters. Equation (3.11) can be then rewritten as:

$$\begin{aligned}
\mathcal{C}(\mathbf{A}|\mathcal{M}_m) &= \mathcal{L}_{inc}(\mathbf{A}|\boldsymbol{\gamma}, \mathcal{M}_m) - [-\ln\ p(\mathbf{A}|\boldsymbol{\gamma}, \mathcal{M}_m)] - \mathcal{Z} \\
&= \mathcal{L}_{inc}(\mathbf{A}|\gamma, \mathcal{M}_m) - [-\ln\ \prod_{j=1}^{d}\ p(\mathcal{A}_j|\gamma_j, \mathcal{M}_m)] - \mathcal{Z} \\
&= \mathcal{L}_{inc}(\mathbf{A}|\gamma, \mathcal{M}_m) - [-\sum_{j=1}^{d}\ \ln\ p(\mathcal{A}_j|\gamma_j, \mathcal{M}_m)] - \mathcal{Z} \qquad (3.14) \\
&\propto \mathcal{L}_{inc}(\mathbf{A}|\gamma, \mathcal{M}_m) - \frac{1}{2}\sum_{j=1}^{d}\gamma_j\ ||\mathcal{A}_j||^2. \qquad (3.15)
\end{aligned}$$

Equation (3.15) corresponds exactly to ARD regularization of the maximum likelihood function in equation (3.6), when included in equation (3.4).

It should be noted that in order to make inferences of the true parameters $\boldsymbol{\Theta}_{true}$, based on our knowledge of the system at this **level**, the expected value $\langle\boldsymbol{\Theta}\rangle_{p(\boldsymbol{\Theta}|\mathbf{x}, \mathcal{M}_m)}$ is chosen, instead of $\boldsymbol{\Theta}_{\mathrm{MAP}}$, where the expectation is taking wrt. the posterior of the parameter of interest $p(\boldsymbol{\Theta}|\mathbf{x}, \mathcal{M}_m)$:

$$\boldsymbol{\Theta}^* = \langle\boldsymbol{\Theta}\rangle = \int \boldsymbol{\Theta}\ p(\boldsymbol{\Theta}|\mathbf{X}, \gamma, \mathcal{M}_m)\ \mathrm{d}\boldsymbol{\Theta}. \qquad (3.16)$$

This estimate happens to minimize the mean square cost function $\mathcal{J}(\boldsymbol{\Theta}_{true}, \boldsymbol{\Theta}^*)$, which penalizes the wrong estimate $\boldsymbol{\Theta}^*$, where

$$\mathcal{J}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^*) = \frac{(\boldsymbol{\Theta} - \boldsymbol{\Theta}^*)^2}{2}. \qquad (3.17)$$

The expected loss function $\mathcal{J}(\boldsymbol{\Theta}^*)$ is then found by marginalizing the above equation, wrt. $\boldsymbol{\Theta}$ as

$$\mathcal{J}(\boldsymbol{\Theta}^*) = \int \mathcal{J}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^*) p(\boldsymbol{\Theta}|\mathbf{X}, \mathcal{M}_m)\ \mathrm{d}\boldsymbol{\Theta}. \qquad (3.18)$$

[3]The diagonal choice of the covariance matrix reflects our assumption that column in the factor loading matrix are independent from each other.

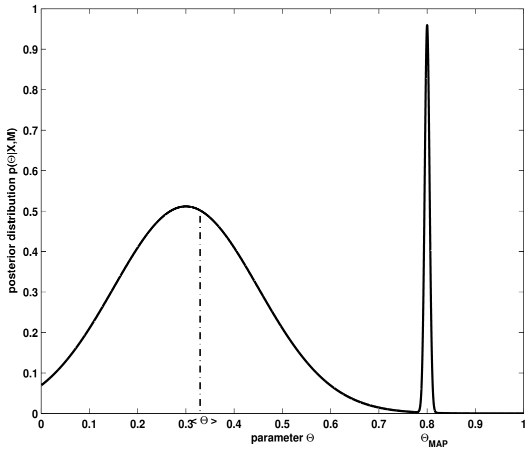


Figure 3.2: This figure illustrates a hypothetical example of a posterior distribution $p(\boldsymbol{\Theta}|\mathbf{X}, \mathcal{M}_m)$ over the parameter space $\boldsymbol{\Theta}$ of a model $\mathcal{M}_m$. Choosing the parameter that maximizes the posterior ($\boldsymbol{\Theta}_{MAP}$), in this case, will result in the highest posterior probability (equivalently the lowest training error). But because the narrow peak only contains a fraction of the total probability mass, the spike is particular for the used training set. Thus the resulting model is sensitive to this choice of parameters, and so may not explain further observations. This is referred to as overfitting. To solve this problem, we choose the average $\boldsymbol{\Theta}^* =< \boldsymbol{\Theta} >$ over all the possible parameters, weighted by their posterior probability (equation (3.16)).

Thus, minimizing this quantity gives the best estimate in mean square sense (3.17). Minimizing the above quantity requires computing its partial derivative wrt. $\boldsymbol{\Theta}^*$ and solving for zero

$$
\begin{aligned}
\frac{\partial \mathcal{J}(\boldsymbol{\Theta}_{est})}{\partial \boldsymbol{\Theta}^*} &= \int \frac{\partial \mathcal{J}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^*)}{\partial \boldsymbol{\Theta}^*} p(\boldsymbol{\Theta}|\mathbf{X}, \mathcal{M}_m)\ \mathrm{d}\boldsymbol{\Theta} \\
&= \int [\boldsymbol{\Theta} - \boldsymbol{\Theta}^*] p(\boldsymbol{\Theta}|\mathbf{X}, \mathcal{M}_m)\ \mathrm{d}\boldsymbol{\Theta} \\
&= [\langle \boldsymbol{\Theta} \rangle - \boldsymbol{\Theta}^*] = 0, \\
\Rightarrow \boldsymbol{\Theta}^* &= \langle \boldsymbol{\Theta} \rangle. \qquad (3.19)
\end{aligned}
$$

A further supported to this choice is illustrated as a hypothetical example [29] of a posterior distribution in figure 3.2. If the estimate is chosen to maximize the posterior distribution $\boldsymbol{\Theta}^* = \boldsymbol{\Theta}_{MP}$, then the model is chosen to be in a narrow peak. The problem is that the peak only contains a fraction of the total probability mass, which means that the model will explain the training data very well, but will be very sensitive to the values of the parameters, and may not explain further observations. To solve the problem we take the average over all possible values weighted by their posterior probability, as in equation (3.16). How confident we are about this estimation, can be expressed by the second moment of the posterior distribution (and can be showed as error bars).

Note that in the *fully Bayesian* approach, all variables in the Bayesian network are treated as stochastic variables and can be expressed as $\{\boldsymbol{\Theta}, \mathcal{R}\} \subseteq \mathcal{H}$, where $\boldsymbol{\Theta}$ is those hidden variables we want to reason or make prediction about their values. The remaining hidden variables $\mathcal{R}$ are regarded as nuisance variables. This holds also for the hyperparameters as γ in equation (3.11) and their ancestors (if any). The posterior of a particular parameter θ_j (e.g., γ) can then be found using the marginalization over the rest of parameters

$$p(\theta_j|\mathbf{X},\mathcal{M}_m) = \int p(\{\theta_k\}_{k=1}^K|\mathbf{X},\mathcal{M}_m) \prod_{all\ k\neq j} \mathrm{d}\theta_k. \tag{3.20}$$

All the other (nuisance) hidden variables $\mathcal{R} \neq \mathbf{\Theta}$ are integrated over (marginalized), to get the joint distribution $p(\mathbf{X},\mathbf{\Theta}|\mathcal{M}_m)$ in the nominator of equation (3.10)

$$\begin{aligned} p(\mathbf{X},\mathbf{\Theta}|\mathcal{M}_m) = p(\mathbf{X}|\mathbf{\Theta},\mathcal{M}_m)p(\mathbf{\Theta}|\mathcal{M}_m) &= \int p(\mathcal{H},\mathbf{X}|\mathcal{M}_m)\ \mathrm{d}\mathcal{R} \\ &= \int p(\{\mathbf{\Theta},\mathcal{R}\},\mathbf{X}|\mathcal{M}_m)\ \mathrm{d}\mathcal{R}, \end{aligned} \tag{3.21}$$

which coincide with the following quotation.

"Integrating over a nuisance parameter is very much like estimating the parameter from data, and then using that estimate in our equations."

G. L. Bretthorst [13], S

ince we are not implementing a fully Bayesian learning, some of the hyperparameters or their ancestors are treated as deterministic variables (rectangles in the graph), where their values are inferred using *type*-II maximum likelihood.
The real strength of Bayesian framework only emerges when going to higher levels of inference.

Level 2: Model comparison:
In fact, Bayesian inference leads to a natural method for comparing models. One can compare alternative models, $\mathcal{M}_m$ by calculating the posterior

$$p(\mathcal{M}_m|\mathbf{X}) = \frac{p(\mathbf{X}|\mathcal{M}_m)p(\mathcal{M}_m)}{p(\mathbf{X})} \tag{3.22}$$

$$\propto p(\mathbf{X}|\mathcal{M}_m)p(\mathcal{M}_m), \tag{3.23}$$

where the normalization factor $p(\mathbf{X})$ was omitted, since it is independent of the choice of the model in this level. The data dependent term $p(\mathbf{X}|\mathcal{M}_m)$ is the normalizing constant of the posterior $p(\mathbf{\Theta}|\mathbf{X},\mathcal{M}_m)$ in the previous level of inference (equation (3.10)) and it is called the *evidence* for the model $\mathcal{M}_m$. If all models have equal probability prior to comparison, the model prior $p(\mathcal{M}_m)$ will not be considered. The maximum[4] of equation (3.23), and thereby our choice of the model, is then equivalent to the maximum of the 'evidence' $p(\mathbf{X}|\mathcal{M}_m)$, which can be expressed as:

$$\text{evidence} = p(\mathbf{X}|\mathcal{M}_m) = \int p(\mathbf{X}|\mathbf{\Theta},\mathcal{M}_m)p(\mathbf{\Theta}|\mathcal{M}_m)\ \mathrm{d}\mathbf{\Theta}. \tag{3.24}$$

Because of this marginalization over the parameters, the evidence automatically penalizes too "complex" models. Complex models have more degrees of freedom so can model a wide range of data sets, therefore the probability given to any one data set, say $\mathbf{X}$,

[4] A fully Bayesian framework will take the whole posterior into account, and not only the model that maximize this distribution.

is relatively low. Models that are too simple will not be able to fit the whole data set $\mathbf{X}$ adequately so will also be given relatively small probability for $\mathbf{X}$. Only models complex enough to explain the data sufficiently, but not too complex to spread themselves too thinly, will be scored highly. This leads to a natural 'Occam's Razor' for model selection [48].
Note that asymptotically the overfitting problem is avoided simply because no parameter in the pure Bayesian approach is actually *fit* to the data (3.24).

3.2.1 Conjugate-Exponential Models

Bayesian model inference relies on the marginal likelihood, which has at its core a set of prior distributions over the parameters of each possible structure, $p(\mathbf{\Theta}|\mathcal{M}_m)$. Priors comes in several flavors, and can be roughly categorized into *subjective, objective* and *empirical* approach. To some degree all those priors are subjective, since they are based on 'our' choice. Here the emphasis is not on whether we use a prior or not, but rather on *what* knowledge (if any) is put into the prior [32].
As its name implies the subjective priors includes as much prior knowledge as possible, based either on previous experiments or on expert knowledge. A favorable class within the subjective priors are the *conjugate* priors in the *exponential family*. The priors are said to be conjugate if the posterior distribution resulting from multiplying the likelihood and prior term is of the same form as the prior, this can be mathematically expressed as

$$f(\mathbf{\Theta}|\tilde{\alpha}) = p(\mathbf{\Theta}|\mathbf{X}) \propto f(\mathbf{\Theta}|\alpha)p(\mathbf{X}|\mathbf{\Theta}), \tag{3.25}$$

where $f(\mathbf{\Theta}|\alpha)$ is some probability distribution specified by a parameter (or set of parameters) α.
In the course of the thesis we consider *conjugate-exponential models* (CEM), CEMare models that satisfy the following two conditions:

Condition (1). *The incomplete data likelihood $\mathcal{L}_{inc}(\mathbf{\Theta})$ is in the exponential family*:

$$\mathcal{L}_{inc}(\mathbf{\Theta}) \quad = \quad p(\mathbf{x}_i|\mathbf{\Theta}) = g(\mathbf{\Theta})f(\mathbf{x}_i)\exp\{\phi(\mathbf{\Theta})^\top \mathbf{u}(\mathbf{x}_i)\}, \tag{3.26}$$

where $g(\mathbf{\Theta})$ is a normalization constant, ϕ is the vector of the so-called *natural parameters*, and $\mathbf{u}$ and f are functions defining the exponential family.

Condition (2). *The parameter prior is conjugate to the $\mathcal{L}_{inc}(\mathbf{\Theta})$:*

$$p(\mathbf{\Theta}|\eta,\nu) \quad = \quad h(\eta,\nu)g(\mathbf{\Theta})^\eta \exp\{\phi(\mathbf{\Theta})^\top \nu\}, \tag{3.27}$$

where η and ν are *hyperparameters* of the prior. Note that $g(\mathbf{\Theta})$ and $\phi(\mathbf{\Theta})$ are the same as in condition (1). Condition (2) usually implies condition (1). Apart from some particular cases, the exponential family are the only classes of distributions with fixed number of sufficient statistics, therefore conjugate priors exist only for this family.
In Bayesian inference we are interest in determining the posterior over the parameters $p(\mathbf{\Theta}|\mathbf{X})$. Using equations (3.26) and (3.27) into equation (3.25) we get

$$p(\boldsymbol{\Theta}|\mathbf{X}) \propto p(\boldsymbol{\Theta}|\eta,\nu)p(\mathbf{X}|\boldsymbol{\Theta}) \propto p(\boldsymbol{\Theta}|\tilde{\eta},\tilde{\nu}) \tag{3.28}$$

where

$$\tilde{\eta} = \eta + N$$
$$\tilde{\nu} = \nu + \sum_{i=1}^{N} \mathbf{u}(\mathbf{x}_i),$$

where $\tilde{\nu}$ and $\tilde{\eta}$ are the *updated* parameters of the posterior distribution which has the same functional form as the prior.

3.2.2 Approximation Methods

In practice, in most cases, it is computationally intractable to perform exact inference. Bayesian inference requires calculating marginals over hidden variables. It can be seen by taking the evidence in equation (3.10), and equation (3.21)

$$\begin{aligned} p(\mathbf{X}|\mathcal{M}_m) &= \int p(\boldsymbol{\Theta},\mathbf{X}|\mathcal{M}_m)\ \mathrm{d}\boldsymbol{\Theta} \\ &= \int\int p(\{\boldsymbol{\Theta},\mathcal{R}\},\mathbf{X}|\mathcal{M}_m)\ \mathrm{d}\mathcal{R}\ \mathrm{d}\boldsymbol{\Theta}. \end{aligned} \tag{3.29}$$

These integrals are typically high-dimensional, non-linear and in many cases non-analytic. The intractability of exact inference in both discrete and continuous models has lead to the development of a number of *approximate inference* techniques. As mentioned in the introduction, these approximation methods can be partitioned in two groups [14]:

Samplings approximations:

Instead of trying to determine the posterior, sampling methods are stochastic approaches that attempt to obtain a number of samples from that posterior. The most widely used samplings techniques lie in the family of *Markov Chain Monte Carlo* (MCMC) methods. Those methods have been applied to several machine learning problems [36, 2]. MCMC are computationally expensive, and very slow [17, 32].

Deterministic approximations:

This family includes among others, the previously discussed maximum likelihood, *maximum a posteriori* (MAP), *laplace methods*, and *variational methods*.

3.2.3 Laplace approximation and BIC

Laplace approximation to the evidence $p(\mathbf{X},\mathcal{M}_m)$ makes a local Gaussian approximation around a MAP parameter estimate, which after some mathematical manipulations [12, 32] can be written as:

$$p(\mathbf{X}|\mathcal{M}_m)_{Laplace} = p(\mathbf{\Theta}_{\mathrm{MAP}}, \mathbf{X}|\mathcal{M}_m)p(\mathbf{\Theta}_{\mathrm{MAP}})|2\pi\mathrm{H}^{-1}|^{1/2} \tag{3.30}$$

$$\Rightarrow \ln p(\mathbf{X}|\mathcal{M}_m)_{Laplace} = \ln p(\mathbf{\Theta}_{\mathrm{MAP}}, \mathbf{X}|\mathcal{M}_m) - \frac{1}{2}\ln|\mathrm{H}| + \mathcal{O}(1), \tag{3.31}$$

where H is the hessian of the log posterior evaluated at $\mathbf{\Theta}_{\mathrm{MAP}}$ [32], and $\mathcal{O}(1)$ is all terms independent on N. Assuming that the prior is non-zero at $\mathbf{\Theta}_{\mathrm{MAP}}$, in the limit of large N, the above equation becomes the BIC score:

$$\ln p(\mathbf{X}|\mathcal{M}_m)_{\mathrm{BIC}} = \ln p(\mathbf{\Theta}_{\mathrm{MAP}}, \mathbf{X}|\mathcal{M}_m) - \frac{d_{\mathcal{M}_m}}{2}\ln N, \tag{3.32}$$

where $d_{\mathcal{M}_m}$ is the dimension of the model $\mathcal{M}_m$. There are two interesting points in the BIC expression, first it is independent on the prior $p(\mathbf{\Theta}|\mathcal{M}_m)$, this feature is mostly interesting for point estimate methods such as ML and can to some degree be interpreted as the prediction error (PE) in equation (3.7), therefore it was written in equation (3.9). The second interesting point is that BIC is invariant to reparameterisation of the model [32], and is appealing in Bayesian approach, since this should fall out of an exact Bayesian treatment in any case.

However, the laplace approximation (3.30) can be poor for small data set (for which, in principle, the advantages of Bayesian integration over ML are largest.) and the gaussian approximation requires computing or approximating the Hessian at the MAP estimate, which can be computationally costly.

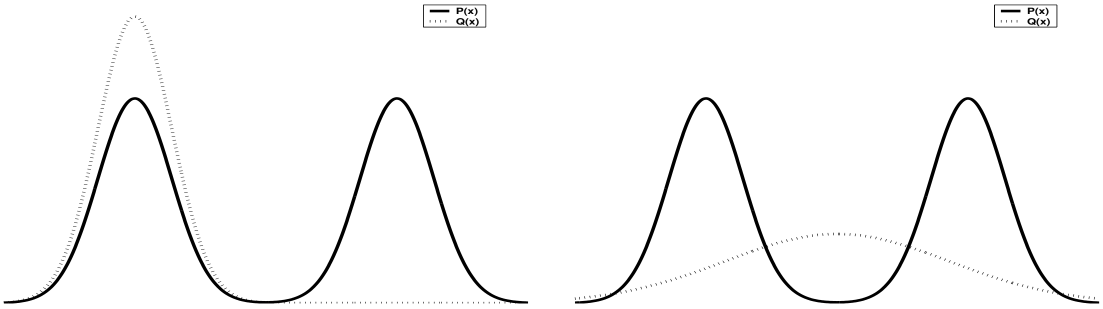
My focus in this thesis is on the *variational methods* that approximate expectations of functions under the posterior.

3.3 Variational Inference

Unlike sampling methods, variational inference is a deterministic approximate method [14] that attempt to optimize directly the accuracy of the approximated posterior distribution. The aim of variational approximation can be roughly explained as a method where we convert a complex problem into a simpler problem by decoupling degrees of freedom in the original problem. Variational methods can be applied, to approximate exact distributions by simpler *variational distributions*, which has loose dependency structure, by using an assumption that, certain hidden variables may become approximately independent when conditioned on the observed data. The *variational inference* is therefore referred to the case, where these approximated distributions are the posterior distributions of hidden variables $\mathcal{H}$:

$$p(\mathcal{H}|\mathbf{X}) \approx \mathcal{Q}(\mathcal{H}). \tag{3.33}$$

The inference is then performed by minimizing the difference between *variational distribution* $\mathcal{Q}_{\mathcal{H}}$ and the true posterior. The difference is measured in terms of the dissimilarity function $d(\mathcal{Q}, P)$. A choice of dissimilarity function where the minimization is tractable is the *Kullback-Leibler divergence*.



(a) KL($\mathcal{Q}||P$) (b) KL($P||\mathcal{Q}$)

Figure 3.3: This figure illustrates the difference between minimizing KL($\mathcal{Q}||P$) (a) or KL($P||\mathcal{Q}$) (b). This difference is due to the asymmetry of the $\mathcal{Q}$ divergence. Suppose P is bimodal distribution (solid line), and we attempt to approximate it using a unimodal distribution $\mathcal{Q}$ (dashed line), that in: (a) minimizes KL($\mathcal{Q}||P$), this will give a $\mathcal{Q}$ distribution with almost all probability mass in one mode of P and neglecting the other mode. And in: (b) minimizes KL($P||\mathcal{Q}$) that will give a $\mathcal{Q}$ that covers both modes, but which also places a high probability between the modes, where P is negligible. Thus a care should be taken when minimizing the KL between the posterior and the variational distribution.

3.3.1 Kullback-Leibler divergence

The Kullback-Leibler (KL) divergence is an entropy-like measure, defined as

$$\mathrm{KL}(\mathcal{Q}||P) = \int \mathcal{Q}(\mathbf{x}) \ln \frac{\mathcal{Q}(\mathbf{x})}{P(\mathbf{x})} \, \mathrm{d}\mathbf{x}. \tag{3.34}$$

The KL divergence has the property of being zero when $\mathcal{Q} = P$ and positive otherwise, moreover it is not symmetric and thus

$$\mathrm{KL}(\mathcal{Q}||P) \neq \mathrm{KL}(P||\mathcal{Q}).$$

A simple illustration in figure 3.3 shows the difference between minimizing the above two divergences. Where the first one attempts to cover a part of the distribution, and the second one tries to cover both modes.

3.3.2 Variational Maximum Likelihood

In section 3.1 we discussed the problem of direct maximization of the log-likelihood function in the presence of hidden variables (incomplete data log-likelihood $\mathcal{L}_{inc}(\boldsymbol{\Theta})$, equation (3.3)). The function is rewritten here for convenience

$$\mathcal{L}_{inc}(\boldsymbol{\Theta}) = \sum_i \ln p(\mathbf{x}_i|\boldsymbol{\Theta}) = \sum_i \ln \int p(\mathbf{x}_i, \mathbf{S}|\boldsymbol{\Theta}) \, \mathrm{d}\mathbf{S}, \tag{3.35}$$

where we have assumed the data is IID, the model $\mathcal{M}$ is omitted from the expression for simplicity. Here one should remember that in ML, $\boldsymbol{\Theta}$ is a deterministic variable (no distribution over it), whereas the hidden state $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^N$ are stochastic variables (continuous or discrete).
We can simplify the problem of maximizing $\mathcal{L}_{inc}$ wrt. $\boldsymbol{\Theta}$ by making use of the following

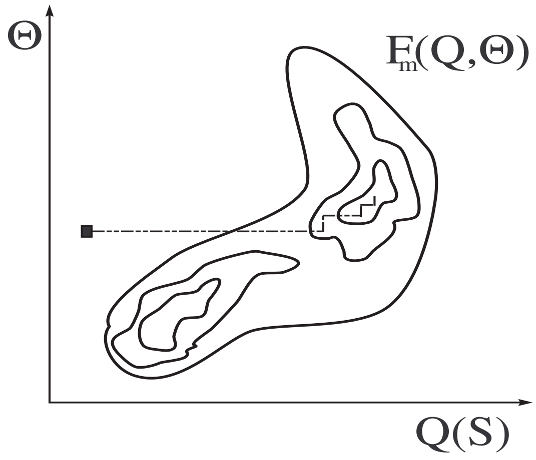


Figure 3.4: This figure shows that EM is a coordinate ascent algorithm in $\mathcal{F}$.

insight [40],[42]: Any distribution $\mathcal{Q}(\mathbf{S})$ over the hidden variables defines a *lower bound* on $\mathcal{L}_{inc}(\mathbf{\Theta})$. In fact for each data point $\mathbf{x}_i$ we use a distinct distribution $\mathcal{Q}_{\mathbf{s}_i}$ over the hidden variables to get the lower bound. This can be shown by a simple manipulation of equation (3.35)

$$\begin{aligned}
\mathcal{L}_{inc}(\mathbf{\Theta}) \;=\; \sum_i \ln p(\mathbf{x}_i|\mathbf{\Theta}) &= \sum_i \int \ln \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)\frac{p(\mathbf{x}_i,\mathbf{s}_i|\mathbf{\Theta})}{\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)}\,\mathrm{d}\mathbf{s}_i && (3.36)\\
&\geq \sum_i \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)\ln\frac{p(\mathbf{x}_i,\mathbf{s}_i|\mathbf{\Theta})}{\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)}\,\mathrm{d}\mathbf{s}_i && (3.37)\\
&= \mathcal{F}(\{\mathcal{Q}_{\mathbf{s}_i}\}_{i=1}^N,\mathbf{\Theta}), && (3.38)
\end{aligned}$$

where the last equality follows from the fact that the observed data is IID. This assumption results also in the following important independency

$$\mathcal{Q}_{\mathbf{S}}(\mathbf{S}) = \prod_{i=1}^{N}\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i). \tag{3.39}$$

A more detailed derivation of the above important expression can be found in Appendix A (A.1). This inequality is referred to as *Jensen's inequality* and is based on the concavity property of the logarithmic function. Defining the *complete data log-likelihood* to be $\mathcal{L}_c(\mathbf{\Theta}) = \ln p(\mathbf{X},\mathbf{S}|\mathbf{\Theta})$[5], the lower bound $\mathcal{F} \geq \mathcal{L}_{inc}(\mathbf{\Theta})$ is the negative of a quantity known in statistical physics as the *free energy*:

$$\begin{aligned}
\mathcal{F}(\{\mathcal{Q}_{\mathbf{s}_i}\}_{i=1}^N,\mathbf{\Theta}) &= \sum_i \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)\ln\frac{p(\mathbf{x}_i,\mathbf{s}_i|\mathbf{\Theta})}{\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)}\,\mathrm{d}\mathbf{s}_i && (3.40)\\
&= \sum_i \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)\ln p(\mathbf{x}_i,\mathbf{s}_i|\mathbf{\Theta})\,\mathrm{d}\mathbf{s}_i - \sum_i \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)\ln \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)\,\mathrm{d}\mathbf{s}_i\\
&= <\mathcal{L}_c(\mathbf{\Theta})>_{\mathcal{Q}_{\mathbf{S}}} - \sum_i \mathbb{H}(\mathcal{Q}(\mathbf{s}_i)), && (3.41)
\end{aligned}$$

or equivalently

[5] Some times its negative $-\mathcal{L}_c(\mathbf{\Theta})$ will be referred to, in the thesis, as the *energy* of the global configuration $(\mathbf{X},\mathbf{S})$.

$$\begin{aligned}\mathcal{F}(\{\mathcal{Q}_{\mathbf{s}_i}\}_{i=1}^N, \mathbf{\Theta}) &= \sum_i \int \mathcal{Q}(\mathbf{s}_i) \ln p(\mathbf{x}_i|\mathbf{s}_i, \mathbf{\Theta}) \, d\mathbf{s}_i - \sum_i \int \mathcal{Q}(\mathbf{s}_i) \ln \frac{\mathcal{Q}(\mathbf{s}_i)}{p(\mathbf{s}_i)} \, d\mathbf{s}_i \\ &= < \ln p(\mathbf{x}_i|\mathbf{s}_i, \mathbf{\Theta}) >_{\mathcal{Q}_\mathbf{S}} - \sum_i \mathrm{KL}(\mathcal{Q}(\mathbf{s}_i)||p(\mathbf{s}_i)), \end{aligned} \tag{3.42}$$

where $< \mathcal{L}_c(\mathbf{\Theta}) >_{\mathcal{Q}_\mathbf{S}}$ is the expected energy under $\mathcal{Q}_\mathbf{S}$, and $\mathbb{H}(\mathcal{Q}(\mathbf{s}_i)) = \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \, d\mathbf{s}_i$ is the entropy of $\mathcal{Q}(\mathbf{s}_i)$. The *Expectation Maximization* (EM) [23, 44, 40, 5] alternates between maximizing $\mathcal{F}$ wrt. $\mathcal{Q}_\mathbf{S}$ and $\mathbf{\Theta}$ respectively , while holding the other fixed, which is coordinate ascent in the function space of variational distribution of hidden variables $\mathcal{Q}_\mathbf{S}(\mathbf{S})$ and the deterministic parameters $\mathbf{\Theta}$, this is illustrated in figure 3.4. The EM steps can be written as:

$$\textbf{E step:} \quad \mathcal{Q}_{\mathbf{s}_i}^{k+1} \leftarrow \underset{\mathcal{Q}_{\mathbf{s}_i}}{\mathrm{argmax}} \ \mathcal{F}(\mathcal{Q}, \mathbf{\Theta}^k), \quad \forall i \tag{3.43}$$

$$\begin{aligned} \textbf{M step:} \quad \mathbf{\Theta}^{k+1} &\leftarrow \underset{\mathbf{\Theta}}{\mathrm{argmax}} \ \mathcal{F}(\mathcal{Q}^{k+1}, \mathbf{\Theta}), \\ \Rightarrow \quad \mathbf{\Theta}^{k+1} &\leftarrow \underset{\mathbf{\Theta}}{\mathrm{argmax}} \ < \mathcal{L}_c(\mathbf{\Theta}) >_{\mathcal{Q}_\mathbf{S}}, \qquad (3.44) \\ \Rightarrow \quad \mathbf{\Theta}^{k+1} &\leftarrow \underset{\mathbf{\Theta}}{\mathrm{argmax}} \ < \ln p(\mathbf{X}|\mathbf{S}, \mathbf{\Theta}) >_{\mathcal{Q}_\mathbf{S}} . \qquad (3.45) \end{aligned}$$

Thus the **M step** will correspond to either maximizing[6] the expected log-likelihood $< \mathcal{L}_c(\mathbf{\Theta}) >_{\mathcal{Q}_\mathbf{S}}$ (3.44), since the entropy in (3.41) is independent on $\mathbf{\Theta}$, or maximizing $< \ln p(\mathbf{X}|\mathbf{S}, \mathbf{\Theta}) >_{\mathcal{Q}_\mathbf{S}}$, since the KL divergence in (3.42) is also independent on $\mathbf{\Theta}$. The maximum of the E-step is reached when $\mathcal{Q}_{\mathbf{s}_i}^{k+1} = P(\mathbf{S}|\mathbf{x}_i, \mathbf{\Theta})$, at that point the bound becomes an equality: $\mathcal{F}(\mathcal{Q}_{\mathbf{s}_i}^{k+1}, \mathbf{\Theta}^k) = \mathcal{L}_{inc}(\mathbf{\Theta}^k)$, this can easily be deduced from equation (3.40) by replacing $\mathcal{Q}(\mathbf{s}_i)$ with $P(\mathbf{s}_i|\mathbf{x}_i, \mathbf{\Theta})$

$$\mathcal{F}(\{\mathcal{Q}_{\mathbf{s}_i}\}_{i=1}^N, \mathbf{\Theta}) = \sum_i \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{p(\mathbf{x}_i, \mathbf{s}_i|\mathbf{\Theta})}{P(\mathbf{s}_i|\mathbf{x}_i, \mathbf{\Theta})} \, d\mathbf{s}_i \tag{3.46}$$

$$= \sum_i \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln p(\mathbf{x}_i|\mathbf{\Theta}) \, d\mathbf{s}_i \tag{3.47}$$

$$= \sum_i \ln p(\mathbf{x}_i|\mathbf{\Theta}) \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \, d\mathbf{s}_i \tag{3.48}$$

$$= \sum_i \ln p(\mathbf{x}_i|\mathbf{\Theta}) = \mathcal{L}_{inc}(\mathbf{\Theta}). \tag{3.49}$$

Since $\mathcal{F} = \mathcal{L}_{inc}(\mathbf{\Theta})$ at the beginning of each M-step, and since $\mathbf{\Theta}$ do not change in the E-step, we are guaranteed not to decrease the likelihood after each combined EM step. Maximizing $\mathcal{F}$ in the E-step is equivalent to minimizing $\mathrm{KL}(\mathcal{Q}(\mathbf{S})||P(\mathbf{X}, \mathbf{S}|\mathbf{\Theta}))$ (see (3.40) and (3.34)).

By constraining $\mathcal{Q}$ to be for instance *factorized* $\mathcal{Q} = \prod_i \mathcal{Q}_{\mathbf{s}_i}$, the E-step can be simplified, and $\mathcal{F}$ will still be optimized as a functional of constrained distributions $\mathcal{Q}$ using *calculus of variations* [5]. This is the key step of variational approximations. The E-step of this *variational* EM consist of a sub loop where $\mathcal{Q}_{\mathbf{s}_i}$ is optimized. This can be done by taking

[6]By maximizing a function $f(x, y)$ wrt. x (i.e., $x \leftarrow \mathrm{argmax}_x f(x, y)$), we mean that, one should take the partial derivative of f wrt. x and equate it to zero $\frac{\partial}{\partial x} f(x, y) = 0$, and then solve for x.

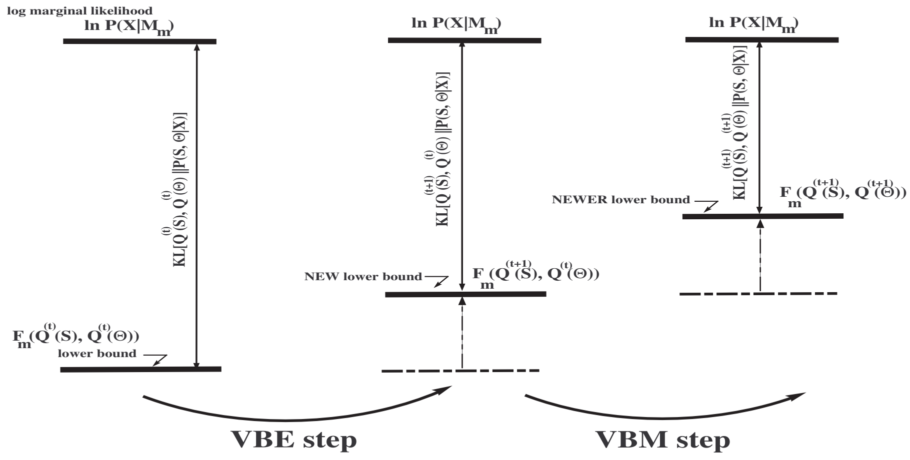


Figure 3.5: The variational Bayesian EM (VBEM) algorithm. In the VBE step, the variational posterior over hidden variables $\mathcal{Q}_{\mathbf{S}}(\mathbf{S})$ is set according to equation (3.56). In the VBM, the variational posterior over parameters is set according to equation (3.57). Each step is guaranteed to increase (or leave unchanged) the lower bound on the marginal likelihood. Note that the exact marginal likelihood is a fixed quantity, for each model, and is not affected by the VBEMsteps-it is only the lower bound that changes.

the derivative wrt. the parameters of $\mathcal{Q}_{\mathbf{s}_i}$.

Note that ML learning of model parameters, is performed by maximizing the *total* or incomplete-data log likelihood $\mathcal{L}_{inc}(\mathbf{\Theta}) = \ln p(\mathbf{X}|\mathbf{\Theta})$ (3.35), but as a consequences of using the EM algorithm we end up maximizing the complete-data log-likelihood

$$< \mathcal{L}_c(\mathbf{\Theta}) >_{\mathcal{Q}} = < \ln p(\mathbf{X}, \mathbf{S}|\mathbf{\Theta}) >_{\mathcal{Q}},$$

as seen in equation (3.41). Thus it appears that we are maximizing the incorrect quantity, but doing so is in fact guaranteed to increase (or keep unchanged) the quantity of interest, as shown in figure 3.4.

3.3.3 Variational Bayes

Contrary to ML, Bayesian learning avoids the problems of overfitting and can be used to model selection (section 3.2). But because of computations such as (3.29), the problem is computationally intractable. In this section we deal with a variational approximation to the integrals required for Bayesian learning. The process is almost the same as in EM, where the *incomplete data log-likelihood* of the deterministic variables $\mathbf{\Theta}$ was lower bounded. Except that in *variational Bayes* (VB) it is the model *evidence*[7] $p(\mathbf{X}|\mathcal{M})$ that is r bounded, and the set of parameters $\mathbf{\Theta}$ is regarded as stochastic variables, where their prior distributions $p(\mathbf{\Theta}|\mathcal{M})$ are included in the process. Thus the *variational distribution* to infer here is one that approximate the joint posterior distribution of $\mathbf{S}$ and $\mathbf{\Theta}$: $\mathcal{Q}_{\{\mathbf{\Theta},\mathbf{S}\}} \approx p(\mathbf{\Theta}, \mathbf{S}|\mathbf{X}, \mathcal{M})$. The assumption that the variational distribution are factorized (separable) is usually used, but for the moment only between $\mathcal{Q}_{\mathbf{\Theta}}$[8] and $\mathcal{Q}_{\mathbf{S}}$

$$\mathcal{Q}_{\{\mathbf{\Theta},\mathbf{S}\}} \approx \mathcal{Q}_{\mathbf{\Theta}}\ \mathcal{Q}_{\mathbf{S}}.$$

[7] The evidence is still regarded as the incomplete data likelihood since it only depends on the observed data and deterministic hidden variables which is the model $\mathcal{M}$ in this case.

[8] For simplicity, trough out the report, when writing $\mathcal{Q}_\alpha$, the argument is explicitly α, i.e. $\mathcal{Q}_\alpha = \mathcal{Q}(\alpha)$, except other argument is specified.

The evidence in equation (3.24) can then be rewritten as:

$$
\begin{aligned}
\mathcal{L}_{inc}(\mathcal{M}) = \ln p(\mathbf{X}|\mathcal{M}_m) &= \int \mathcal{Q}_{\mathbf{\Theta}}\mathcal{Q}_{\mathbf{S}} \ln p(\mathbf{X}|\mathcal{M}_m)\ \mathrm{d}\mathbf{S}\mathrm{d}\mathbf{\Theta} \\
&= \int \mathcal{Q}_{\mathbf{\Theta}}\mathcal{Q}_{\mathbf{S}} \ln[\frac{p(\mathbf{X},\mathbf{S},\mathbf{\Theta}|\mathcal{M}_m)}{\mathcal{Q}_{\mathbf{\Theta}}\mathcal{Q}_{\mathbf{S}}}\frac{\mathcal{Q}_{\mathbf{\Theta}}\mathcal{Q}_{\mathbf{S}}}{p(\mathbf{S},\mathbf{\Theta}|\mathbf{X},\mathcal{M}_m)}]\mathrm{d}\mathbf{S}\mathrm{d}\mathbf{\Theta} \\
&= \int \mathcal{Q}_{\mathbf{\Theta}}\mathcal{Q}_{\mathbf{S}} [\ln\frac{p(\mathbf{X},\mathbf{S},\mathbf{\Theta}|\mathcal{M}_m)}{\mathcal{Q}_{\mathbf{\Theta}}\mathcal{Q}_{\mathbf{S}}} + \ln\frac{\mathcal{Q}_{\mathbf{\Theta}}\mathcal{Q}_{\mathbf{S}}}{p(\mathbf{S},\mathbf{\Theta}|\mathbf{X},\mathcal{M}_m)}]\mathrm{d}\mathbf{S}\mathrm{d}\mathbf{\Theta} \\
&= \mathcal{F}_m(\mathcal{Q}_{\mathbf{\Theta}},\mathcal{Q}_{\mathbf{S}}) + \mathrm{KL}(\mathcal{Q}_{\mathbf{\Theta}}\mathcal{Q}_{\mathbf{S}}||p(\mathbf{S},\mathbf{\Theta}|\mathbf{X},\mathcal{M}_m)) && (3.50) \\
&\geq \mathcal{F}_m(\mathcal{Q}_{\mathbf{\Theta}},\mathcal{Q}_{\mathbf{S}}) && (3.51) \\
&= \mathcal{F}_m(\mathcal{Q}_{\mathbf{\Theta}},\{\mathcal{Q}_{\mathbf{s}_i}\}_{i=1}^N), && (3.52)
\end{aligned}
$$

where in (3.51) we used Jensen's inequality (this can also be explained by the fact that KL ≥ 0 (section 3.3.1)), and in the last equality follows from the fact that the observed data is IID. Note the similarity between equations (3.38) and (3.51). While we maximize the former wrt. hidden states *distributions* and the parameters, the latter is maximized wrt. hidden states *distributions* and the parameters *distributions*. Due to the factorization of the hidden variables $\mathbf{\Theta}$ and $\mathbf{S}$ the variational Bayesian algorithm can be implemented as the EM algorithm , this is in fact called *variational Bayesian* EM where we iteratively maximizes $\mathcal{F}_m$ wrt. the distributions, $\mathcal{Q}_{\mathbf{X}}$ and $\mathcal{Q}_{\mathbf{S}}$, which is coordinate ascent in the function space of variational distributions, this is similar to the illustration in figure 3.4, except this time we have a distribution $\mathcal{Q}_{\mathbf{\Theta}}(\mathbf{\Theta})$ in stead of a deterministic parameter $\mathbf{\Theta}$. From equation (3.50) we can see that maximizing $\mathcal{F}_m$ is equivalent to minimizing the KL divergence between $\mathcal{Q}_{\mathbf{\Theta}}\mathcal{Q}_{\mathbf{S}}$ and the joint posterior $p(\mathbf{S},\mathbf{\Theta}|\mathbf{X},\mathcal{M}_m)$. To infer the variational distributions lets first right down the lower bound expression

$$
\mathcal{F}_m(\mathcal{Q}_{\mathbf{S},\mathbf{\Theta}}) = \int \mathcal{Q}_{\mathbf{S}}(\mathbf{S})\mathcal{Q}_{\mathbf{\Theta}}(\mathbf{\Theta}) \ln\frac{p(\mathbf{X},\mathbf{S},\mathbf{\Theta}|\mathcal{M}_m)}{\mathcal{Q}_{\mathbf{S}}(\mathbf{S})\mathcal{Q}_{\mathbf{\Theta}}(\mathbf{\Theta})}\mathrm{d}\mathbf{\Theta}\mathrm{d}\mathbf{S}. \tag{3.53}
$$

Using the calculus of variations we can prove that the solution for each of the individual $\mathcal{Q}_{\mathcal{H}_k}$ distributions that maximizes the functional $\mathcal{F}$ is of the form:

$$
\mathcal{Q}_{\mathcal{H}_k}(\mathcal{H}_k) = \frac{\exp < \ln p(\mathbf{X},\mathcal{H}|\mathcal{M}_m) >_{\mathcal{Q}_{j\neq k}}}{\int \exp < \ln p(\mathbf{X},\mathcal{H}|\mathcal{M}_m) >_{\mathcal{Q}_{j\neq k}}\ \mathrm{d}\mathcal{H}_k}, \tag{3.54}
$$

or equivalently

$$
\ln \mathcal{Q}_{\mathcal{H}_k}(\mathcal{H}_k) =< \ln p(\mathbf{X},\mathcal{H}|\mathcal{M}_m) >_{\mathcal{Q}_{j\neq k}} + \mathrm{k}, \tag{3.55}
$$

where $\mathcal{H} = \{\mathcal{H}_k\}_{k=1}^K$ represent all hidden variables in the model: $\{\mathbf{\Theta},\mathbf{S}\} \subseteq \mathcal{H}$, this generale expression will become useful later when we assume further factorization for some of the variables in $\mathbf{\Theta}$.

Due to the constraint made by the Lagrange multipliers in the calculus of variations []. All the elements resulting from equation (3.54) (including those from the nominator), that are independent on $\mathcal{H}_k$, represent the inverse of the normalization factor $\mathcal{Z}_{\mathcal{H}_k}$, which ensures that $\mathcal{Q}_{\mathcal{H}_k}(\mathcal{H}_k)$ behaves as a probability distribution, this can be shortly seen in e.g. VBE-step.

Using equations (3.39) and (3.54) we can write down the VBEM (similar to EM) algorithm as follows:

VBE step:

$$
\begin{aligned}
\mathcal{Q}^{(t+1)}_{\mathbf{s}_i}(\mathbf{s}_i) &= \frac{\exp < \ln p(\mathbf{X},\mathbf{S},\mathbf{\Theta}|\mathcal{M}_m) >_{\mathcal{Q}^{(t)}_{\mathbf{\Theta}}}}{\int \exp < \ln p(\mathbf{X},\mathbf{S},\mathbf{\Theta}|\mathcal{M}_m) >_{\mathcal{Q}^{(t)}_{\mathbf{\Theta}}} \; \mathrm{d}\mathbf{s}_i} \\
&= \frac{\exp < \sum_n \ln p(\mathbf{x}_n,\mathbf{s}_n|\mathbf{\Theta},\mathcal{M}_m) + \ln p(\mathbf{\Theta}|\mathcal{M}_m) >_{\mathcal{Q}^{(t)}_{\mathbf{\Theta}}}}{\int \exp < \ln p(\mathbf{X},\mathbf{S},\mathbf{\Theta}|\mathcal{M}_m) >_{\mathcal{Q}^{(t)}_{\mathbf{\Theta}}} \mathrm{d}\mathbf{s}_i} \\
&= \exp < \ln p(\mathbf{x}_i,\mathbf{s}_i|\mathbf{\Theta},\mathcal{M}_m) >_{\mathcal{Q}^{(t)}_{\mathbf{\Theta}}} \\
&\quad \frac{\exp < \sum_{n\neq i} \ln p(\mathbf{x}_n,\mathbf{s}_n|\mathbf{\Theta},\mathcal{M}_m) + \ln p(\mathbf{\Theta}|\mathcal{M}_m) >_{\mathcal{Q}^{(t)}_{\mathbf{\Theta}}}}{\int \exp < \ln p(\mathbf{X},\mathbf{S},\mathbf{\Theta}|\mathcal{M}_m) >_{\mathcal{Q}^{(t)}_{\mathbf{\Theta}}} \mathrm{d}\mathbf{s}_i} \\
\mathcal{Q}^{(t+1)}_{\mathbf{s}_i}(\mathbf{s}_i) &= \boldsymbol{\frac{\exp < \ln p(\mathbf{x}_i,\mathbf{s}_i|\mathbf{\Theta},\mathcal{M}_m) >_{\mathcal{Q}^{(t)}_{\mathbf{\Theta}}}}{\mathcal{Z}_{\mathbf{s}_i}}} \qquad \forall i
\end{aligned} \tag{3.56}
$$

and similarly

VBM step:

$$
\begin{aligned}
\mathcal{Q}^{(t+1)}_{\mathbf{\Theta}}(\mathbf{\Theta}) &= \frac{\exp < \ln p(\mathbf{X},\mathbf{S},\mathbf{\Theta}|\mathcal{M}_m) >_{\mathcal{Q}^{(t+1)}_{\mathbf{S}}}}{\int \exp < \ln p(\mathbf{X},\mathbf{S},\mathbf{\Theta}|\mathcal{M}_m) >_{\mathcal{Q}_{\mathbf{S}}\ (t+1)} \; \mathrm{d}\mathbf{\Theta}} \\
&= \frac{\exp < \ln p(\mathbf{X},\mathbf{S}|\mathbf{\Theta},\mathcal{M}_m) + \ln p(\mathbf{\Theta}|\mathcal{M}_m) >_{\mathcal{Q}^{(t+1)}_{\mathbf{S}}}}{\mathcal{Z}_{\mathbf{\Theta}}} \\
\mathcal{Q}^{(t+1)}_{\mathbf{\Theta}}(\mathbf{\Theta}) &= \boldsymbol{\frac{p(\mathbf{\Theta}|\mathcal{M}_m) \;\; \exp < \ln p(\mathbf{X},\mathbf{S}|\mathbf{\Theta},\mathcal{M}_m) >_{\mathcal{Q}^{(t+1)}_{\mathbf{S}}}}{\mathcal{Z}_{\mathbf{\Theta}}}}
\end{aligned} \tag{3.57}
$$

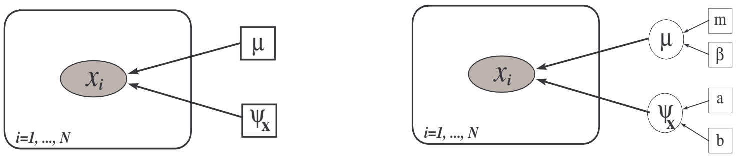
where in both steps we used the properties of the expectation operator $< \cdot >$[9] and the logarithmic function[10]. The VBEMsteps are illustrated in figure 3.5

[9] $\langle x+y\rangle = \langle x\rangle + \langle y\rangle$

[10] $\ln(x \cdot y) = \ln x + \ln y$

CHAPTER 4

Linear Latent Variable Models



(a) Maximum-likelihood model. (b) Bayesian approach model.

Figure 4.1: The Bayesian network for a probabilistic model represents a set of N observed data points with a Gaussian distribution of mean μ and precision $\mathbf{\Psi}_X$. The plate (section 2.2) indicates that the contained node and its connected edges are duplicated N times, and that the observed variable x is assumed IID. (a) represents the MLmodel, where the hidden parameters $\mathbf{\Theta} = \{\mu, \psi_x\}$ are deterministic variables; whereas (b) is the Bayesian approach, where both parameters are regarded as stochastic variables.

In this chapter we will present the different models used through out this report, we will also discuss their advantages and drawbacks, based on their performances, mainly in estimating the probability density of some given toy data. A more detailed comparison based on some real world data or more complex toy data, will be discussed later in chapter 6.
The models are presented as Bayesian nets (section 2.2), and they will be learned using both ML- and VB-techniques.
In the next section I will discuss density estimation using a single Gaussian, and based on its failure, I will give the reason why one should use better models, namely *Latent variable models*.

4.1 Density modelling using a single Gaussian

A simple and widely used example of density function is the *Gaussian* or *normal* distribution defined on $\mathbf{x} \in \mathbb{R}^d$

$$p(\mathbf{X}|\mathbf{\Theta}) = \mathcal{N}(\mathbf{X}|\mu, \Psi_x^{-1}) = \frac{|\Psi_x|^{1/2}}{(2\pi)^{d/2}} \exp[-\frac{1}{2}(\mathbf{X}-\mu)^\top \Psi_x (\mathbf{X}-\mu)], \tag{4.1}$$

with mean μ and a $[d \times d]$ symmetric and positive definite covariance matrix $\Sigma \equiv \Psi_x^{-1}$. Ψ_x is referred to as the *precision* matrix, and for mathematical convenience will be used through out the thesis instead of the covariance matrix. The *univariate* Gaussian distribution is a special case of the *multivariate* distribution (4.1), where Ψ_x is $[1 \times 1]$. Consider a simple problem of a model that fits a univariate Gaussian distribution with parameters $\boldsymbol{\Theta} = \{\mu, \psi_x\}$, which represents mean and precision respectively, to some observed data $\mathbf{x} = \{x_i\}_{i=1}^{N}$. The Bayesian network for such model in two versions: ML and VB are shown in figure 4.1(a), and figure 4.1(b) respectively, and in both models the $\mathbf{x}$ is assumed to be IID.

Maximum likelihood estimation

We start by inferring parameters using ML. The incomplete-data log likelihood can be written as:

$$\mathcal{L}_{inc}(\boldsymbol{\Theta}|\mathcal{M}) = \ln p(\mathbf{x}|\mu, \Psi_x^{-1}) = \sum_{i=1}^{N} \ln p(x_i|\mu, \Psi_x^{-1}). \tag{4.2}$$

For most choices of density function, the optimum $\boldsymbol{\Theta}$ have to be found by an iterative numerical procedure such as EM-algorithm. However for the special case of (multivariate) normal density, the maximum likelihood solution can be found analytically by differentiating (4.2), with $p(x_i|\mu, \Psi_x^{-1})$ given by (4.1). Since there are no *hidden states* in the ML model, inferring parameters can be regarded as an *one step* of the "EM-algorithm" consisting of only an M-step:

M-step: Compute the sample mean, and the sample covariance

$$\tilde{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad , \quad \tilde{\psi_x}^{-1} = \Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \tilde{\mu})^2. \tag{4.3}$$

Variational Bayes estimation

Variational inference can be used to learn an approximate posterior distribution over the parameters. However it is first necessary to complete the model by defining prior distributions over μ and ψ_x. As discussed earlier (see section 3.2.1), we choose conjugate priors to make inference tractable. Due to the restriction that the precision must be positive, *Gamma distribution* (Appendix B) is chosen, and μ is chosen to be normal distributed

$$p(\mu) = \mathcal{N}(\mu|m, \beta^{-1}); \tag{4.4}$$
$$p(\psi_x) = \Gamma(\psi_x|a, b). \tag{4.5}$$

The Bayesian network for this model is shown in figure 4.1. The variational distribution is chosen to be fully factorized:

$$\mathcal{Q}(\boldsymbol{\Theta}) = \mathcal{Q}(\mu, \psi_x) = \mathcal{Q}(\mu)\mathcal{Q}(\psi_x). \tag{4.6}$$

For full conjugacy, we could use *Normal-Gamma* (or *Normal-Wishart in multivariate case*), our choice of a factorized variational distribution will be justified in example (4.1). As mentioned in section 3.2.1, using conjugate priors result in an optimal $\mathcal{Q}$ that has the same distribution as the prior

$$\mathcal{Q}(\mu) = \mathcal{N}(\mu|\tilde{m}, \tilde{\beta}^{-1}); \tag{4.7}$$
$$p(\psi_x) = \Gamma(\psi_x|\tilde{a}, \tilde{b}). \tag{4.8}$$

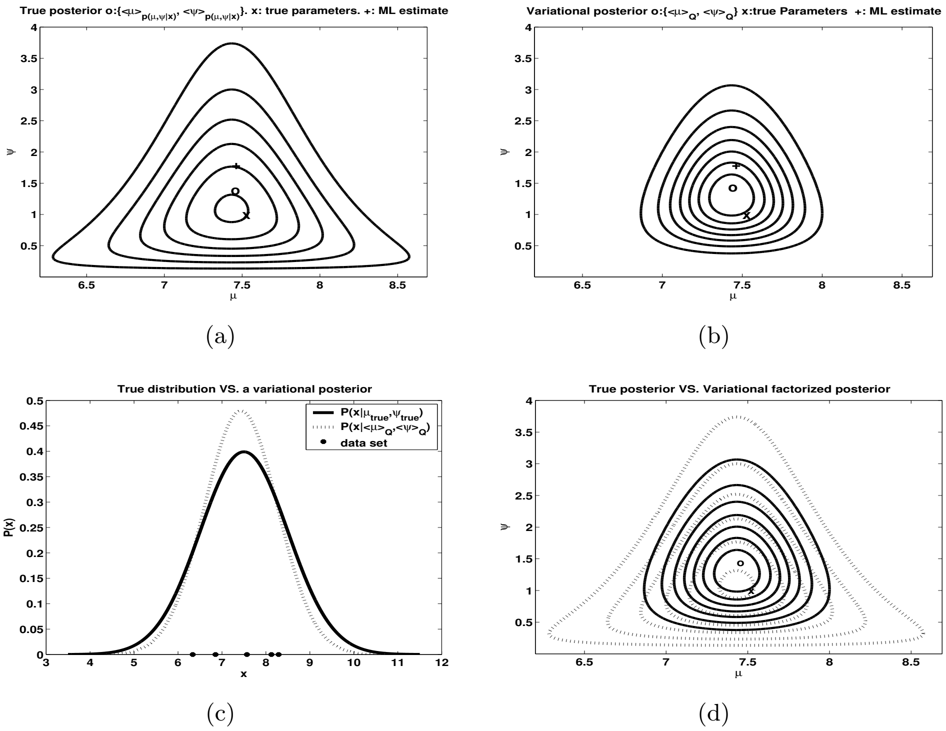


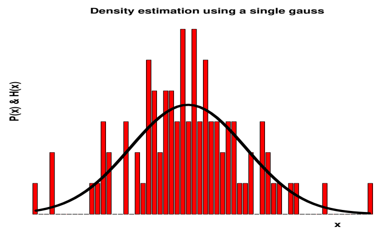
Figure 4.2: This figure illustrate that when factorizing the variational posteriors $\mathcal{Q} \approx \mathcal{Q}(\mu)\mathcal{Q}(\psi_x)$ (b) it fails in capturing the correlations between variables (b), however if the right $\mathrm{KL}(\mathcal{Q}||P)$ rather than $\mathrm{KL}(P||\mathcal{Q})$ (see figure 3.3) is minimized, which is the case here, the optimal variational distribution will lay on the area of high probability. This can easily be seen in (d). In (c) we can see the data points and their distribution (solid line), together with a distribution (dashed line) whose parameters are the expectations of the variational posterior $\mathbf{\Theta} = \{<\mu>_{\mathcal{Q}}, <\psi_x>_{\mathcal{Q}}\}$.

Inference involves then updating the set of variational parameters $\mathbf{\Theta} = \{\tilde{m}, \tilde{\beta}, \tilde{a}, \tilde{b}\}$. Using equation (3.55) we get

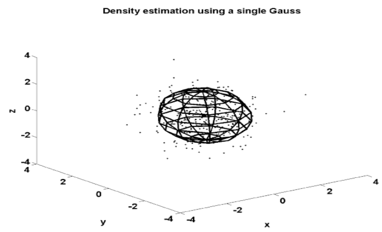
$$\begin{aligned}
\tilde{\beta} &= \beta + N < \psi_x >; && (4.9)\\
\tilde{m} &= \frac{1}{\tilde{\beta}}\left(\beta m + < \psi_x > \sum_{i=1}^{N} x_i\right); && (4.10)\\
\tilde{a} &= a + \frac{N}{2}; && (4.11)\\
\tilde{b} &= b + \frac{1}{2}\sum_{i=1}^{N}(x_i^2 - 2x_i < \mu > + < \mu^2 >). && (4.12)
\end{aligned}$$

All the expectations are wrt. $\mathcal{Q}$. Note the similarity between these updates and the one from equation (3.28). At convergence the variational distribution over the parameters will be the separable distribution closest to the true posterior, in KL divergence sense. This can be illustrated by the following example.

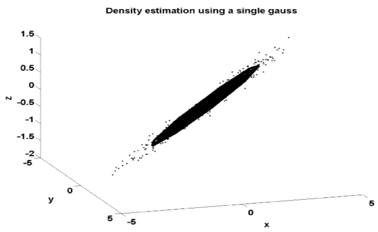
Example 4.1 *Consider a small data set* $\mathbf{X} = \{x_i\}_{i=1}^{N}$ *of* $N = 5$ *samples drawn from a normal distribution with parameters* $\mu_{true} = 7.5$ *and* $\psi_{x_{true}} = 1$*. The hyperparameters are chosen to give a broad (uninformative) priors over* μ *and* ψ_x *(*Appendix *B),*



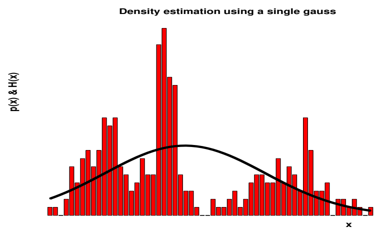
(a)



(b)



(c)



(d)

Figure 4.3: (a) and (b) shows that when the observed data is uni-modal and can be assumed normal distributed, the simple normal Gaussian model gives a good representation to data. However, this simple model has significant limits, it is not suited for multi-modal distributions as shown in (d). Another drawback is that the number of independent parameters can become excessive, where the data can be represented with fewer parameters as in (c), where the latent dimension of $\mathbf{X}$ is lower than the observed d ($2D$ in this case), i.e, several eigenvalues are close to zero.

by setting $a = b =$ 1e-3. Figure 4.2(a) and (b) show the true joint posterior $p(\mu, \psi_x|\mathbf{x})$ and the converged factorized variational posterior $\mathcal{Q} = \mathcal{Q}(\mu)\mathcal{Q}(\psi_x)$. It can be seen that the factorized (separable) variational distribution fails in capturing the correlation between the variables, but in the other hand it is similar to the true posterior distribution, particularly in the areas of high probabilities which are of interest. This can in fact be explained by figure 3.3, where we showed that when minimizing $\mathrm{KL}(\mathcal{Q}||P)$ the learning favors the distribution $\mathcal{Q}$ that fits an area of high probability (or one mode in case of bimodal). Figure 4.2(c) shows the distributions whose parameters are the expectation of the variational posterior, i.e., $\mathbf{\Theta} = \{< \mu >_{\mathcal{Q}}, < \psi_x >_{\mathcal{Q}}\}$, together with the data points and the distribution they were sampled from.

The results from the above example will be taken as argument, in future models to perform a factorized variational distribution between the mean and the covariance.

While the simple normal distribution is widely used, it suffers from some significant limitations. In particular, it can be insufficiently flexible since it can only represent uni-modal distributions, (figure 4.3(a) and (b), for $1D$ and $3D$ respectively.), and fails in representing multi-modal distributions (figure 4.3(d)). A more general family of distributions can be obtained by considering mixture of Gaussians section 5.1, corresponding to the introduction of a discrete latent variable. On the other hand the normal distribution can often be proven to be flexible [6] in that the number of independent parameters can be excessive, and grows rapidly with the dimensionality d, where often fewer parameters are needed, due to the assumption that in higher dimension, several eigen values are very small. This is illustrated in figure 4.3(c), where the *hidden* dimensionality of the data is lower than the observed one ($\mathsf{rank}(\mathbf{X}) = 2 < d = 3$), and therefore data can be described by fewer parameters. Each of those approaches will be discussed in following sections.

4.2 Latent variable models

A Gaussian model with full covariance matrix have several disadvantages [33], all related to the fact that, the sample covariance (4.3) then contains $d(d+1)/2$ free parameters (the factor 1/2 is due to the symmetry). There are further d parameters for the mean, making $d(d+3)/2$. which become unwieldy for high-dimensional data. Consequently computing the covariance in "M-step" (4.3) requires $\mathcal{O}(d^2N)$, and excessively large number of data is required to ensure that the maximum likelihood for Σ is well determined ($N > d+1$)[1]. One way to reduce the number of free parameters in the model is to consider a diagonal covariance matrix, which has just d free parameters. This, however, corresponds to a very strong assumption, namely that the components of $\mathbf{X}$ are statistically independent[2], and such a model is therefore unable to capture the correlations between different components. Next we show how the number of degrees of freedom within the model can be controlled, while still allowing correlations to be captured, by using latent (or 'hidden') variables. The goal of a latent variable model is to express the distribution $p(\mathbf{x})$ ($\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$) of the variables $\mathbf{x}_i = \{x_{i_j}\}_{j=1}^{d_X}$ (where d_X is what we previously noted as d) in terms of a smaller number of latent variables $\mathbf{s}_i = \{s_{i_k}\}_{k=1}^{d_S}$ where $d_S < d_X$. This can be interpreted as a form of dimensionality reduction or feature extraction. Assuming that the latent variable are continuous now, marginalizing over them is done by the integration in equation (3.3) repeated here for convenience

$$\mathcal{L}_{inc}(\mathbf{\Theta}) = \ln p(\mathbf{X}|\mathbf{\Theta}) = \sum_i \ln \int p(\mathbf{x}_i, \mathbf{s}|\mathbf{\Theta})\ \mathrm{d}\mathbf{s} = \sum_i \ln \int p(\mathbf{x}_i|\mathbf{s}, \mathbf{\Theta}) p(\mathbf{s})\ \mathrm{d}\mathbf{s}, \quad (4.13)$$

where the model $\mathcal{M}$ is omitted for brevity. To keep this integration tractable, we limit ourselves to Gaussian distributions. Assuming a single data point $\mathbf{x}$, we can define $p(\mathbf{x}|\mathbf{s}, \mathbf{\Theta})$ trough the following mapping from data space to latent space, together with a Gaussian prior $p(\mathbf{s})$ for the latent state

$$\mathbf{X} = f(\mathbf{s}; \mathbf{A}) + \varepsilon, \qquad \text{where } \left\{ \begin{array}{ll} \mathbf{s} \sim & \mathcal{N}(\mathbf{0}, \mathbf{C}) \\ \varepsilon \sim & \mathcal{N}(\mathbf{0}, \mathbf{\Psi}^{-1}) \end{array} \right. \quad (4.14)$$

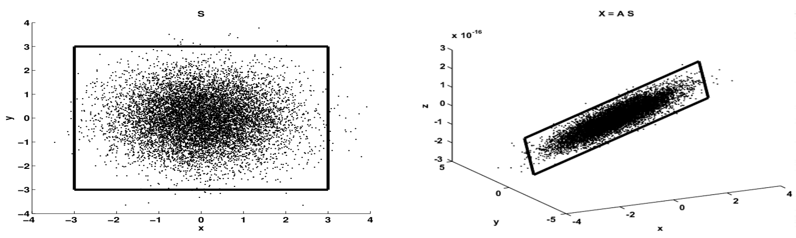
where $f(\mathbf{s}; \mathbf{A})$ is a function of the latent variable $\mathbf{s}$ with parameters $\mathbf{A}$, and ε is the $\mathbf{s}$-independent zeros mean noise process, with precision matrix Ψ. If the components in ε are uncorrelated (i.e., Ψ is a diagonal matrix), the conditional distribution $p(\mathbf{x}|\mathbf{s}, \mathbf{\Theta})$ will factorize as follows

$$p(\mathbf{x}|\mathbf{s}, \mathbf{\Theta}) = \prod_{j=1}^{d_X} p(x_j|\mathbf{s}, \mathbf{\Theta}). \quad (4.15)$$

The definition of latent variable model is completed when specifying the distribution $p(\varepsilon)$, the mapping $f(\mathbf{s}; \mathbf{A})$, and the prior $p(\mathbf{s})$.

[1] If $N \leq d+1$ the covariance becomes singular.

[2] Note that this should not be confused with IID.



(a) (b)

Figure 4.4: This figure illustrate how the hidden states $\mathbf{s}$ of dimension d_S ($2D$ in this case), is transformed to a higher dimensional data space d_X ($3D$ in this case) using the mixing matrix $\mathbf{A}$ from the linear generative model (4.16), the rectangle is added to illustrate the plan where the latent data is lying. Note that the added noise is not illustrated here for simplicity.

4.3 Factor Analysis and PPCA

A simple latent variable model is achieved by constraining the mapping to be a linear function. Thus equation (4.14) becomes:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mu + \varepsilon, \tag{4.16}$$

with parameters $\mathbf{\Theta} = \{\mathbf{A}, \mu, \Psi\}$. Since the mapping is now restricted to be linear, and since the output of a linear system whose input is Gaussian distributed ($p(\mathbf{s})$), is again Gaussian distributed, everything stays entirely in the Gaussian domain. The conditional distribution of the latent states given the observed variables, $p(\mathbf{x}|\mathbf{s}, \mathbf{\Theta})$ (4.15) can be written as [3]:

$$\mathbf{x}|\{\mathbf{s}, \mathbf{\Theta}\} \sim \mathcal{N}(\mathbf{A}\mathbf{s} + \mu, \Psi^{-1}) \tag{4.17}$$

where $\mathbf{x}|\cdot$ is a shorthand for $p(\mathbf{x}|\cdot)$. The convolution of the above quantity with Gaussian prior $p(\mathbf{s})$ (4.13) can be performed analytically. This gives the marginal distribution of the observed data $p(\mathbf{x}|\mathbf{\Theta})$, which is also Gaussian:

$$\mathbf{x}|\mathbf{\Theta} \sim \mathcal{N}(\mu, \mathbf{A}\mathbf{C}\mathbf{A}^\top + \Psi^{-1}) \tag{4.18}$$

Due to the degeneracy between $\mathbf{C}$ and $\mathbf{A}$ [37], there is no loss of generality in restricting $\mathbf{C}$ to be either a diagonal or even the identity matrix[4] $I_{d_\mathbf{s}}$, where $\mathbf{A}$ will include all the informations that might be assigned to $\mathbf{C}$. Thus setting $\mathbf{C} = I_{d_\mathbf{s}}$, the above equation then becomes:

$$\mathbf{x}|\mathbf{\Theta} \sim \mathcal{N}(\mu, \mathbf{A}\mathbf{A}^\top + \Psi^{-1}) \tag{4.19}$$

Furthermore the noise precision Ψ *must* be restricted in some way, for the model to capture any interesting or informative projections in the state $\mathbf{s}$, otherwise the learning will assign Ψ^{-1} to the sample covariance of the observed data by setting $\mathbf{A} = 0$ (i.e.,

[3]Since $<\mathbf{x}|\mathbf{s},\mathbf{\Theta}> = <(\mathbf{A}\mathbf{s} + \mu + \varepsilon)|\cdot> = \mathbf{A} <\mathbf{s}|\cdot> + <\mu|\cdot> + <\varepsilon|\cdot> = \mathbf{A}\mathbf{s} + \mu$ and $cov(\mathbf{x}|\mathbf{s},\mathbf{\Theta}) = <(\mathbf{A}\mathbf{s} + \mu + \varepsilon)(\mathbf{A}\mathbf{s} + \mu + \varepsilon)^\top|\cdot> = <\varepsilon\varepsilon^\top|\cdot> = \Psi^{-1}$

[4]If we split $\mathbf{C}$ to its eigenvectors U and eigenvalues Λ and rewrite the covariance of $\mathbf{x}$ (4.17) the degeneracy becomes clear: $\mathbf{A}\mathbf{C}\mathbf{A}^\top = (\mathbf{A}U\Lambda^{1/2})(\mathbf{A}U\Lambda^{1/2})^\top$. To make $\mathbf{C}$ diagonal: $\Rightarrow \mathbf{A} \leftarrow \mathbf{A}U$, or identity matrix $I_{d_\mathbf{s}}$: $\Rightarrow \mathbf{A} \leftarrow \mathbf{A}U\Lambda^{1/2}$.

$\Psi = \Psi_x$ (4.3)). Thus the model will explain all the structure in the data as noise. Figure 4.4 gives an intuitive spatial way to think about the model in equation (4.16) (where $\mu = 0$ in this case.). Consider an unseen d_S dimensional Gaussian data $\mathbf{s}$, with identity covariance $\mathbf{C} = I_{d_\mathbf{s}}$, therefore the circular (spherical ball, for $d_S > 2$) shape in figure 4.4(a). This ball is then stretched and rotated into d_X-dimensional space by the mixing matrix $\mathbf{A}$ as seen in figure 4.4(b). The resulting shape is then convolved with the diagonal covariance of ε (Ψ^{-1}) to get the final covariance model for $\mathbf{x}$. The learning then consist of getting this model to be as close as possible to the sample covariance of the data. By constraining Ψ we can force interesting information to appear in both Ψ and $\mathbf{A}$.

Note that by constraining the covariance $\mathbf{C}$ to be the identity matrix, there will be an infinite number of solutions to the mixing matrix $\mathbf{A}$, this can easily be shown by rewritten the covariance model for $\mathbf{x}$ in (4.19) as

$$\mathbf{A}\mathbf{A}^\top + \Psi^{-1} = \mathbf{A}(\mathbf{V}\mathbf{V}^\top)\mathbf{A}^\top + \Psi^{-1} \qquad \forall\, \mathbf{V} \text{ such that } \mathbf{V}\mathbf{V}^\top = I_{d_\mathbf{s}} \tag{4.20}$$

where $\mathbf{V}$ is any $[d_S \times d_S]$ orthonormal matrix, their are several methods to avoid this non-uniqueness, one of the most used criteria is the one introduced by *Kaiser*, the so-called *varimax* [27, 39], which states, that one should choose the orthonormal matrix V which maximizes the following quantity

$$\sum_k d_S \left[\sum_j (\frac{\mathcal{A}_{j,k}}{h_j^2})^2 - \frac{1}{d_S}\{\sum_{j'}(\frac{\mathcal{A}_{j',k}}{h_{j'}^2})\}^2 \right] \tag{4.21}$$

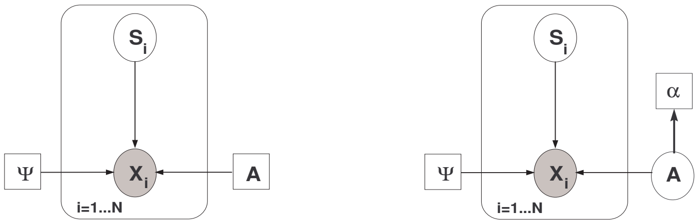
where $h_j^2 = \sum_k \mathcal{A}_{j,k}^2$. Maximizing the above criteria will result in large numbers of $\mathcal{A}_{i,j}$ being either zero or large value, which is actually an answer to a simple structure that is easy to interpret.

Restricting the noise precision matrix Ψ in an appropriate way, will result in powerful models used by among others *Machine learning* community, two cases are stated here:

- $\Psi \leftarrow$ **diag**(Ψ)**:** The latent variable model is the standard statistical model known as maximum likelihood *Factor Analysis* [39] (MLFA). The bayesian network of such model is shown in figure 4.5. The unknown states $\mathbf{s}$ are called *factors*, the matrix $\mathbf{A}$ is called the *factor loading* matrix and the diagonal elements in Ψ^{-1} are called the *uniqueness*

- $\Psi = \sigma^{-2} I_{d_\mathbf{s}}$**:** In this case the latent variable model is referred to as *Probabilistic Principle Component Analysis* (PPCA) [8] or sensible PCA (SPCA) [37], where σ^2 is referred to as the *global noise level.* In this case the columns of $\mathbf{A}$ will span the *principle subspace* (the same found by PCA). Setting $\sigma^2 \to 0$ conventional PCA is then recovered.

From above one can wonder, what is the consequences of the small difference in the shape of noise, spherical for PPCA and elliptical for FA, might be. However it is easy to see from the generative model (4.16), and from the model covariance of the marginal distribution (4.19) that, FA model is insensitive to rescaling the coordinate of data (i.e., scaling parallel to the original axes), since the rescaling of the j'th dimension will corresponds



(a) ML Factor Analysis model. (b) MAP model for **A**.

Figure 4.5: This figure shows the Bayesian net for two learning methods ML (a) and MAP (b). Both cases include two statistical models for (a): when $\mathbf{\Psi}$ is a diagonal $[d_X \times d_X]$ matrix, i.e. the distribution of the noise $\mathcal{N}(0, \mathbf{\Psi}^{-1})$ is an *hyperellipsoid*, the model corresponds to maximum likelihood FA. Whereas if $\mathbf{\Psi}$ is replaced by the $[1 \times 1]$ dimensional precision $\psi = \sigma^{-2}$, i.e., the noise is *hyperspherical*, the model corresponds to maximum likelihood PPCA. (b) is the augmented model where the parameter $\mathbf{A}$ is no longer a deterministic variable, but a stochastic variable with a (Gaussian) prior distribution, controlled by the (deterministic) *hyperparameter* α (precision). The resulting model is the penalized ML or MAP model, though only for $\mathbf{A}$, since $\mathbf{\Psi}$ is still deterministic and inferred using ML.

to multiplying the scaling factor to the corresponding row of $\mathbf{A}$ and the corresponding elements in the diagonal $\mathbf{\Psi}$, independently on the other dimensions. However, FA is sensitive to the choice of coordinate system in data space, i.e. an orthogonal transformation of data can not be captured by the diagonalized $\mathbf{\Psi}$, since the noise is only measured along the data axis.
PPCA on the other hand, is exactly the other way around. Its insensitive to rotation, since this rotation does not affect the spherical shaped model of noise, and can be incorporated by left multiplying $\mathbf{A}$ by the same rotation matrix [37]. However, because of this spherical property of $\mathbf{\Psi} = \psi d_X$, PPCA can only handle the case of rescaling the data coordinate, when it is performed by the same factor (for all axis). A simple test that reveals the weakness of PPCA compare to FA in separating the information from noise can be seen in 4.2.

The restrictions applied to linear latent variable model can be regarded as way of capturing the covariance structure of the d_X-dimensional observed data trough $\mathbf{A}\mathbf{A}^\top + \Psi^{-1}$ using at most $d_X(d_S + 1)$ parameters. This might be interpreted as a sort of *discrete* regularization in which one can tune the complexity of the model by choosing the dimension of latent space d_S. This means that one can cover the whole spectrum of covariance matrices with $\mathcal{O}(d_X)$ parameters for a very flexible model, to a more complex model with $\mathcal{O}(d_X^2)$ parameters. However, given a data set, we are mostly interested in finding the most appropriate value of d_S.
From the above items, it can be seen that a learning algorithm for PPCA will be a special case of learning FA-model. The problem of fitting a FA (or PPCA) model to the observed data, can be thought of as equivalent to inferring the appropriate model parameters $\mathbf{\Theta} = \{\mathbf{A}, \mu, \Psi\}$ as well as the hidden states $\mathbf{s}$, which when plugged into the generative model (4.16) are most likely to generate the observed data distribution.

Learning FA and PPCA using EM.

The simplest FA model is the one where all the the parameters are deterministic, this model is MLFA, shown in figure 4.5(a). Given a set of N data points $\mathbf{X}$ we wish to estimate the parameters $\boldsymbol{\Theta} = \{\mathbf{A}, \mu, \boldsymbol{\Psi}\}$. When a closed form ML is intractable the EM algorithm can be used, as described earlier (see section 3.3.2), in the EM formalism instead of maximizing the incomplete data log-likelihood $\mathcal{L}_{inc}(\boldsymbol{\Theta}) = \ln p(\mathbf{X}|\boldsymbol{\Theta})$ of the observed data (3.35), we attempt to maximize the *complete data* log-likelihood $\mathcal{L}_c(\boldsymbol{\Theta}) = p(\mathbf{X}, \mathbf{S}|\boldsymbol{\Theta})$. Since this quantity is a function of the the hidden states $\mathbf{S}$, which we obviously can not observe, we must work with the *expectation* of this quantity wrt. some distribution $\mathcal{Q}(\mathbf{X})$ (the left hand term in equation (3.41)). We showed also that this expectation is always a lower bound to the incomplete data likelihood for any arbitrary $\mathcal{Q}(\mathbf{S})$ (3.38), and is only equal to $\mathcal{L}_{inc}(\boldsymbol{\Theta})$ when the expectation is taking wrt. the posterior distribution of $\mathbf{S}$ (i.e., when $\mathcal{Q}(\mathbf{S}) = p(\mathbf{S}|\mathbf{X}, \boldsymbol{\Theta})$). The complete data log likelihood can be written as follows:

$$\begin{aligned}
\mathcal{L}_c(\boldsymbol{\Theta}) &= \ln \prod_i^N p(\mathbf{x}_i, \mathbf{s}_i|\boldsymbol{\Theta}) && (4.22)\\
&= \sum_i^N \ln p(\mathbf{x}_i, \mathbf{s}_i|\boldsymbol{\Theta})\\
&= \sum_i^N \ln p(\mathbf{x}_i|\mathbf{s}_i, \boldsymbol{\Theta}) + \sum_i^N \ln p(\mathbf{s}_i), && (4.23)
\end{aligned}$$

where the factorization of data in (4.22) is due to the assumption that $\mathbf{X}$ is IID. Using equations (4.17) for $p(\mathbf{X}|\mathbf{S}, \boldsymbol{\Theta})$ and (4.18) for $p(\mathbf{S})$, the expectation of the complete data log likelihood $\mathcal{L}_c(\boldsymbol{\Theta})$ wrt. to some distribution $\mathcal{Q}(\mathbf{S})$ is then expressed as

$$\begin{aligned}
< \mathcal{L}_c(\boldsymbol{\Theta}) > &= < \sum_i^N \ln p(\mathbf{x}_i|\mathbf{s}_i, \boldsymbol{\Theta}) > + < \sum_i^N \ln p(\mathbf{s}_i) >\\
&= \sum_i^N < \ln p(\mathbf{x}_i|\mathbf{s}_i, \boldsymbol{\Theta}) > + \sum_i^N < \ln p(\mathbf{s}_i) > && (4.24)\\
&= \sum_i^N < \ln \frac{|\boldsymbol{\Psi}|^{1/2}}{(2\pi)^{d_X/2}} \exp\Big\{ -\frac{1}{2}(\mathbf{x}_i - \mathbf{A}\mathbf{s}_i)^\top \boldsymbol{\Psi}(\mathbf{x}_i - \mathbf{A}\mathbf{s}_i)\Big\} >\\
&\quad + \sum_i^N < \ln \frac{1}{(2\pi)^{d_S/2}} \exp\Big\{ -\frac{1}{2}\mathbf{s}_i^\top \mathbf{s}_i\Big\} > && (4.25)\\
&= \sum_i^N < \ln \frac{|\boldsymbol{\Psi}|^{1/2}}{(2\pi)^{d_X/2}} \exp\Big\{ -\frac{1}{2}(\mathbf{x}_i - \mathbf{A}\mathbf{s}_i)^\top \boldsymbol{\Psi}(\mathbf{x}_i - \mathbf{A}\mathbf{s}_i)\Big\} >\\
&\quad + \sum_i^N < \ln \frac{1}{(2\pi)^{d_S/2}} \exp\Big\{ -\frac{1}{2}\mathbf{s}_i^\top \mathbf{s}_i\Big\} >, && (4.26)
\end{aligned}$$

where linearity property of the expectation operator $< . >$ is used, for brevity the suffix $\mathcal{Q}(\mathbf{S})$ is omitted. According to equation (3.45), the quantity to maximize in the M step is $< \ln p(\mathbf{X}|\mathbf{S}, \boldsymbol{\Theta}) >$, which is, the left hand term in equation (4.25), using the fact that $< y >_{\mathcal{Q}_z} = y$ for any variable y independent on z, we get

$$
\begin{aligned}
< \ln p(\mathbf{X}|\mathbf{S},\boldsymbol{\Theta}) > &= \mathrm{k} + \frac{N}{2}\ln|\boldsymbol{\Psi}| - \frac{1}{2}\sum_{i=1}^{N}\langle(\mathbf{x}_i - \mathbf{A}\mathbf{s}_i)^\top\boldsymbol{\Psi}(\mathbf{x}_i - \mathbf{A}\mathbf{s}_i)\rangle && (4.27)\\
&= \mathrm{k} + \frac{N}{2}\ln|\boldsymbol{\Psi}| - \frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i^\top\boldsymbol{\Psi}\mathbf{x}_i - 2\mathbf{x}_i^\top\boldsymbol{\Psi}\mathbf{A} < \mathbf{s}_i > \\
&\quad + \mathrm{TR}[\mathbf{A}^\top\boldsymbol{\Psi}\mathbf{A} < \mathbf{s}_i\mathbf{s}_i^\top >]). && (4.28)
\end{aligned}
$$

Here the relation $\mathbf{x}^\top\mathbf{A}\mathbf{x} = \mathrm{TR}[\mathbf{A}\mathbf{x}\mathbf{x}^\top]$, where $\mathrm{TR}[\cdot]$ is the trace operator, has been used. The term k represent constants.
Maximizing the quantity $< \ln p(\mathbf{X}|\mathbf{S},\boldsymbol{\Theta}) >$ involves taking partial derivative wrt. each parameter of interest. For the case of $\mathbf{A}$ it is convenient to work with (4.28) as

$$
\frac{\partial}{\partial\mathbf{A}} < \ln p(\mathbf{X}|\mathbf{S},\boldsymbol{\Theta}) > = -\frac{1}{2}\sum_{i=1}^{N}\big(-2\boldsymbol{\Psi}\mathbf{x}_i < \mathbf{s}_i >^\top + 2\boldsymbol{\Psi}\mathbf{A} < \mathbf{s}_i\mathbf{s}_i^\top > \big), \qquad (4.29)
$$

where $\frac{\partial}{\partial Z}A^\top ZB = AB^\top$. For the case of $\boldsymbol{\Psi}$ on the other hand it is convenient to work with (4.27) as

$$
\frac{\partial}{\partial\boldsymbol{\Psi}} < \ln p(\mathbf{X}|\mathbf{S},\boldsymbol{\Theta}) > = \frac{N}{2}\boldsymbol{\Psi}^{-1} - \frac{1}{2}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^\top + \big(\sum_{i=1}^{N}\mathbf{x}_i\langle\mathbf{s}_i\rangle^\top\big)\mathbf{A}^\top - \frac{1}{2}\mathbf{A}\big(\sum_{i=1}^{N}\langle\mathbf{s}_i\mathbf{s}_i^\top\rangle\big)\mathbf{A}^\top \qquad (4.30)
$$

Care must be taking when constraining $\boldsymbol{\Psi}$ properly, which is fortunately in case of FA as easy as taking the diagonal of the unconstrained ML estimate. In case of PPCA we take the average of these diagonal elements.
Now setting equations (4.29) and (4.30) to zeros and solve for $\mathbf{A}$ and $\boldsymbol{\Psi}^{-1}$ respectively to get the M step:

M step:

$$
\begin{aligned}
\{\mathbf{A},\boldsymbol{\Psi}\}_{\mathrm{ML}} &= \underset{\{\mathbf{A},\boldsymbol{\Psi}\}}{\operatorname{argmax}}\ \langle\ln p(\mathbf{X}|\mathbf{S},\{\mathbf{A},\boldsymbol{\Psi}\})\rangle && (4.31)\\
\mathbf{A} &= \big(\sum_{i=1}^{N}\mathbf{x}_i < \mathbf{s}_i >^\top\big)\big(< \mathbf{s}_i\mathbf{s}_i^\top > \big)^{-1} && (4.32)\\
(\mathrm{FA}) \Rightarrow \boldsymbol{\Psi}^{-1} &= \frac{1}{N}\mathrm{DIAG}\big[\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^\top - \big(\mathbf{x}_i\langle\mathbf{s}_i\rangle^\top\big)\mathbf{A}^\top\big] && (4.33)\\
(\mathrm{PPCA}) \Rightarrow \psi^{-1} &= \frac{1}{N}\mathrm{TR}\big[\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^\top - \big(\mathbf{x}_i\langle\mathbf{s}_i\rangle^\top\big)\mathbf{A}^\top\big]. && (4.34)
\end{aligned}
$$

We are still left with the problem of determining the actual values of $\langle\mathbf{s}_i\rangle$ and $\langle\mathbf{s}_i\mathbf{s}_i^\top\rangle$. As mentioned earlier, in order to guarantee that we indeed maximizing the incomplete data log likelihood, it is essential that the expected complete log likelihood is maximized by taking the expectation wrt. the posterior $p(\mathbf{S}|\mathbf{X},\boldsymbol{\Theta})$. Thus the expectations $\langle\mathbf{s}_i\rangle$ and $\langle\mathbf{s}_i\mathbf{s}_i^\top\rangle$ should actually be computed wrt. that posterior. In this relatively simple case, we can actually obtain an analytical expression for the posterior distribution $p(\mathbf{s}_i|\mathbf{x}_i,\boldsymbol{\Theta})$ using Bayes rule as follows:

$$\begin{aligned}
p(\mathbf{s}_i|\mathbf{x}_i,\mathbf{\Theta}) &\propto p(\mathbf{x}_i|\mathbf{s}_i)p(\mathbf{s}_i) \\
\Rightarrow \ln p(\mathbf{s}_i|\mathbf{x}_i,\mathbf{\Theta}) &= \mathcal{L}_c(\mathbf{\Theta}) + \mathrm{k} \\
&= -\frac{1}{2}\left(\mathbf{x}_i^\top \mathbf{\Psi}\mathbf{x}_i - 2\mathbf{x}_i^\top \mathbf{A}^\top \mathbf{\Psi}\mathbf{x}_i + \mathbf{s}_i^\top (I_{d_\mathbf{s}} + \mathbf{A}^\top \mathbf{\Psi}\mathbf{A})\mathbf{s}_i\right) + \mathrm{k}. \qquad (4.35)
\end{aligned}$$

From the quadratic form we can infer that the posterior distribution of $\mathbf{s}_i$ is Gaussian:

$$p(\mathbf{s}_i|\mathbf{x}_i,\mathbf{\Theta}) = \mathcal{N}(\mathbf{m}_{\mathbf{s}_i}^{(i)}, \Sigma_\mathbf{s}), \qquad (4.36)$$

with

$$\begin{aligned}
\Sigma_\mathbf{s} &= \left(I_{d_\mathbf{s}} + \mathbf{A}^\top \mathbf{\Psi}\mathbf{A}\right)^{-1} \qquad (4.37) \\
\mathbf{m}_{s_i}^{(i)} &= \Sigma_\mathbf{s}\mathbf{A}^\top \mathbf{\Psi}\mathbf{x}_i \\
&= \left(I_{d_\mathbf{s}} + \mathbf{A}^\top \mathbf{\Psi}\mathbf{A}\right)^{-1}\mathbf{A}^\top \mathbf{\Psi}\mathbf{x}_i \\
&= \mathbf{M}\mathbf{x}_i, \qquad (4.38)
\end{aligned}$$

where we define $\mathbf{M} \equiv \left(I_{d_\mathbf{s}} + \mathbf{A}^\top \mathbf{\Psi}\mathbf{A}\right)^{-1}\mathbf{A}^\top \mathbf{\Psi}$. Given this distribution, the E step can then be written as:

E step:

$$\begin{aligned}
\langle \mathbf{s}_i \rangle &= \mathbf{m}_{s_i}^{(i)} = \mathbf{M}\mathbf{x}_i; \qquad (4.39) \\
\langle \mathbf{s}_i\mathbf{s}_i^\top \rangle &= \Sigma_\mathbf{s} + \langle \mathbf{s}_i \rangle \langle \mathbf{s}_i \rangle^\top \\
&= \left(I_{d_\mathbf{s}} + \mathbf{A}^\top \mathbf{\Psi}\mathbf{A}\right)^{-1} + \mathbf{M}\mathbf{x}_i\mathbf{x}_i^\top \mathbf{M}^\top \\
&= I_{d_\mathbf{s}} - \mathbf{A}^\top \left(\mathbf{\Psi} + \mathbf{A}\mathbf{A}^\top\right)^{-1}\mathbf{A} + \mathbf{M}\mathbf{x}_i\mathbf{x}_i^\top \mathbf{M}^\top \qquad (4.40) \\
\Rightarrow \langle \mathbf{s}_i\mathbf{s}_i^\top \rangle &= I_{d_\mathbf{s}} - \mathbf{M}\mathbf{A} + \mathbf{M}\mathbf{x}_i\mathbf{x}_i^\top \mathbf{M}^\top, \qquad (4.41)
\end{aligned}$$

where we applied the *Sherman-Morrison-Woodbury* matrix inversion theorem on equation (4.40) as

$$\mathbf{A}^\top \left(\mathbf{\Psi} + \mathbf{A}\mathbf{A}^\top\right)^{-1} = \left(I_{d_\mathbf{s}} + \mathbf{A}^\top \mathbf{\Psi}\mathbf{A}\right)^{-1}\mathbf{A}^\top \mathbf{\Psi} = \mathbf{M}. \qquad (4.42)$$

Notice that the second form is much easier to evaluate since $\left(I_{d_\mathbf{s}} + \mathbf{A}^\top \mathbf{\Psi}\mathbf{A}\right)$ is a smaller matrix than $\left(\mathbf{\Psi} + \mathbf{A}\mathbf{A}^\top\right)$ and $\mathbf{\Psi}$ is diagonal.

As mentioned earlier in section 3.1, the maximum likelihood does not take into account the model complexity, and thus tends to prefer more complex models. Take the case of inferring $\mathbf{A}$ $[d_S \times d_X]$, the closest d_S gets to d_X, the more it becomes quadratic in shape, and the fraction of information that should be explained as noise, and hence included in $\mathbf{\Psi}$, will now be explained as useful information and included in $\mathbf{A}$ instead, as extra columns. To prevent this type of problems the penalized log likelihood was introduced in form of ARD (see equations (3.4) and 3.6). This quantity was later expressed in a nicer probabilistic way, as a MAP estimate in equation (3.15), generated by multiplying the incomplete likelihood with the parameter prior (3.11). We define a *hyperellipsoidal* Gaussian prior of the mixing matrix $\mathbf{A} = \{\mathcal{A}_k\}_{k=1}^d$

$$\begin{aligned} p(\mathbf{A}|\alpha) &= \mathcal{N}(\mathbf{A}; 0, \alpha^{-1}) && (4.43)\\ &= \big(\frac{|\alpha|}{2\pi}\big)^{d_X/2} \exp\big(-\frac{1}{2}\mathbf{A}\alpha\mathbf{A}^\top\big) && (4.44)\\ &= \prod_{k=1}^{d} \big(\frac{\alpha_k}{2\pi}\big)^{d_X/2} \exp\big(-\frac{\alpha_k}{2}\mathcal{A}_k^\top\mathcal{A}_k\big). && (4.45) \end{aligned}$$

α is a diagonal matrix with diagonal element β_i representing the precision of column vector $\mathcal{A}_i$. This matrix has to be inferred as the other deterministic parameters. The variable d over the product symbol in equation (4.45) is the estimate to the true hidden dimensionality d_S.

As seen in figure 4.5(b), the change we made to the MLFA model, to become a MAP model (though only for $\mathbf{A}$ since $\mathbf{\Psi}$ is still deterministic and therefore inferred using ML.) can be expressed graphically by adding a hyperparameter α as a *parent* to $\mathbf{A}$, which becomes now a circle due to the (Gaussian) distribution over it.

The new cost function is now proportional to the posterior distribution ($p(\mathbf{A}|\mathbf{X}, \alpha)$) and can be expressed using the Bayes rule as:

$$\begin{aligned} p(\mathbf{A}|\mathbf{X}, \alpha) &\propto p(\mathbf{X}|\mathbf{A})\ p(\mathbf{A}|\alpha)\\ \Rightarrow \quad \ln p(\mathbf{A}|\mathbf{X}, \alpha) &= \mathcal{L}_{inc}(\mathbf{A}, \mathbf{\Psi}) + \ln p(\mathbf{A}|\alpha) + \mathrm{k}, && (4.46) \end{aligned}$$

where the constant k represent all the values that are independent on $\mathbf{A}$ (in this case $-\log p(\mathbf{X}|\alpha)$). Using equation (4.45) in (4.46) we get:

$$\ln p(\mathbf{A}|\mathbf{X}, \alpha) \quad \propto \quad \mathcal{L}_{inc}(\mathbf{A}, \mathbf{\Psi}) - \frac{1}{2}\sum_{k}^{d} \alpha_k \mathcal{A}_k^\top \mathcal{A}_k. \tag{4.47}$$

Since we showed that $\mathcal{L}_c$ is a lower bound to the true data log likelihood $\mathcal{L}_{inc}$, i.e., $\mathcal{L}_c \geq \mathcal{L}_{inc}$, equation (4.47) can be rewritten as:

$$\ln p(\mathbf{A}|\mathbf{X}, \alpha) \quad \propto \quad \mathcal{L}_c(\mathbf{A}, \mathbf{\Psi}) - \frac{1}{2}\sum_{k}^{d} \alpha_k \mathcal{A}_k^\top \mathcal{A}_k. \tag{4.48}$$

Since this penalty term increases together with the number of valid columns in the mixing matrix $\mathbf{A}$ (i.e., d estimation to the latent space dimensionality.). In this way the new cost function is no longer maximized by increasing the hidden dimensionality of the model.

$\mathbf{A}$ and $\mathbf{\Psi}$ are inferred by maximizing the MAP (for $\mathbf{A}$) function (4.48)

$$\begin{aligned} \mathbf{A}_{\mathrm{MAP}} &= \underset{\mathbf{A}}{\mathrm{argmax}} \quad \ln p(\mathbf{A}|\mathbf{X}, \alpha)\\ &= \underset{\mathbf{A}}{\mathrm{argmax}} \quad \mathcal{L}_c(\mathbf{A}, \mathbf{\Psi}) - \frac{1}{2}\sum_{k}^{d} \alpha_k \mathcal{A}_k^\top \mathcal{A}_k, && (4.49) \end{aligned}$$

we replace $\mathcal{L}_c(\mathbf{A}, \mathbf{\Psi})$ used in the EM algorithm (4.22) by the new cost function $\ln p(\mathbf{A}|\mathbf{X}, \alpha)$ and proceed exactly as before. Without going into details, we present the updating formula for $\mathbf{A}$, this result is achieved by taking the expectation of (4.48), and maximizing

with respect to $\mathbf{A}$. In the case of FA each *row* $\mathcal{A}_j$ in *factor loading* $\mathbf{A}$ is updated in the M step as:
MAP M step: for FA

$$\mathcal{A}_j = \left(\sum_i^N x_{i,j}\langle \mathbf{s}_i\rangle^T\right)\ \left(\sum_i^N \langle \mathbf{s}_i\mathbf{s}_i^T\rangle\ +\ \psi_j\alpha\right)^{-1}. \tag{4.50}$$

Since for the PPCA case the noise is hyper-spherical, with a single precision parameter ψ. The updating formula for $\mathbf{A}$ in PPCA becomes:

MAP M step: for PPCA

$$\mathbf{A} = \left(\sum_i^N \mathbf{x}_i\langle \mathbf{s}_i\rangle^T\right)\ \left(\sum_i^N \langle \mathbf{s}_i\mathbf{s}_i^T\rangle\ +\ \psi\alpha\right)^{-1}. \tag{4.51}$$

Note that $\mathbf{\Psi}$ (FA) and ψ (PPCA) are still inferred using ML in equations (4.33) and (4.34) respectively.
The estimation of α is somewhat more difficult, since the likelihood of α (see figure 4.5(b)) requires marginalizing the random value $\mathbf{A}$.

$$p(\mathbf{X}|\alpha) = \int p(\mathbf{X}|\mathbf{A})\ p(\mathbf{A}|\alpha)\ \mathrm{d}\mathbf{A}. \tag{4.52}$$

The above integration is hard to track analytically, and requires a *Taylor* expansion or some sampling approach as Monte Carlo methods to approximate the integrand, the joint distribution $p(\mathbf{A}, \mathbf{X}|\alpha)$.

The model can be further improved by fitting a distribution over the hyperparameter α, since this parameter represent the inverse of the variance, the values in α should be positive and therefore a *Gamma distribution* (Appendix B), which belong to the *exponential family* (section 3.2.1), is an appropriate choice. Drawing a distribution over α makes it possible to integrate over it and choose a more appropriate value instead of taking the α that maximizes the likelihood. The new model can be seen in figure 4.5(b), where we introduce new hyperparameters a_α and b_α. The prior distribution of α can be then expressed as:

$$\begin{aligned} p(\alpha) &= \prod_k^d p(\alpha_k) \\ &= \prod_k^d \mathcal{G}(\alpha_k, a_k, b) \\ &= \prod_k^d \frac{b^{a_k}}{\Gamma(a_k)}\alpha^{a_k-1}\exp(-b\alpha_k). \end{aligned} \tag{4.53}$$

The new model is too complex and makes the marginalization over the parameters too difficult to be solved analytically, as was the case in the two precedents ones. Thus fighting the problem of inferring more complex model (e.g. large d in FA) inherited in ML, by maximizing the penalized ML (or MAP) instead, requires interchanging

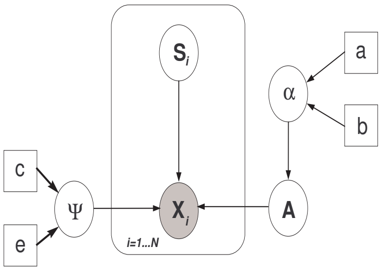


Figure 4.6: This figure shows the Bayesian net for VB – FA and VB – PPCA model. As in figure 4.5 the model correspond to VB – FA when the noise is hyper-elliptical with a diagonal precision matrix $\mathbf{\Psi}$, and corresponds to VB – PPCA when the noise is hyper-spherical with a single precision ψ parameter. This distribution is controlled by the deterministic hyperparameters k and e.

the deterministic parameter of interest with a random variable. To control the parameter's distribution a hyperparameter is introduced, hence increasing the hierarchical model. Further increase in the hierarchical model becomes rapidly hard to solve using the standard EM, due to the intractable required integrations (e.g. (4.52)). One should remember here that the resulting MAP model (technique) is still a point estimate (4.49), which is also a victim of overfitting (see e.g., figure 3.2). Lets see what *Variational Bayes* can do about these problems.

4.3.1 Variational FA and PPCA

In the previous section we saw that, increasing the flexibility of the FA model, and incorporating controlling terms, to get a the penalized or MAP model, makes the parameter estimation hard to solve, since they require some analytically intractable integrations. *Variational techniques* are used to approximate such integrals. Furthermore MAP, like ML, is still point estimate and therefore suffers from overfitting. In the Bayesian approach the whole parameter posterior $p(\mathbf{\Theta}|\mathbf{X})$ is taken into account, and when inferring parameters, the average wrt. to this posterior, over all parameter space is performed. The second moments are then taking as *uncertainty* of these inferences, and can be showed as error bars.
Variational Bayes incorporate both variational techniques and Bayesian approach to optimize directly the accuracy of the approximate posterior distribution [46].
While in EM we maximized the lower bound $\mathcal{F}(\mathcal{Q}(\mathbf{S}), \mathbf{\Theta})$ of the incomplete log likelihood $\mathcal{L}_{inc}(\mathbf{\Theta})$ (3.38). In VB we maximize the lower bound $\mathcal{F}(\mathcal{Q}(\mathcal{H}))$ of the log evidence $\ln p(\mathbf{X}|\mathcal{M}_m)$(3.52)

$$\begin{aligned}\ln p(\mathbf{X}|\mathcal{M}_m) &= \mathcal{F}(\mathcal{Q}(\mathcal{H})) + \mathrm{KL}(\mathcal{Q}(\mathcal{H})||p(\mathcal{H}|\mathbf{X})) \\ &\geq \mathcal{F}(\mathcal{Q}(\mathcal{H})),\end{aligned} \tag{4.54}$$

where $\mathcal{H}$ represent all hidden random variables, i.e. $\mathcal{H} = \{\mathbf{S}, \mathbf{A}, \alpha, \mathbf{\Psi}\}$. The Bayesian nets representation of variational FA/PPCA is shown in figure 4.6, where $\mathbf{\Psi}$ in the case of VB – FA (ψ for VB – PPCA) is Gamma distributed (Appendix B) random variable, controlled by the hyperparameter k and e, this choice is due to the fact that the diagonal elements of $\mathbf{\Psi}$ (ψ for VB – PPCA) represent the precisions of the noise model, and therefore should be greater than zeros, furthermore to be used in VB, $p(\mathbf{\Psi}|c,e)$ ($p(\psi|c,e)$) must be a conjugate prior. The prior distribution of α, $\mathbf{S}$ and $\mathbf{A}$ was presented in the previous section and they are all of them chosen to conjugate priors:

$$\begin{aligned}
p(\mathbf{S}) &= \mathcal{N}(\mathbf{S};\mathbf{0},I_d) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{s}_i;0,I_d) && (4.55)\\
p(\mathbf{A}|\alpha) &= \mathcal{N}(\mathbf{A};\mathbf{0},\alpha) = \prod_{k=1}^{d} \mathcal{N}(\mathcal{A}_k;0,\alpha_k) && (4.56)\\
p(\alpha|a,\mathbf{b}) &= \mathcal{G}(\alpha;a,\mathbf{b}) = \prod_{k=1}^{d} \mathcal{G}(\alpha;a,\mathbf{b}_k) && (4.57)\\
\text{FA} \Rightarrow p(\mathbf{\Psi}|c,\mathbf{e}) &= \mathcal{G}(\mathbf{\Psi};c,\mathbf{e}) = \prod_{j=1}^{d_X} \mathcal{G}(\psi_j;c,\mathbf{e}_j) && (4.58)\\
\text{PPCA} \Rightarrow p(\psi|c,e) &= \mathcal{G}(\psi;c,e). && (4.59)
\end{aligned}$$

It was earlier assumed the factorization of the variational distributions of $\mathbf{\Theta}$ and $\mathbf{S}$, here this factorization is extend to include $\mathbf{\Psi}$, hence

$$\mathcal{Q}(\mathcal{H}) = \mathcal{Q}(\mathbf{S})\mathcal{Q}(\mathbf{A},\alpha)\mathcal{Q}(\mathbf{\Psi}).$$

This choice is supported by structure of the model (see figure 4.6), where $\mathbf{\Psi}$ is conditionally independent (given the data) on the rest of the random variables, i.e., there is no direct arc between them.
The solution to each of the individual $\mathcal{Q}(\mathcal{H}_i)$ distribution that maximizes the free energy $\mathcal{F}(\mathcal{Q}(\mathcal{H}))$, can be found using equation (3.54) or equivalently (3.55) as:

$$\begin{aligned}
\ln \mathcal{Q}(\mathcal{H}_l) &= \langle \ln p(\mathbf{X},\mathcal{H}|\mathcal{M}_m)\rangle_{Q_{k\neq l}} + \mathrm{k}\\
&= \langle \ln p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{\Psi},\alpha|\mathcal{M}_m)\rangle_{Q_{k\neq l}} + \mathrm{k}, && (4.60)
\end{aligned}$$

where $\langle\cdot\rangle_{Q_{k\neq l}}$ denotes expectation taken with respect to all distributions except $\mathcal{Q}(\mathcal{H}_l)$.Using the Bayesian net in figure 4.6 we can easily express the the joint distribution of everything, for a FA model $p(\mathbf{X},\mathcal{H}|\mathcal{M}_m) = p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{\Psi},\alpha|\mathcal{M}_m)$ as:

$$\begin{aligned}
p(\mathbf{X},\mathcal{H}|\mathcal{M}_m) &= [\prod_{i}^{N} p(\mathbf{x}_i|\mathbf{s}_i,\mathbf{A},\mathbf{\Psi})p(\mathbf{s}_i)]\ p(\mathbf{A}|\alpha)p(\alpha|\mathbf{a},b)p(\mathbf{\Psi}|\mathbf{c},d) && (4.61)\\
\Rightarrow \ln p(\mathbf{X},\mathcal{H}|\mathcal{M}_m) &= \sum_{i}^{N} \ln p(\mathbf{x}_i|\mathbf{s}_i,\mathbf{A},\mathbf{\Psi}) + \sum_{i}^{N} \ln p(\mathbf{s}_i) + \ln p(\mathbf{A}|\alpha) + \ln p(\alpha) + \ln p(\mathbf{\Psi})
\end{aligned}$$

$$
\begin{aligned}
= \quad & \frac{N}{2}\ln|\mathbf{\Psi}| - \frac{1}{2}\sum_{i}^{N}(\mathbf{x}_i - \mathbf{A}\mathbf{s}_i)^\top\mathbf{\Psi}(\mathbf{x}_i - \mathbf{A}\mathbf{s}_i) \\
& -\frac{1}{2}\sum_{i}^{N}\mathbf{s}_i^\top\mathbf{s}_i \\
& +\frac{d_X}{2}\sum_{k}^{d}\ln\alpha_k - \frac{1}{2}\sum_{k}^{d}\alpha_k\mathcal{A}_k^\top\mathcal{A}_k \\
& +\sum_{k}^{d}(a_k-1)\ln\alpha_k - \sum_{k}^{d} b\ \alpha_k \\
& +\sum_{j}^{d_X}(c_j-1)\ln\psi_j - \sum_{j}^{d_X} e\ \psi_j \quad + \mathrm{k}.
\end{aligned} \tag{4.62}
$$

By setting $d_X = 1$ in the last line on the above equation, corresponding to a single noise precision, the joint probability for PPCA model, can be recovered.

4.3.2 Density estimation

By taking the expectation of (4.62) with respect to $Q_k(\mathcal{H}_k)$ for all $k \neq l$ we obtain the expression for $Q_l(\mathcal{H}_l)$ that is the closest to the posterior distribution $p(\mathcal{H}_l|\mathbf{X}, \mathbf{\Psi})$, in terms of minimizing the previous mentioned kullback-Liebler divergence. Proceeding this way we are going to give the expression for the needed functions namely $\{\mathcal{Q}(\mathbf{S}), \mathcal{Q}(\mathbf{A}), \mathcal{Q}(\alpha), \mathcal{Q}(\mathbf{\Psi})\}$. Since we are using conjugate prior, each variational posterior $\mathcal{Q}(\mathcal{H}_i)$ will have the same form as its prior $p(\mathcal{H}_i)$, see equation (3.28) in section 3.2.1

$\underline{\mathcal{Q}(\mathbf{S})\textbf{:}}$

To solve for $\mathcal{Q}(\mathbf{S})$ we take the expectation of (4.62) wrt. $\{\mathcal{Q}(\mathbf{A}), \mathcal{Q}(\alpha), \mathcal{Q}(\mathbf{\Psi})\}$, this stage can be regarded as the E step in the VBEM algorithm

$$
\begin{aligned}
\langle p(\mathbf{X},\mathcal{H})\rangle_{\mathcal{Q}_\mathbf{A}\mathcal{Q}_\alpha\mathcal{Q}_\mathbf{\Psi}} \quad &= \quad -\frac{1}{2}\sum_{i}^{N}\left(\mathbf{x}_i^\top\langle\mathbf{\Psi}\rangle\mathbf{x}_i - 2\mathbf{s}_i^\top\langle\mathbf{A}^\top\rangle\langle\mathbf{\Psi}\rangle\mathbf{x}_i + \mathbf{s}_i^\top\langle\mathbf{A}^\top\mathbf{\Psi}\mathbf{A}\rangle\mathbf{s}_i\right) \\
& \quad\quad -\frac{1}{2}\sum_{i}^{N}\mathbf{s}_i^\top\mathbf{s}_i \\
&= \quad -\frac{1}{2}\sum_{i}^{N}\left[\mathbf{x}_i^\top\langle\mathbf{\Psi}\rangle\mathbf{x}_i - 2\mathbf{s}_i^\top\langle\mathbf{A}^\top\rangle\langle\mathbf{\Psi}\rangle\mathbf{x}_i + \mathbf{s}_i^\top\left(I + \langle\mathbf{A}^\top\mathbf{\Psi}\mathbf{A}\rangle\right)\mathbf{s}_i\right].
\end{aligned}
$$

We can see that the above expectation is quadratic in $\mathbf{s}_i$. By using (4.55) we can express $Q(\mathbf{S})$ as:

$$
\begin{aligned}
Q(\mathbf{S}) \quad &= \quad \prod_{i}^{N} Q(\mathbf{s}_i) \\
&= \quad \prod_{i}^{N}\mathcal{N}\left(\mathbf{s}_i; \mu_\mathbf{S}^{(i)}, \Sigma_\mathbf{S}\right),
\end{aligned} \tag{4.63}
$$

where

$$
\begin{aligned}
\Sigma_{\mathbf{S}} &= \left(I_d + \langle \mathbf{A}^\top \mathbf{\Psi} \mathbf{A} \rangle\right)^{-1}; &(4.64)\\
\mu_{\mathbf{S}}^{(i)} &= \Sigma_{\mathbf{S}} \langle \mathbf{A}^\top \rangle \langle \mathbf{\Psi} \rangle \mathbf{x}_i, &(4.65)
\end{aligned}
$$

where for FA model, $\mathbf{\Psi}$ is diagonal and hence

$$
\begin{aligned}
\langle \mathbf{A}^\top \mathbf{\Psi} \mathbf{A} \rangle &= \langle \sum_j^{dx} \psi_j \mathcal{A}_j \mathcal{A}_j^\top \rangle \\
&= \sum_j^{dx} \langle \psi_j \rangle \langle \mathcal{A}_j \mathcal{A}_j^\top \rangle, &(4.66)
\end{aligned}
$$

where $\mathcal{A}_j$ is a column vector corresponding to the j'th row. For a single precision parameter in PPCA, this expectation becomes:

$$
\langle \mathbf{A}^\top \mathbf{\Psi} \mathbf{A} \rangle = \langle \mathbf{A}^\top \psi I_d \mathbf{A} \rangle = \langle \psi \rangle \langle \mathbf{A}^\top \mathbf{A} \rangle. \quad (4.67)
$$

$\underline{\mathcal{Q}(\alpha)}$**:**

To solve for $\mathcal{Q}(\alpha)$ we take the expectation of (4.62) wrt. $\{\mathcal{Q}(\mathbf{S}), \mathcal{Q}(\mathbf{A}), \mathcal{Q}(\mathbf{\Psi})\}$

$$
\begin{aligned}
\langle \ln p(\mathbf{X}, \mathcal{H}) \rangle_{\mathcal{Q}_{\mathbf{S}} \mathcal{Q}_{\mathbf{A}} \mathcal{Q}_{\mathbf{\Psi}}} &= \frac{d}{2} \sum_k^d \ln \alpha_k - \frac{1}{2} \sum_k^d \alpha_k \langle \|\mathcal{A}_k^\top\|^2 \rangle \\
&\quad + \sum_k^d (a-1) \ln \alpha_k - \sum_k^d b_k \alpha_k + \mathrm{k} \\
&= \sum_k^d (a + \frac{d}{2} - 1) \ln \alpha_k - \sum_k^d \left(b_k + \frac{\langle \|\mathcal{A}_k^\top\|^2 \rangle}{2} \right) \alpha_k + \mathrm{k}, &(4.68)
\end{aligned}
$$

where k in the above equation includes all the terms not involving the variable of interest (α in this case). Using (4.60) together with (4.57) we get

$$
\begin{aligned}
Q(\alpha) &= \prod_k^d Q(\alpha_k) \\
&= \prod_k^d \mathcal{G}\left(\alpha_k; \tilde{a}, \tilde{b}_k\right),
\end{aligned}
$$

where

$$
\begin{aligned}
\tilde{a} &= [a + \frac{d}{2}]; &(4.69)\\
\tilde{b}_k &= [b_k + \frac{\langle \|\mathcal{A}_k^\top\|^2 \rangle}{2}]. &(4.70)
\end{aligned}
$$

Note that we used a single hyperparameter a to control the *shape* of all d columns in $\mathbf{A}$. This is a heuristic choice, based on several experiments, though not shown in the report. However it can easily be extended.

$Q_{\mathbf{A}}(\mathbf{A})$:

As in the case of inferring $Q(\alpha)$, we take the expectation of (4.62), but this time with respect to $\{\mathcal{Q}(\mathbf{S}), \mathcal{Q}(\alpha), \mathcal{Q}(\mathbf{\Psi})\}$. We begin by rewriting (4.62) to a scalar form and retaining only the terms involving $\mathbf{A}$ we get:

$$\begin{aligned}
\ln p(\mathbf{X}, \mathcal{H}) &= -\frac{1}{2}\sum_i^N \sum_j^{dx} \psi_j (x_{i,j} - \mathcal{A}_j^\top \mathbf{s}_i)^2 - \frac{1}{2}\sum_j^{dx} \mathcal{A}_j^\top \alpha \mathcal{A}_j + \mathrm{k} \\
&= -\frac{1}{2}\sum_i^N \sum_j^{dx} \psi_j (x_{i,j}^2 - 2x_{i,j}\mathcal{A}_j^\top \mathbf{s}_i + \mathcal{A}_j^\top \mathbf{s}_i \mathbf{s}_i^\top \mathcal{A}_j) - \frac{1}{2}\sum_j^{dx} \mathcal{A}_j^\top \alpha \mathcal{A}_j + \mathrm{k} \\
&= -\frac{1}{2}\sum_j^{dx} \psi_j \left[\sum_i^N x_{i,j}^2 - 2\mathcal{A}_j^\top \left(\sum_i^N x_{i,j}\mathbf{s}_i \right) + \mathcal{A}_j^\top \left(\sum_i^N \mathbf{s}_i \mathbf{s}_i^\top + \frac{1}{\psi_j}\alpha \right) \mathcal{A}_j \right] + \mathrm{k}
\end{aligned} \tag{4.71}$$

Where $\mathcal{A}_j$ represent a column vector corresponding to the j'th row of $\mathbf{A}$. The expectation of (4.71) can be written as:

$$\begin{aligned}
\langle \ln p(\mathbf{X}, \mathcal{H}) \rangle_{\mathcal{Q}_{\mathbf{S}}\mathcal{Q}_\alpha \mathcal{Q}_{\mathbf{\Psi}}} &= -\frac{1}{2}\sum_j^{dx} \langle \psi_j \rangle \left[\sum_i^N x_{i,j}^2 - 2\mathcal{A}_j^\top \left(\sum_i^N x_{i,j} \langle \mathbf{s}_i \rangle \right) \right. \\
&\quad \left. + \mathcal{A}_j^\top \left(\sum_i^N \langle \mathbf{s}_i \mathbf{s}_i^\top \rangle + \frac{1}{\langle \psi_j \rangle} \langle \alpha \rangle \right) \mathcal{A}_j \right] + \mathrm{k}.
\end{aligned} \tag{4.72}$$

Using (4.60) and the fact that the above expectation is quadratic in $\mathcal{A}_j$, we can infer $Q(\mathbf{A})$ as:

$$\begin{aligned}
Q(\mathbf{A}) &= \prod_j^{dx} Q(\mathcal{A}_j) \\
&= \prod_j^{dx} \mathcal{N}\left(\mathcal{A}_j; \mu_{\mathcal{A}}^{(j)}, \Sigma_{\mathcal{A}}^{(j)} \right),
\end{aligned} \tag{4.73}$$

where for FA

$$\Sigma_{\mathcal{A}}^{(j)} = \left(\langle \psi_j \rangle \sum_i^N \langle \mathbf{s}_i \mathbf{s}_i^\top \rangle + \langle \mathbf{A} \rangle \right)^{-1}; \tag{4.74}$$

$$\mu_{\mathcal{A}}^{(j)} = \langle \psi_j \rangle \Sigma_{\mathcal{A}}^{(j)} \left(\sum_i^N \mathbf{x}_{i,j} \langle \mathbf{s}_i \rangle \right), \tag{4.75}$$

and for PPCA

$$\Sigma_{\mathbf{A}} = \left(\langle \psi \rangle \sum_i^N \langle \mathbf{s}_i \mathbf{s}_i^\top \rangle + \langle \mathbf{A} \rangle \right)^{-1}; \tag{4.76}$$

$$\mu_{\mathcal{A}}^{(j)} = \langle \psi \rangle \Sigma_{\mathbf{A}} \left(\sum_i^N \mathbf{x}_{i,j} \langle \mathbf{s}_i \rangle \right). \tag{4.77}$$

$\underline{\mathcal{Q}(\boldsymbol{\Psi})\textbf{:}}$

Following the same steps as before, we write down the expectation of (4.62) wrt. $\{\mathcal{Q}(\mathbf{S}), \mathcal{Q}(\mathbf{A}), \mathcal{Q}(\alpha)\}$:

$$\begin{aligned}
\mathrm{FA}: \langle \ln p(\mathbf{X}, \mathcal{H})\rangle_{\mathcal{Q}_\mathbf{S}\mathcal{Q}_\mathbf{A}\mathcal{Q}_\alpha} &= \sum_j^{d_X} \big(c + \frac{N}{2} - 1\big) \ln \psi_j + \mathrm{k} \\
&\quad - \sum_j^{d_X} \big(e_j + \frac{1}{N}\mathrm{DIAG}_j[\sum_i^N \langle (\mathbf{x}_i - \mathbf{A}^\top \mathbf{s}_i)(\mathbf{x}_i - \mathbf{A}^\top \mathbf{s}_i)^\top \rangle]\big)\psi_j \quad (4.78) \\
\mathrm{PPCA}: \langle \ln p(\mathbf{X}, \mathcal{H})\rangle_{\mathcal{Q}_\mathbf{S}\mathcal{Q}_\mathbf{A}\mathcal{Q}_\alpha} &= \big(c + \frac{N d_X}{2} - 1\big) \ln \psi + \mathrm{k} \\
&\quad - \big(e + \frac{1}{N}\mathrm{TR}[\sum_i^N \langle (\mathbf{x}_i - \mathbf{A}^\top \mathbf{s}_i)(\mathbf{x}_i - \mathbf{A}^\top \mathbf{s}_i)^\top \rangle]\big)\psi. \quad (4.79)
\end{aligned}$$

This corresponds to a the log of a Gamma distribution, of the following form:

$$Q(\boldsymbol{\Psi}) = \prod_j^{d_X} Q(\psi_j) = \prod_j^{d_X} \mathcal{G}\left(\psi_j; \tilde{c}, \tilde{e}_j\right),$$

as in the case of a (4.69), c is unique for all diagonal elements. The updates are

$$\begin{aligned}
\mathrm{FA}: \tilde{c} &= [c + \frac{N}{2}] && (4.80) \\
\tilde{\mathbf{e}} &= \mathbf{e} + \frac{1}{2}\mathrm{DIAG}[\sum_i^N \langle (\mathbf{x}_i - \mathbf{A}^\top \mathbf{s}_i)(\mathbf{x}_i - \mathbf{A}^\top \mathbf{s}_i)^\top \rangle] && (4.81) \\
\mathrm{PPCA}: \tilde{c} &= [c + \frac{N}{2} d_X] && (4.82) \\
\tilde{e} &= e + \frac{1}{2}\mathrm{TR}[\sum_i^N \langle (\mathbf{x}_i - \mathbf{A}^\top \mathbf{s}_i)(\mathbf{x}_i - \mathbf{A}^\top \mathbf{s}_i)^\top \rangle]. && (4.83)
\end{aligned}$$

Note that $\mathbf{e}$ in FA is a d_X-dimensional vector, while in PPCA is a scalar variable. The presented formula for the parameters of the variational posterior distribution, should be run, given some data, until some stop criteria is achieved. In the codes I made, the criteria was a combination of two criteria, a max number of iterations n_{max} and when the free energy $\mathcal{F}_{M_m}$ is stable in a range of iteration, i.e., If $\Delta\mathcal{F} = \mathcal{F}_n - \mathcal{F}_{n-1}$ falls bellow some critical value ϵ then convergence can be assumed. However it is not easy to define such simple thresholds that scales appropriately with both model complexity and size of data set.

Now that two learning algorithms ML and VB for the two models FA and PPCA are presented, lets look at a simple toy data example, where the performance of (VB) FA vs. (VB) PPCA in capturing information from data is tested.

Example 4.2 *Consider a 4-dimensional Gaussian toy data set* $\mathbf{X} = \{\mathbf{x}_j\}_{j=1}^4$ *of size* $N = 100$*, where* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *are uncorrelated with each other and the rest of variables in the data set.* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *have a small and large variances respectively,* $\mathbf{x}_3$ *and* $\mathbf{x}_4$ *are highly negative correlated and have almost the same variance. a small variance Gaussian*

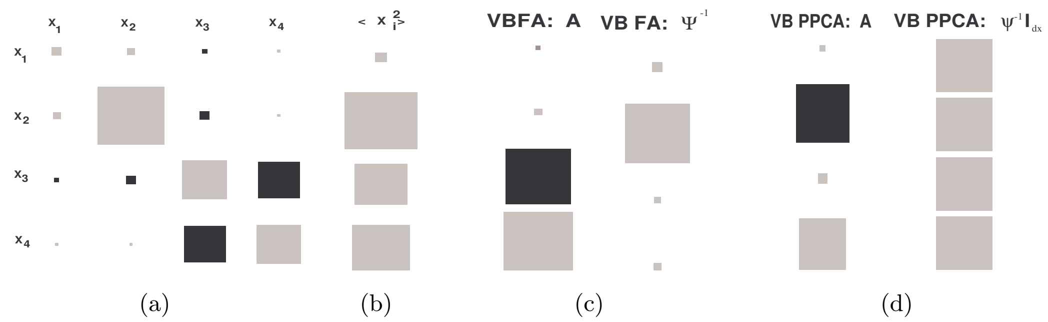


Figure 4.7: This figure is based on example (4.2), where a $4D$ Gaussian toy data of size $N = 100$ was generated. The correlation between the different variables is shown in (a), note the high negative correlation between $\mathbf{x}_3$ and $\mathbf{x}_4$. (b) shows the diagonal elements of (a) (sample variance). The data is fitted to FA (c) and PPCA (d). It can easily be seen that FA with its diagonal noise precision is able to find both the noisy component $\mathbf{x}_2$ shown as large rectangle in the right column of (c), and the large negative correlation between $\mathbf{x}_3$ and $\mathbf{x}_4$ left column in (c). PPCA gets confused by the large variance in $\mathbf{x}_2$ and finds correlation between the wrong components $\mathbf{x}_2$ and $\mathbf{x}_4$. The latent dimensionality in both VBFA and VBPPCA was set to a fixed value 1.

noise is added to the data set $\mathbf{X}$. *All the quantities of interest are shown as Hinton diagrams in figure 4.7, where positive and negative values corresponds to gray and black squares respectively, and their absolute values corresponds to the area. The covariance matrix of the data set can be seen in (a), and its diagonal in (b). One can see that the* $\mathbf{x}_2$ *is much noisier than the rest of variables. The* VBFA[5] *model with a* 1 *dimensional latent variable, is fitted to data. With its diagonal noise precision* VBFA *succeeds in modelling the noisy component* $\mathbf{x}_2$. *The diagonal element of* $\mathbf{\Psi}^{-1}$ *are shown in column to the right in (c).* VBFA *finds also the negative large correlation between the two correlated components* $\mathbf{x}_3$ *and* $\mathbf{x}_4$, *shown in the left column in (c).* PPCA *with its single precision noise is confused by the high variance in* $\mathbf{x}_2$ *and finds a correlation between* $\mathbf{x}_2$ *and* $\mathbf{x}_4$, *shown in the left column in (d). The* PPCA *noise estimate is the average of the estimated noise in the four direction (right column in (d)). The failure of* PPCA *compering to* FA *in extracting information in noisy data, can be roughly explained by the fact that* PPCA *does pay attention to both variance and covariance, whereas* FA *only pays attention to the covariance [37, 8].*

Other properties such as finding the effective latent space, and how $\mathbf{A}$ relates to the principle components of data in both PPCA and FA will be seen in the applications chapter.
Note that since we are dealing with a single factor analyzer, inferring the mean can be expressed as in equation (4.3) in the case of ML, and as (4.10) in VB case, and could be subtracted from data before performing our analysis. For brevity of the model and analysis I did not include it here, but it will be unavoidable in the case of *mixture models*.

[5] Due to the large size of data $N = 100$ wrt. the dimension $d_X = 4$, both EMFA and VBFA gives almost the same result, furthermore the aim is to compare FA with PPCA. Therefor only VB is considered here.

CHAPTER 5

Mixture Models

In the previous section it was shown how data which lies on a linear subspace can be modelled through single Gaussian or linear latent variable models, such as FA and PPCA. These models can not handle multi-modal data, this was shown for the case of a single Gaussian (figure 4.3(d)) in section 4.1. It is obvious that the linear models FA and PPCA will face the same problem.
A more flexible model can be obtained by considering *mixture models*. Mixture models can be shown as a weighted average of M simple components densities $p(\mathbf{x}|\mathbf{\Theta}_m)$

$$p(\mathbf{x}|\mathbf{\Theta}) \quad = \quad \sum_{m=1}^{M} \pi_m p_m(\mathbf{x}|\theta_m), \tag{5.1}$$

where π_m is the *mixing proportion* of the the m'th component, and satisfies $\pi_m \geq 0$ and $\sum_{m=1}^{M} \pi_m = 1$. This guarantees that $p(\mathbf{x}|\mathbf{\Theta})$ is a valid density function. $p_m(\mathbf{x}|\theta_m)$ is the d_X-dimensional density model corresponding to the m'th component, and controlled by the set of parameters θ_m. Given the expression of mixture models (5.1), several interesting models can be obtained depending on the choice of the form for the component densities $p_m(\mathbf{x}|\theta_m)$:

$$\begin{aligned}
\mathrm{GMM} \Rightarrow p_m(\mathbf{x}|\theta_m) &\sim \mathcal{N}(\mathbf{X}; \mu_m, \mathbf{\Psi}_{xm}^{-1}) && (5.2)\\
\mathrm{MFA} \Rightarrow p_m(\mathbf{x}|\theta_m) &\sim \mathcal{N}(\mathbf{X}; \mu_m, \mathbf{A}_m\mathbf{A}_m^{\top} + \mathbf{\Psi}^{-1}) && (5.3)\\
\mathrm{MPPCA} \Rightarrow p_m(\mathbf{x}|\theta_m) &\sim \mathcal{N}(\mathbf{X}; \mu_m, \mathbf{A}_m\mathbf{A}_m^{\top} + \psi^{-1} I_{d_{\mathbf{s}}}), && (5.4)
\end{aligned}$$

where GMM, MFA and MPPCA stands for *Gaussian Mixture Models*,*Mixture of Factor Analyzers* and *Mixture of Probabilistic Principle Components* respectively. These are the models discussed in the following sections. Learning these models using ML requires the computing the (log) likelihood function $\mathcal{L}_{inc}(\mathbf{\Theta})$. Consider an observed data set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ independently drawn from the mixture distribution in (5.1). $\mathcal{L}_{inc}(\mathbf{\Theta})$ is given by

$$\begin{aligned}
p(\mathbf{X}|\mathbf{\Theta}) &= \prod_{i=1}^{N}[\sum_{m=1}^{M} \pi_m p_m(\mathbf{x}_i|\theta_m)]\\
\Rightarrow \mathcal{L}_{inc}(\mathbf{\Theta}) &= \ln p(\mathbf{X}|\mathbf{\Theta}) = \sum_{i=1}^{N} \ln[\sum_{m=1}^{M} \pi_m p_m(\mathbf{x}_i|\theta_m)]. && (5.5)
\end{aligned}$$

A direct maximization of the above quantities is a hard task. However, re-interpreting the mixture model as a latent variable model, by introducing an unobserved *state* $\mathbf{z}$, makes it possible to use EM algorithm to infer the parameters of interest as discussed in section 3.3.2, where the complete data log likelihood $\mathcal{L}_c(\mathbf{\Theta}) = \ln p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})$ is maximized instead of the incomplete data log likelihood $\mathcal{L}_{inc}(\mathbf{\Theta}) = p(\mathbf{X}|\mathbf{\Theta})$. The hidden variables $\mathbf{z}_i$ can be expressed as an M dimensional vector with 1 in the element whose index m corresponds to the selected m'the mixture component, and the rest are set to 0 [7] as

$$\mathbf{z}_i = \underbrace{[0\ 0\ \cdots 0\ 1\ 0 \cdots 0\ 0]}_{M\ elements}{}^{\top},$$

where $p(z_{i,m} = 1) = \pi_m$ in equation (5.5), and $\mathbf{Z} = \{z_{i,m}\}_{i=1,m=1}^{N,M}$ is an $[M \times N]$ matrix, when it is considered as discrete variable. Thus, it can be thought of $\mathbf{z}_i$ as a binary way to write the index of the component that gave rise to the data point $\mathbf{x}_i$. A nice property of this way of expressing $\mathbf{z}_i$, is $\sum_i^N \mathbf{z}_{i,m} = N_m$, where N_m is the number of data points in the m'th component. Conditional on $\mathbf{z}_m$, the data points are assumed to be independently drawn from the distribution of the data given the parameters of the m'th component θ_m. $\mathcal{L}_c(\mathbf{\Theta})$ can now be expressed as follows:

$$\begin{aligned}
\mathcal{L}_c(\mathbf{\Theta}) &= \ln p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta}) \\
&= \ln \prod_i^N p(\mathbf{x}_i, \mathbf{z}_i|\mathbf{\Theta}) \\
&= \ln \prod_i^N \prod_m^M [p(\mathbf{x}_i|z_{i,m} = 1, \mathbf{\Theta}) p(z_{i,m} = 1)]^{z_{i,m}} \\
&= \sum_i^N \sum_m^M z_{i,m} [\ln p(\mathbf{x}_i|z_{i,m} = 1, \mathbf{\Theta}) + \ln \pi_m].
\end{aligned} \tag{5.6}$$

Bayesian Learning on the other hand requires the introduction of prior distribution (see section 3.2) over the parameters as well as the mixing proportions. Since all the used models (5.2-5.59) are defined together with their required priors in the previous chapter, except the mean μ which was neglected for simplicity, where it is just the sample mean in the case of ML, and the Bayesian case was dealt with in example (4.1). In mixture model the mean can not be computed separately and added at last as was the case in e.g., FA or PPCA. A conjugate prior of μ_m could be the normal distribution

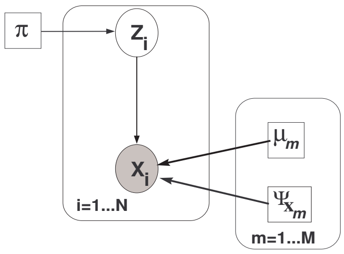
$$p(\mu) = \prod_m^M \mathcal{N}(\mu_m|\mathbf{m}_m, \mathbf{V}_m) \tag{5.7}$$

where $\mathbf{m}_m$ and $\mathbf{V}_m$ are the the mean and the precision respectively for the prior of the mean μ_m of the m'th component. Since in this chapter I use the general case of multivariate GMM, the precision $\mathbf{\Psi}_m$ is now a $[d_X \times d_X]$ matrix, thus, the prior $p(\mathbf{\Psi}_m)$ is the Wishart distribution (see Appendix B)

$$\mathbf{\Psi}_m \sim \mathcal{W}(\mathbf{\Psi}_m; \mathbf{C}_m, \mathbf{D}_m), \tag{5.8}$$

where Gamma is the univariate special case.

Now I only need to introduce the priors over the new variables. Another way of writing $p(z_{i,m} = 1) = \pi_m$ [41] is



(a)

Figure 5.1: This figure shows the Bayesian net for ML Gaussian Mixture Model (GMM), μ_m and $\mathbf{\Psi}_m$ are the mean and the precision of m'th Gaussian component $p(\mathbf{X}|\theta_m)$, **pi** is the set of mixing proportions. Note that the extension from a single Gaussian consisted of adding the mixing proportions and the plate over the model parameters.

$$
\begin{aligned}
p(\mathbf{z}_i|\pi) &= \prod_{m=1}^{M} \pi_m^{z_{i,m}} \\
(\mathbf{Z} \sim \mathrm{IID}) \Rightarrow p(\mathbf{Z}|\pi) &= \prod_{i=1}^{N}\prod_{m=1}^{M} \pi_m^{z_{i,m}}, \qquad (5.9)
\end{aligned}
$$

(5.10)

or equivalently, by using $N_m = \sum_i^N \mathbf{z}_{i,m}$:

$$p(\{N_m\}_{m=1}^M|\pi) = \prod_{i=1}^{N} \binom{N}{N_1, \cdots, N_M} \prod_{m=1}^{M} \pi_m^{N_m}, \qquad (5.11)$$

where the (5.9) follows from the fact that $\mathbf{Z}$ is IID. Both equations (5.9) and (5.11) are *multinomial distribution* with parameter π. The *Binomial distribution* is the special case where $M = 2$ as seen in example (2.2). A conjugate prior for π is the *Dirichlet distribution*:

$$p(\pi|\mathbf{u}) \sim \mathbf{D}(\pi;\mathbf{u}) = \frac{\Gamma(\sum_m u_m)}{\prod_m \Gamma(u_m)} \prod_m \pi_m^{u_m - 1}, \qquad (5.12)$$

where u_m ($\mathbf{u} = \{u_m\}_{m=1}^M$) can be interpreted as a virtual count for value m, before seen $\mathbf{Z}$. The *Beta distribution* mentioned in example (2.2) is the special case when $M = 2$, and represent a conjugate prior for the Binomial distribution. The properties of Dirichlet and Beta are described in APPENDIX B.

5.1 Gaussian Mixture Models

GMM and its extension has been used by many researchers and leads to interesting works, e.g., in text modelling [21], in modelling annotated data [4]. In the next subsection I introduce ML of GMM's.

5.1.1 ML Gaussian Mixture Models

Bayesian net for ML learning of GMMs is shown in figure 5.1. ML learning can be performed using EM algorithm (section 3.3.2), where the M step corresponds to maximizing the expected complete data log likelihood $\langle \mathcal{L}_c(\mathbf{\Theta}) \rangle$, wrt. the variational distribution of the hidden states $\mathcal{Q}(\mathbf{Z})$ (3.45), using equation (5.6) this expectation can be expressed as:

$$\langle \mathcal{L}_c(\mathbf{\Theta}) \rangle_{\mathcal{Q}(\mathbf{Z})} = \sum_i^N \sum_m^M \langle z_{i,m} \rangle [\ln p(\mathbf{x}_i | z_{i,m} = 1, \mathbf{\Theta}) + \ln \pi_m]. \tag{5.13}$$

M step: in the M step we maximize the quantity in (5.13) wrt. to the parameter of interest, i.e., $\mathbf{\Theta} = \{\pi_m, \mu_m, \mathbf{\Psi}_m\}$.

Differentiating $\langle \mathcal{L}_c(\mathbf{\Theta}) \rangle$ wrt. μ_m

$$\frac{\partial \langle \mathcal{L}_c(\mathbf{\Theta}) \rangle}{\partial \mu_m} = \sum_i^N \langle z_{i,m} \rangle \frac{\partial}{\partial \mu_m} \ln p(\mathbf{x}_i | z_{i,m} = 1, \mathbf{\Theta}) = \mathbf{0}, \tag{5.14}$$

where $p(\mathbf{x}_i | z_{i,m} = 1, \mathbf{\Theta})$ is the following normal distribution:

$$\begin{aligned} p(\mathbf{x}_i | z_{i,m} = 1, \mathbf{\Theta}) &= \mathcal{N}(\mathbf{x}_i; \mu_m, \mathbf{\Psi}_{X_m}^{-1}) \\ &= \frac{|\mathbf{\Psi}_{X_m}|^{1/2}}{(2\pi)^{d_X/2}} \exp[-\frac{1}{2}(\mathbf{x}_i - \mu_m)^\top \mathbf{\Psi}_{X_m} (\mathbf{x}_i - \mu_m)]. \end{aligned} \tag{5.15}$$

The partial derivative of the logarithm of the above equation can be computed as

$$\frac{\partial}{\partial \mu_m} \ln p(\mathbf{x}_i | z_{i,m} = 1, \mathbf{\Theta}) = (\mathbf{x}_i - \mu_m)^\top \mathbf{\Psi}_{X_m}, \tag{5.16}$$

where I used the relation $\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^\top \mathbf{A} \mathbf{x} = \mathbf{x}^\top (\mathbf{A} + \mathbf{A}^\top)$. Substituting the above result into equation (5.14), gives:

$$\mu_m = \frac{\sum_i^N \langle z_{i,m} \rangle \mathbf{x}_i}{\sum_i^N \langle z_{i,m} \rangle}. \tag{5.17}$$

Following the same steps $\mathbf{\Psi}_{X_m}$ can be inferred to be

$$\mathbf{\Psi} X_m = \frac{\sum_i^N \langle z_{i,m} \rangle (\mathbf{x}_i - \mu_m)(\mathbf{x}_i - \mu_m)^\top}{\sum_i^N \langle z_{i,m} \rangle}. \tag{5.18}$$

A careful look at equations (5.17) and (5.18) it can be seen that these result are similar to focusing on a single component (m'th here) and computing the sample mean and sample covariance.

In order to maximize the quantity in equation (5.13) wrt. π_m, one should keep in mind that π should sum to 1, i.e., $\sum_m^M \pi_m = 1$. To maintain this constraint I used Lagrange multiplier, to augment equation (5.13) as follows:

$$\mathcal{C}(\mathbf{\Theta}) = \langle \mathcal{L}_c(\mathbf{\Theta}) \rangle - \lambda(\sum_m^M -1). \tag{5.19}$$

Differentiating $\mathcal{C}(\mathbf{\Theta})$ wrt. each π_m gives:

$$\frac{\partial}{\partial \pi_m}\mathcal{C}(\mathbf{\Theta}) - \lambda = 0, \qquad for\ 1 \leq m \leq M$$

which result in $\lambda = N$, which used to get the update

$$\pi_m = \frac{\sum_i^N \langle z_{i,m} \rangle}{N}. \tag{5.20}$$

E step: To make sure that the incomplete data log likelihood (which the quantity we are truly interested in maximizing) is maximized while maximizing the complete data log likelihood. The expectation should taken wrt. the posterior distribution of the hidden state $p(\mathbf{Z}|\mathbf{X}, \mathbf{\Theta})$. Hence the expectations used in the M step $\langle z_{i,m} \rangle$ should be computed as follows:

$$\begin{aligned}
\langle z_{i,m} \rangle_{p(\mathbf{Z}|\mathbf{X},\mathbf{\Theta})} &= 1 \cdot p(z_{i,m} = 1|\mathbf{x}_i, \mathbf{\Theta}) + 0 \cdot p(z_{i,m} = 1|\mathbf{x}_i, \mathbf{\Theta}) \\
&= p(z_{i,m} = 1|\mathbf{x}_i, \mathbf{\Theta}) \\
&= \frac{p(\mathbf{x}_i|z_{i,m} = 1, \mathbf{\Theta})p(z_{i,m} = 1)}{\sum_{m'}^M p(\mathbf{x}_i|z_{i,m'} = 1, \mathbf{\Theta})p(z_{i,m'} = 1)} \\
&= \frac{p(\mathbf{x}_i|z_{i,m} = 1, \mathbf{\Theta})\pi_m}{\sum_{m'}^M p(\mathbf{x}_i|z_{i,m'} = 1, \mathbf{\Theta})\pi_{m'}}.
\end{aligned} \tag{5.21}$$

As mentioned earlier, the EM Algorithm is just a nice way to simplify the math behind the intractable maximization of ML. Hence the overfitting problems inherited in the point estimate ML are still there. GMM's are extremely flexible and simply maximizing the $\mathcal{L}_{inc}(\mathbf{\Theta})$ will lead to "infinite overfit" [21]. Given an N large data set $\mathbf{X}$, a possible scenario of such overfitting occurs when $\mu_m = \mathbf{x}_m$ for $m = 1, \cdots, M-1$, and the corresponding covariances shrinks to zero matrix (or equivalently $\mathbf{\Psi}_{X_m}$ grows to infinite matrix). The last (M'th) Gaussian is then fitted to the remaining $N - M + 1$ data points. The generalization error becomes here roughly equal to the single "background" Gaussian [21]. Thus, increasing M results in increasing ML until the limit ($M = N$), where each data point is fitted by a single Gaussian. Appropriate model selection criterions was introduced earlier in section 3.1 AIC and in section 3.2 BIC[1]. These criterions will be tested later on, when the VB for GMM is described. First lets look at simple example where the model in figure 5.1 is learnt using ML.

Example 5.1 *Consider a Mixture of* 5 *Gaussians, where* $N_m = 100$ *data points was sampled from each one of them, i.e* $N = \sum_m^M N_m = 500$. MLGMM *in figure 5.1 was used to estimate the probability density function* $p(\mathbf{X}|\mathcal{M}_j)$ *of the observed data* $\mathbf{X}$. *As expected, during learning, the expected complete data log likelihood* $< \mathcal{L}_c(\mathbf{\Theta}|\mathcal{M}_j) >$ *is always a lower to the incomplete data log likelihood* $\mathcal{L}_{inc}(\mathbf{\Theta}|\mathcal{M}_j)$ *(see section 3.3.2), this can be seen in figure 5.2(d), where both quantities increases until convergence. Figure 5.2(a)*

[1]Even if BIC follows as a limiting case of the VB framework, due to its simple form it can be included in ML directly.

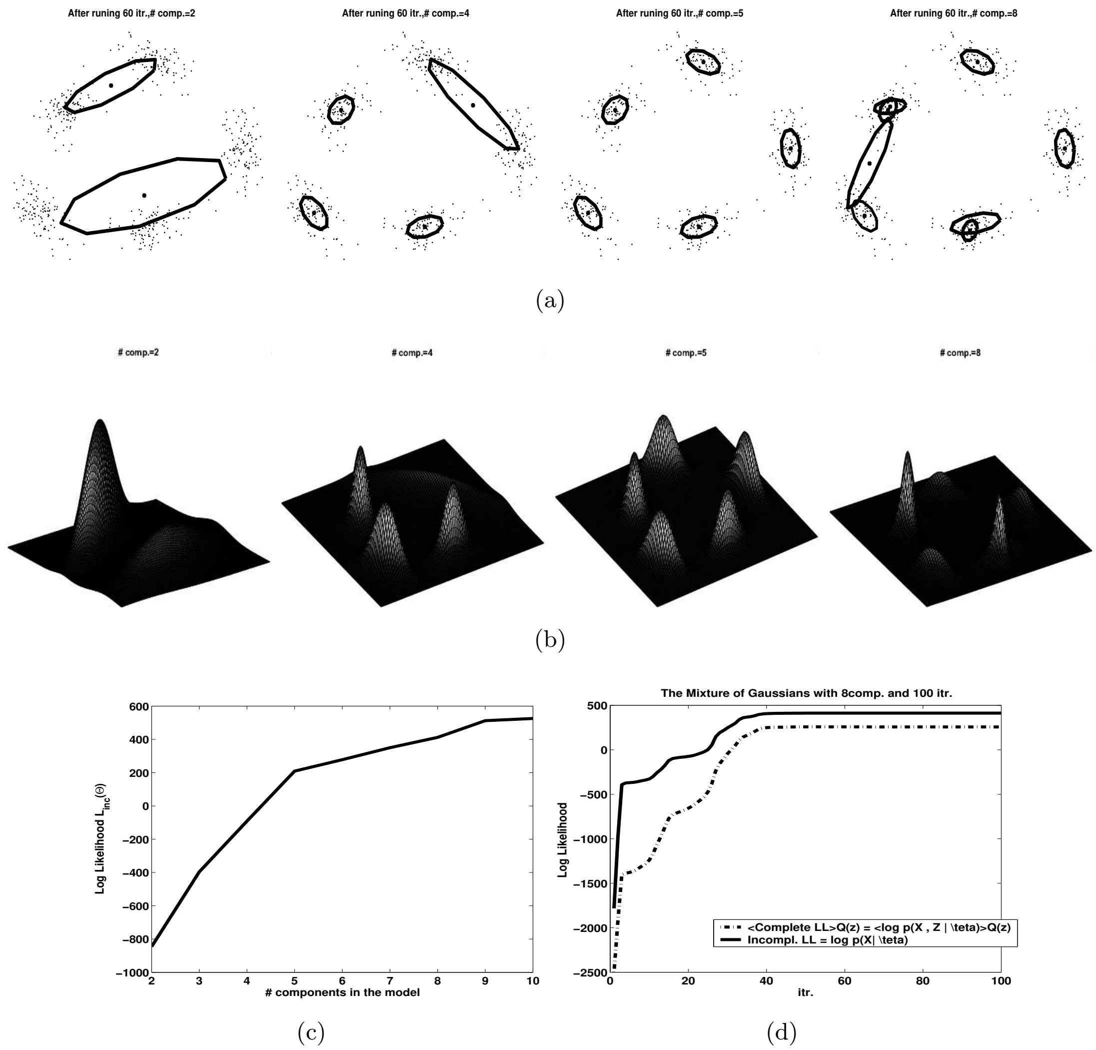
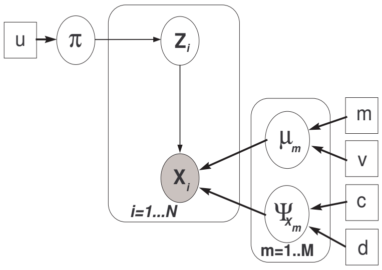


Figure 5.2: This figure shows the ability of the GMM's to find clusters or groups in the observed data given a data set (a). The observed data set $\mathbf{X}$ is generated from a mixture model with 5 mixtures, each of which has independently generated $N_m = 100$ data points i.e., $N_{total} = 500$. (b) shows how GMM can be used as density estimators in unsupervised learning. (c) shows the likelihood scores of several mixture models from $M = 2$, where $(\mathcal{L}_{inc}(\mathbf{\Theta}|\mathcal{M}_2) = -845.2855)$ to $M = 10$, where $(\mathcal{L}_{inc}(\mathbf{\Theta}|\mathcal{M}_{10}) = 525.5203)$. This shows that the likelihood just keeps on growing even for models larger then the true one. This is in fact what we discussed in section 3.1, to be the inherited problem of overfitting, the point estimate maximum likelihood suffers from. Note that these models was learned using EM, which shows that EM has no positive effect on this problem, but it is just a way to solve the intractability of maximizing $\mathcal{L}_{inc}(\mathbf{\Theta})$. The last figure (d) shows how in learning $\langle\mathcal{L}_c(\mathbf{\Theta})\rangle$ is always a lower bound to $\mathcal{L}_{inc}(\mathbf{\Theta}|\mathcal{M})$.

shows the result of 4 *models, with* 2*,* 4*,* 5 *and* 8 *mixtures, are fitting to the same data set. The likelihood score of each of them can be readen from the graph in (c), where it can be seen that the likelihood keeps on increasing even after reaching the true number of mixtures (i.e.,* 5*), hence, the figure represent an example of the overfitting problem inherited in the point estimate* ML*. Figure 5.2(b) shows the resulting estimate of the probability density function* $p(\mathbf{X}|\mathcal{M}_j)$*, obtained by marginalizing the complete data likelihood (*$p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta}, \mathcal{M})$*) over the mixture indicators* $\mathbf{z}_m$ *as follows:*



(a)

Figure 5.3: This figure shows the Bayesian net for VB Gaussian Mixture Model (GMM), μ_m and $\mathbf{\Psi}_m$ are the mean and the precision of m'th Gaussian component $p(\mathbf{X}|\theta_m)$, **pi** is the set of mixing proportions. Note that the extension from a single Gaussian consisted of adding the mixing proportions and the plate over the model parameters.

$$p(\mathbf{X}|\mathbf{\Theta}, \mathcal{M}_j) = \sum_m^M p(\mathbf{x}_i|z_{i,m} = 1, \mathbf{\Theta}, \mathcal{M}_j)p(z_{i,m} = 1|\mathcal{M}_j).$$

5.1.2 Bayesian Gaussian Mixture Models

The Bayesian net for VB GMM is shown in figure 5.3. Thus the joint distribution of all the random variables $\mathcal{H} = \{\mathcal{H}_k\}_k^K$, conditioned on the model $\mathcal{M}_j$, is given by

$$\begin{aligned}
p(\mathbf{X}, \mathcal{H}|\mathcal{M}_j) &= p(\mathbf{X}, \mu, \mathbf{\Psi}_x, \mathbf{Z}, \pi|\mathcal{M}_j) \\
&= [\prod_i^N p(\mathbf{x}_i|\mu, \mathbf{\Psi}_x, \mathbf{z}_i)p(\mathbf{z}_i|\pi)]\ p(\pi|\mathbf{u})\ [\prod_m^M p(\mu_m)p(\mathbf{\Psi}_{x_m})] \\
&= [\prod_i^N \prod_m^M \left(p(\mathbf{x}_i|\mu_m, \mathbf{\Psi}_{x_m}, z_{i,m}|\pi_m)p(z_{i,m}|\pi_m)\right)^{z_{i,m}}] \\
&\quad \cdot p(\pi|\mathbf{u})\ [\prod_m^M p(\mu_m|\mathbf{m}_m, \mathbf{V}_m)p(\mathbf{\Psi}_{x_m}|\mathbf{C}, \mathbf{D})] \\
\Rightarrow \ln p(\mathbf{X}, \mu, \mathbf{\Psi}_x, \mathbf{Z}, \pi|\mathcal{M}_j) &= \sum_i^N \sum_m^M z_{i,m} \ln[p(\mathbf{x}_i|\mu_m, \mathbf{\Psi}_{x_m}, z_{i,m}|\pi_m)p(z_{i,m}|\pi_m)] \\
&\quad + \ln p(\pi|\mathbf{u}) + \sum_m^M \ln[p(\mu_m|\mathbf{m}_m, \mathbf{V}_m)p(\mathbf{\Psi}_{x_m}|\mathbf{C}, \mathbf{D})]. \quad (5.22)
\end{aligned}$$

The definition of the model is completed by defining conjugate priors over the parameters

$$\begin{aligned}
p(\mu_m) &= \mathcal{N}(\mu_m; \mathbf{m}_m, \mathbf{V}^{-1}) && (5.23) \\
p(\mathbf{\Psi}_m) &= \mathcal{W}(\mathbf{\Psi}_m; \mathbf{C}_m, \mathbf{D}) && (5.24) \\
p(\mathbf{Z}|\pi) &= \text{Multin}(\mathbf{Z}; \pi) && (5.25) \\
p(\pi) &= \text{Dirichlet}(\pi|\mathbf{u}). && (5.26)
\end{aligned}$$

A separable prior over μ and $\mathbf{\Psi}$ is chosen for simplicity. The use of NORMAL-WISHART prior would mean that the variational posterior over these two parameters would not

be separable, leading to a slightly improved approximation. However, using a separable prior will still model the interesting area of high probability, as seen in the univariate case in example (4.1).
VB learning involves maximizing the lower bound $\mathcal{F}(\mathcal{Q}(\mathcal{H}))$ (3.52), where $\mathcal{H} = \{\mathbf{Z}, \mu, \mathbf{\Psi}, \pi\}$, which is convenient to rewrite here

$$\begin{aligned}
p(\mathbf{X}|\mathcal{M}) &= \int \mathcal{Q}(\mathcal{H}) \ln \frac{p(\mathbf{X}, \mathcal{H}|\mathcal{M}_j)}{\mathcal{Q}(\mathcal{H})} \, \mathrm{d}\mathcal{H} + \int \mathcal{Q}(\mathcal{H}) \ln \frac{\mathcal{Q}(\mathcal{H})}{p(\mathcal{H}|\mathbf{X}, \mathcal{M}_j)} \, \mathrm{d}\mathcal{H} \\
&= \mathcal{F}(\mathcal{Q}(\mathcal{H})) + \mathrm{KL}(\mathcal{Q}(\mathcal{H})||p(\mathcal{H}|\mathbf{X})) \qquad (5.27) \\
&\geq \mathcal{F}(\mathcal{Q}(\mathcal{H})) \qquad (5.28) \\
&= \int \mathcal{Q}(\mathcal{H}) \ln p(\mathbf{X}, \mathcal{H}|\mathcal{M}_j) \, \mathrm{d}\mathcal{H} - \int \mathcal{Q}(\mathcal{H}) \ln \mathcal{Q}(\mathcal{H}) \, \mathrm{d}\mathcal{H} \\
&= \langle \ln p(\mathbf{X}, \mathcal{H}|\mathcal{M}_j) \rangle_{\mathcal{Q}} - \int \mathcal{Q}(\mathcal{H}) \ln \mathcal{Q}(\mathcal{H}) \, \mathrm{d}\mathcal{H}, \qquad (5.29)
\end{aligned}$$

where we assume $\mathcal{Q}(\mathcal{H})$ factorizes over subsets $\{\mathcal{H}_k\}$ of the variables in $\mathcal{H}$, so that

$$\begin{aligned}
\mathcal{Q}(\mathcal{H}) &= \prod_k \mathcal{Q}_k(\mathcal{H}_k) \\
&= \mathcal{Q}(\mu)\mathcal{Q}(\mathbf{\Psi})\mathcal{Q}(\mathbf{Z}|\pi)\mathcal{Q}(\pi). \qquad (5.30)
\end{aligned}$$

Maximizing the free energy or minimizing the KL divergence in equation (5.27) results in equation (3.54), repeated here for convenience

$$\mathcal{Q}_k(\mathcal{H}_k) = \frac{\exp < \ln p(\mathbf{X}, \mathcal{H}|\mathcal{M}_j) >_{\mathcal{Q}_{l \neq k}}}{\int \exp < \ln p(\mathbf{X}, \mathcal{H}|\mathcal{M}_j) >_{\mathcal{Q}_{l \neq k}} \, \mathrm{d}\mathcal{H}_k}. \qquad (5.31)$$

Due to the conjugacy of the priors, when optimized the factors of $\mathcal{Q}$ have the same form as the corresponding factors in the priors.

$$\begin{aligned}
\mathcal{Q}(\mu_m) &= \mathcal{N}(\mu_m; \tilde{\mathbf{m}}_m, \tilde{\mathbf{V}}^{-1}) \qquad (5.32) \\
\mathcal{Q}(\mathbf{\Psi}_m) &= \mathcal{W}(\mathbf{\Psi}_m; \tilde{\mathbf{C}}_m, \tilde{\mathbf{D}}) \qquad (5.33) \\
\mathcal{Q}(\mathbf{z}_i|\pi) &= \mathrm{Multin}(\mathbf{z}_i|\tilde{\pi}) \qquad (5.34) \\
\mathcal{Q}(\pi) &= \mathcal{MD}(\pi|\tilde{\mathbf{u}}). \qquad (5.35)
\end{aligned}$$

Following equation (5.31), to infer $\mathcal{Q}(\mathcal{H})$ the expectation of the complete data log likelihood[2] in equation (5.60) wrt. all the variational posteriors except the one of interest, i.e., $\mathcal{Q}(\mathcal{H}_k)$. Thinking of the learning as VBEM, the updates in the VBE step, can be expressed as[3]:

VBE step: This step consist of inferring $\mathcal{Q}(\mathbf{Z})$

$$\begin{aligned}
(5.34) \Rightarrow \mathcal{Q}(\mathbf{z}_i|\pi) &= \mathrm{Multin}(\mathbf{z}_i; \tilde{\pi}) \\
&= \prod_i^N \prod_m^M \tilde{\pi}_{i,m}^{z_{i,m}}, \qquad (5.36)
\end{aligned}$$

[2]As mentioned earlier $p(\mathcal{H}|\mathcal{M})$ can also be regarded as $\mathcal{L}_c(\mathcal{M})$, since it is the joint distribution of the observed data and all the random variables, given a "deterministic" model.
[3]The variables with "tilde" ($\tilde{x}$) represent the updates of the initial value x.

where

$$\begin{aligned}\tilde{\pi}_{i,m} &= \frac{\tilde{\tau}_{i,m}}{\sum_n^N \tilde{\tau}_{i,n}} \\ &= \frac{\exp[\frac{1}{2}\langle \ln|\mathbf{\Psi}_m|\rangle - \frac{1}{2}\mathrm{TR}[\langle \mathbf{\Psi}_m\rangle\langle(\mathbf{x}_i-\mu_m)(\mathbf{x}_i-\mu_m)^\top\rangle] + \langle \ln \pi_m\rangle]}{\sum_n^N \tilde{\tau}_{i,n}}.\end{aligned}$$

The expected value of the indicator is

$$\langle z_{i,m}\rangle = \tilde{\pi}_{i,m}. \tag{5.37}$$

VBM step: In this step the sufficient statistics of the variational distributions, over all the hidden random variables is updated

$\underline{\mu}$**:**

$$\begin{aligned}(5.32) \Rightarrow \mathcal{Q}(\mu) &= \mathcal{N}(\mu; \tilde{\mathbf{m}}, \tilde{\mathbf{V}}) \\ &= \prod_m^M \mathcal{N}(\mu_m; \tilde{\mathbf{m}}_m, \tilde{\mathbf{V}}_m),\end{aligned} \tag{5.38}$$

where

$$\tilde{\mathbf{V}}_m = \mathbf{V}_m + \langle \mathbf{\Psi}_m\rangle \sum_i^N \langle z_{i,m}\rangle \tag{5.39}$$

$$\tilde{\mathbf{m}}_m = \tilde{\mathbf{V}}_m^{-1}\big(\mathbf{V}_m\mathbf{m}_m + \langle \mathbf{\Psi}_m\rangle \sum_i^N \langle z_{i,m}\rangle \mathbf{x}_i\big). \tag{5.40}$$

To get an uninformative prior for μ_m it suffice to set the precision $\mathbf{V}_m$ to a very small value, corresponding to a flat distribution, this explanation is supported by the update formula above, for small value of $\mathbf{V}_m$ only the data that decide the updates. The expected value and the covariance of the variational distribution over the mean $\langle \mu_m\rangle$ and $\mathrm{cov}(\mu_m)$ respectively are

$$\langle \mu_m\rangle = \tilde{\mathbf{m}}_m \tag{5.41}$$

$$\mathrm{cov}(\mu_m) = \langle \mu_m\mu_m^\top\rangle - \langle \mu_m\rangle\langle \mu_m\rangle^\top \tag{5.42}$$

$$= \tilde{\mathbf{V}}_m. \tag{5.43}$$

$\underline{\mathbf{\Psi}}$**:**

$$\begin{aligned}(5.33) \Rightarrow \mathcal{Q}(\mathbf{\Psi}) &= \mathcal{W}(\mathbf{\Psi}; \tilde{\mathbf{C}}, \tilde{\mathbf{D}}) \\ &= \prod_m^M \mathcal{W}(\mathbf{\Psi}_m; \tilde{\mathbf{C}}_m, \tilde{\mathbf{D}}_m),\end{aligned} \tag{5.44}$$

where

$$
\begin{aligned}
\tilde{\mathbf{C}}_m &= \mathbf{C}_m + \sum_i^N \langle z_{i,m} \rangle \\
&= \mathbf{C}_m + \tilde{N}_m && (5.45) \\
\tilde{\mathbf{D}}_m &= \mathbf{D}_m + [\sum_i^N \langle z_{i,m} \rangle \langle (\mathbf{x}_i - \mu_m)(\mathbf{x}_i - \mu_m)^\top \rangle]. && (5.46)
\end{aligned}
$$

The expected value of the variational distribution over the precision matrix $\langle \mathbf{\Psi}_m \rangle$ is

$$
\langle \mathbf{\Psi}_m \rangle = \tilde{\mathbf{C}}_m \tilde{\mathbf{D}}_m^{-1}. \quad (5.47)
$$

<u>π:</u>

$$
\begin{aligned}
(5.35) \Rightarrow \mathcal{Q}(\pi) &= \mathcal{D}(\pi; \tilde{\mathbf{u}}) \\
&= \prod_m^M \mathcal{W}(\mathbf{\Psi}_m; \tilde{\mathbf{C}}_m, \tilde{\mathbf{D}}_m), && (5.48)
\end{aligned}
$$

where

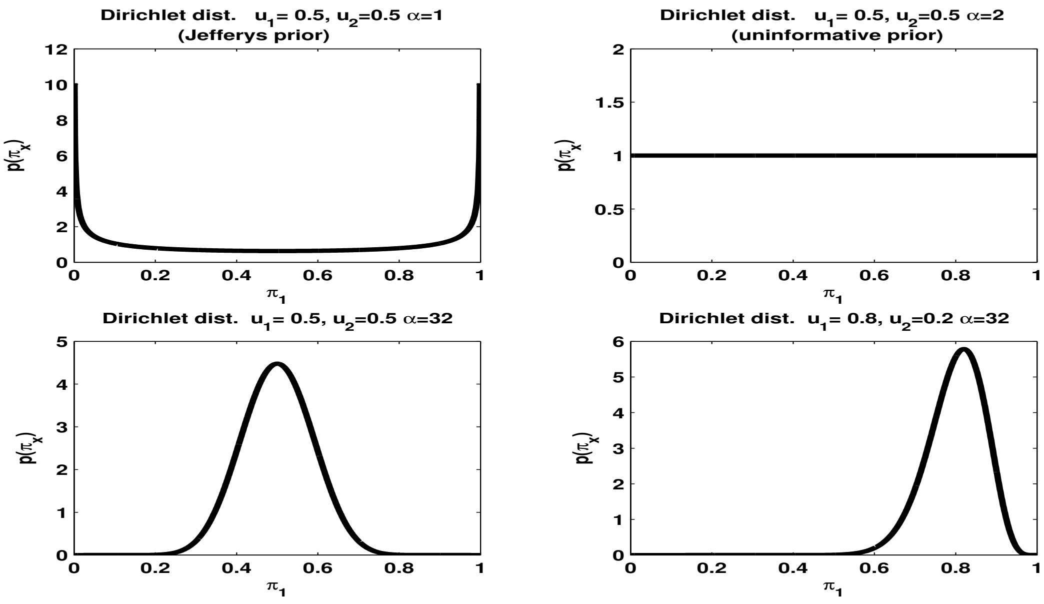
$$
\begin{aligned}
\tilde{u}_m &= u_m + \sum_i^N \langle z_{i,m} \rangle \\
&= \alpha \ddot{u}_m + \tilde{N}_m, && (5.49)
\end{aligned}
$$

where $\ddot{\mathbf{u}} = \{\ddot{u}_m\}_{m=1}^M$ and $\sum_m^M \ddot{u}_m = 1$. α is a scale parameter and its effect is discussed below. The expected value and the covariance of the variational distribution over the the mixing proportions $\langle \pi \rangle$ and $\mathrm{cov}(\pi)$, respectively are

$$
\begin{aligned}
\langle \pi \rangle &= \frac{\tilde{\mathbf{u}}}{\sum_m^M \tilde{u}_m} && (5.50) \\
\mathrm{cov}(\pi) &= \langle \pi \pi^\top \rangle - \langle \pi \rangle \langle \pi \rangle^\top \\
&= \frac{u_0 \mathrm{diag}(\tilde{\mathbf{u}}) - \tilde{\mathbf{u}} \tilde{\mathbf{u}}^\top}{u_0^2 (u_0 + 1)}, && (5.51)
\end{aligned}
$$

where $u_0 = \sum_m^M u_m$.
Note that the effect of Dirichlet $\mathcal{D}(\pi; \alpha \ddot{\mathbf{u}})$ choice for the prior distribution over α can be seen in equation (5.49), where for relatively large values of α (compare to N/M), data has no effect on the updating $\tilde{\mathbf{u}}$, and hence the mixing proportion $\langle \pi \rangle$ remains unchanged from the initialization, this can easily be shown by inserting a large α (and $u_m = 1/M, \ \forall \ m$) in (5.49) and (5.50) result in:

$$
\begin{aligned}
\tilde{\mathbf{u}} &\approx \alpha \ddot{\mathbf{u}} && (5.52) \\
\Rightarrow \langle \pi \rangle &\approx \frac{\alpha \ddot{\mathbf{u}}}{\alpha \sum_m^M \ddot{u}_m} \\
&= \ddot{\mathbf{u}},
\end{aligned}
$$



(a)

Figure 5.4: Plots of $p(\pi_1)$ of a two dimensional Dirichlet (i.e., Beta distr.). The plots are made for different $\alpha = \{1, 2, 4, 32\}$ and , $u_1 = u_2 = 1/M = .5$ for all plot except, down-right side figure. One should remember that Dirichlet is mainly used as a distribution over probabilities. Thus, in our case over the mixing proportion π_1 of component $m = 1$. The figure in top-right side is known as noninformative, and shows that the probability π_1 can take equally any value between 0 and 1. Jefferys prior on top-left side states that π_1 must be either probable or improbable, as it will be seen later in example (5.2), this type of priors result in a quick cancellation of the components. Figure in down-left side shows a informative prior, where α takes on a large value 32, this prior will force us to chose the initial mixing proportion, in this case $\pi_1 = .5$. Another informative case is shown in down-right side figure, where $u_1 = .8$, $u_2 = .2$ and $\alpha = 32$ the initial $\pi_1 = \frac{u_1}{u_1+u_2} = .8$ is the most probable. From these figures one can see that for α it acts as the precision of the distribution, this is even better seen when taking the *logit* of the distribution [14].

since $\sum_m^M \ddot{u}_m = 1$. This is of course nice if it is *a priori* known how many mixtures are wanted, but this is not the case in most practical unsupervised learning problems. In the Dirichlet distribution [14], α can be thought of as a measure of sharpness, similar to the precision in the Gaussian distribution. For large α (high precision) the distribution over π is sharply peaked around the mean $\ddot{\mathbf{u}}$. Thus for an uninformative prior over π, α should be smaller (law precision) i.e, broader distribution. The uninformative case is actually when $\alpha = 1$, for $\alpha < 1$ the Dirichlet prefer the extreme cases (i.e., 0 or 1). This is shown in figure 5.4, where I show the special case of the Dirichlet, for $M = 2$ (Beta distribution (Appendix B)). Note that the choice of the symbol α here, is to emphasize the similarity to α in FA, where it was used as the ARD factor, here though, it is $\alpha\ddot{u}_m$ which "shut down" the improbable m'th component. The effect of α in VB will be some how clearer in example (5.2), see also Appendix B.

5.1.3 Model order selection

In order to get a measure of a given model, we need to compute the negative free energy $\mathcal{F}_j$ of the particular model. Given the assumption in equation (5.31), $\mathcal{F}$ in equation (5.28) can be rewritten as:

$$
\begin{aligned}
\mathcal{F}(\mathcal{Q}(\mathcal{H})) &= \int \mathcal{Q}(\mathcal{H}) \ln \frac{p(\mathbf{X}, \mathcal{H}|\mathcal{M}_j)}{\mathcal{Q}(\mathcal{H})} \, \mathrm{d}\mathcal{H} \\
&= \int \mathcal{Q}(\mathcal{H}) \ln \frac{p(\mathbf{X}|\mathcal{H}, \mathcal{M}_j) p(\mathcal{H}|\mathcal{M}_j)}{\mathcal{Q}(\mathcal{H})} \, \mathrm{d}\mathcal{H} \\
&= \int \mathcal{Q}(\mathcal{H}) \ln p(\mathbf{X}|\mathcal{H}, \mathcal{M}_j) \, \mathrm{d}\mathcal{H} - \int \mathcal{Q}(\mathcal{H}) \ln \frac{\mathcal{Q}(\mathcal{H})}{p(\mathcal{H}|\mathcal{M}_j)} \, \mathrm{d}\mathcal{H} \\
&= \langle \ln p(\mathbf{X}|\mathcal{H}, \mathcal{M}_j) \rangle_{\prod_k \mathcal{Q}_k(\mathcal{H}_k)} - \int \prod_k^K \mathcal{Q}(\mathcal{H}_k) \sum_{k"}^K \ln \frac{\mathcal{Q}(\mathcal{H}_{k"})}{p(\mathcal{H}_{k"}|\mathcal{M}_j)} \, \mathrm{d}\mathcal{H} \\
&= \langle \ln p(\mathbf{X}|\mathcal{H}, \mathcal{M}_j) \rangle_{\prod_k \mathcal{Q}_k(\mathcal{H}_k)} - \sum_k^K \int \mathcal{Q}(\mathcal{H}_k) \ln \frac{\mathcal{Q}(\mathcal{H}_k)}{p(\mathcal{H}_k|\mathcal{M}_j)} \, \mathrm{d}\mathcal{H}_k \\
&= \langle \ln p(\mathbf{X}|\mathcal{H}, \mathcal{M}_j) \rangle_{\mathcal{Q}} - \sum_k^K \mathrm{KL}(\mathcal{Q}(\mathcal{H}_k) || p(\mathcal{H}_k|\mathcal{M}_j))
\end{aligned} \tag{5.53}
$$

Using equations (5.23-5.26) and (5.32-5.35) equation (5.53) becomes:

$$
\begin{aligned}
\mathcal{F} = &-\mathrm{KL}_{\mathcal{D}}(\tilde{\mathbf{u}}; \tilde{\mathbf{u}}) \\
&- \sum_m^M \mathrm{KL}_{\mathcal{W}}(\tilde{\mathbf{C}}_m, \tilde{\mathbf{D}}_m; \mathbf{C}_m, \mathbf{D}_m) \\
&- \sum_m^M \mathrm{KL}_{\mathcal{N}}(\tilde{\mathbf{m}}_m, \tilde{\mathbf{V}}_m; \mathbf{m}_m, \mathbf{V}_m) \\
&+ \sum_m^M L_{av}(m),
\end{aligned} \tag{5.54}
$$

where

$$
L_{av}(m) = \langle p(\mathbf{X}, \mathbf{Z}|\mu, \mathbf{\Psi}, \pi) \rangle \mathcal{Q} \tag{5.55}
$$

The KL_f if is the KL divergence for the distribution f distribution. The KL divergences for the distributions used in this report can be found in Appendix B.

Bayesian Information Criterion BIC

As mentioned before both BIC $= -$MDL are the limiting cases of the VB framework for large data set [1]. From equation (3.9) BIC can be written as

$$
\mathrm{BIC}(M) = \sum_i^N \ln p(\mathbf{x}_i|\mathbf{\Theta}) - \frac{K_M}{2} \ln N \tag{5.56}
$$

where K_M is the number of free parameters in the model with M mixtures. K_M in the case of GMM is given by

$$K_M = M(1 + d_X + \frac{d_X(d_X+1)}{2}). \tag{5.57}$$

The first term is for the mixing proportions, the second for means and the third for the precisions.
To get an understanding of the VB works, lets try to solve the same problem in example (5.1), this time with VB and see how it finds an appropriate number of mixtures.

Example 5.2 *Given the same data set as in example (5.1). The aim is to see how* VB *will select the number of components, this will be compared with the previously introduced* BIC. *The* VBGMM *model is shown as Bayesian net in figure 5.3. The hyperparameters in the model are initialized as follows:*

$$\begin{aligned}
\mathbf{m}_m &= \frac{1}{N}\sum_i^N \mathbf{X} \\
\mathbf{V}_m &= 100 I_d \\
\mathbf{D}_m &= .01 d_X I_{d_\mathbf{x}} \\
\mathbf{C}_m &= 1 d_X I_{d_\mathbf{x}} \\
\mathbf{u}_m &= \alpha/M \\
\alpha &= \{100, 10, \frac{1}{M}\},
\end{aligned}$$

these initializations are the same for all m components. I started with a large number of components $M = 20$. The VB *learning consist of* 100 *iterations. As the learning proceeds, the components that are not supported by the data will "automatically" be shot down. This is done in practice by removing the components with $\langle \pi_m \rangle < \epsilon$. The used choice is $\epsilon = .005\% \sum_m \langle \tilde{\pi}_m \rangle$, i.e., kind of vote, the candidates supported by fewer than $.005\%$ of the population are unwanted (and removed at the end of the learning). Figure 5.5(a)(b) and (c) corresponds to $\alpha = \{100, 10, 0.05\}$ respectively. Left side figure in (a) shows the valid components left after training. (a) (mid.) shows the hinton diagram for the last update of $\langle \pi \rangle$ where it can be seen the size of the mixing proportions for the components. (a) (right) is the history of $\langle \pi \rangle$, it can be seen here that it almost preserve its initial values, following the Dirichlet distribution the learning should end up choosing $\langle \pi \rangle = \ddot{\mathbf{u}}$ (5.50). While in (b) and (c) it can be seen how fast the components are cancelled out, until an appropriate number of components is found, when using small α values. This can be explained by the effect of Dirichlet figure 5.4(down-left) with small α values, where it prefers the extremes, i.e., either $p = 0$ or $p = 1$.*
Figure 5.6(top) shows how the lower bound $\mathcal{F}$ increases throughout the learning until convergence. The effect of cancellation of components on the lower bound $\mathcal{F}$, can be seen by looking at the similarities between the indices of peaks in (down) and the difference in the lower bound $\Delta\mathcal{F} = \mathcal{F}t - \mathcal{F}_{t-1}$ (mid.), it can be seen that high increases occurs exactly after each component have been cancelled out.
Finally in figure 5.6(b) it can be seen how both BIC *and the variational lower bound $\mathcal{F}$ succeeded in finding the true number of components. In this example the number of data is unchanged through the learning $N = 500$. It is interesting to see how the changing the number of data will affect the model selection, This will be seen later.*

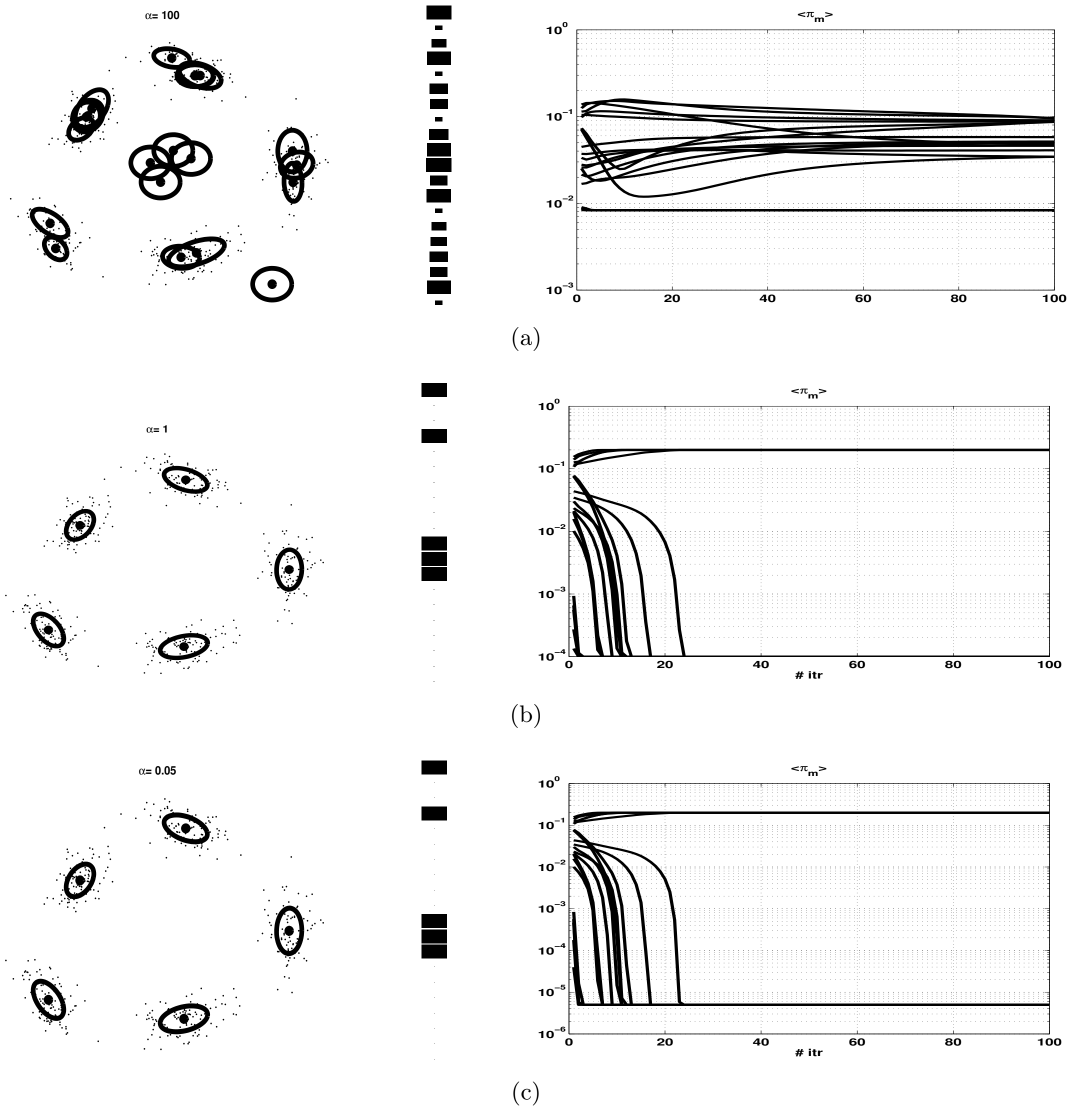
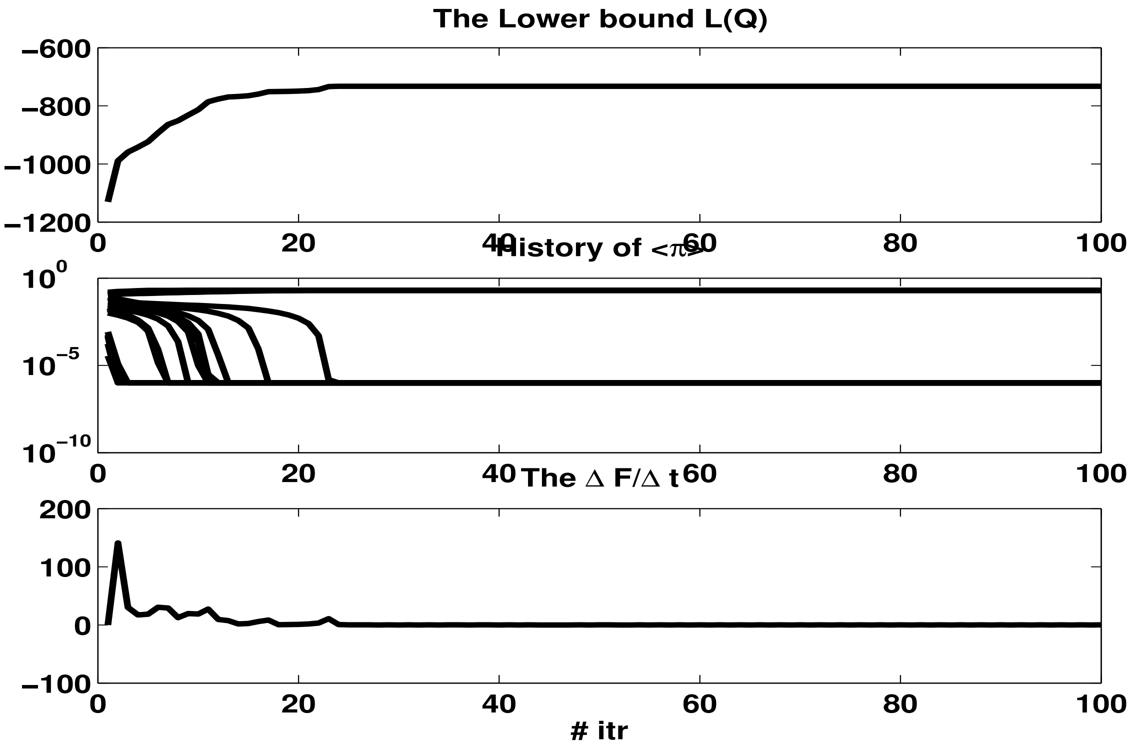


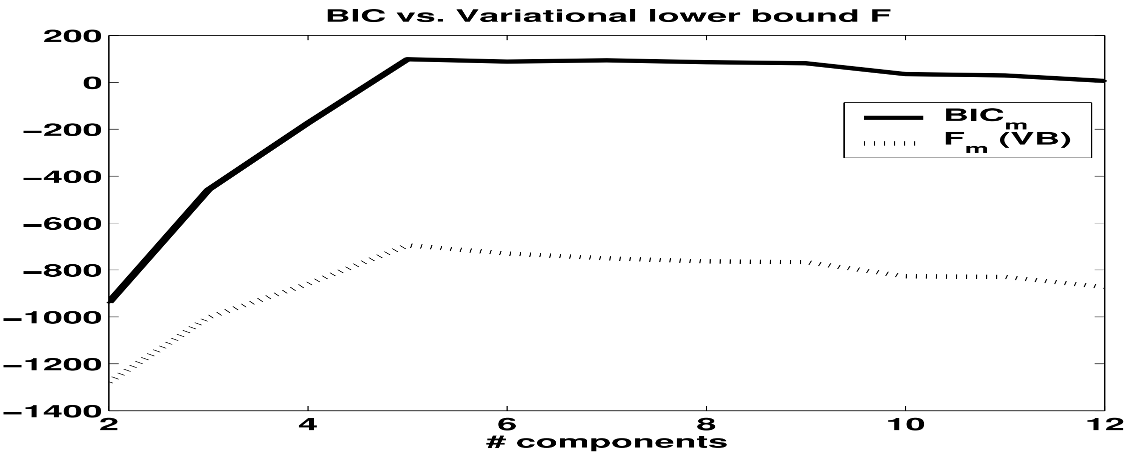
Figure 5.5: This figure shows how VB learning deals with the problem of finding model order. Given the same data set as in example (5.1). As in the case of FA the hidden space was mainly the work of the ARD. Here is the Dirichlet distributed mixing proportions π that has the similar task, of finding the intrinsic number of components. For large value of $\alpha = 100$ (a) the initial number of components remains unchanged. Note that since $\ddot{u}_m = 1/M = .05$ for all m, and $\langle \pi_{init} \rangle = \ddot{u}$ (5.50), if we learn the model longer time all $\langle \pi \rangle$ will converges to that value, as described by the Dirichlet distribution in figure 5.4 (down-left figure). However for smaller values, $\alpha = 10$ (b) or $\alpha = 1/M = .05$ (c) the model finds an appropriate number of components $\tilde{M} = M = 5$, already around $itr = 35$ right-side figure in (b) and (c), where the curves represent the values of $\langle \pi \rangle$ during learning. One can see how fast the values drops down for small values of α, this can be explained by figure 5.4 (top-left figure), where for small α the model prefer the extremes, i.e., 0 or 1. .

$\boxed{5.2}$ Mixture of Factor Analysis and PCA

An important advantage of linear latent variable models discussed in chapter 4 is that they define a proper probability model which can be extended to mixture model. As was the case in the previous section, where the extension from a single Gaussian to mixture of Gaussians (GMM) was by, roughly speaking interchanging $p_m(\mathbf{x}|\theta_m)$ in equation (5.1)



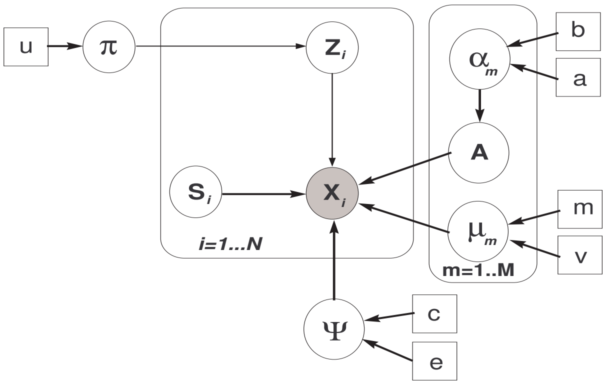
(a)



(b)

Figure 5.6: This figure shows how the lower bound $\mathcal{F}$ increases (a) until convergence. In (b) I showed the history of updating the mixing proportions $\langle\pi\rangle$, where $\alpha = 0.01$. In (b) the difference in $\Delta\mathcal{F} = \mathcal{F}_{t+1} - \mathcal{F}_t$ is plotted. Note that high increases occurs when components are cancelled out, this can be seen by looking at similarity of time (itr.) indices for tops in $\Delta\mathcal{F}/\Delta t$ with indices for cancellations in the history of $\langle\pi\rangle$. (b) shows model selection problem, for the same data used in examples (5.2) and (5.1). The x axis represent the number of components in each M_j. It can clearly be seen that both BIC and VB using the lower bound, finds the "true" latent number of mixtures, namely 5 .

by the Gaussian distribution $\mathcal{N}(\mathbf{x}; \mu_m, \mathbf{\Psi}_{x_m}^{-1})$, as in equation (5.2). For *Mixture of Factor Analyzers* equation (5.58) is used, i.e., a linear combination of component distributions, repeated here for convenience:



(a)

Figure 5.7: This figure shows Bayesian net for Variational Bayes *Mixture Factor Analysers*. Note that I did not insert the noise precision into inside the plate, since I assume the noise to be sensor noise, i.e, the same noise for all the components. Note also that their is two hidden state in this model $\mathbf{S}$ (from the linear FA) and $\mathbf{Z}$ (from mixture distributions). As in the case of linear latent variable models, by constraining the noise precision $\mathbf{\Psi}$ to $\sigma^{-2}I_{d_\mathbf{x}}$ a *mixture of probabilistic* PCA, is obtained.

$$\begin{aligned} \text{MFA} \Rightarrow p_m(\mathbf{x}|\theta_m) &\sim \mathcal{N}(\mathbf{X};\mu_m, \mathbf{A}_m\mathbf{A}_m^\top + \mathbf{\Psi}^{-1}) && (5.58)\\ \text{MPPCA} \Rightarrow p_m(\mathbf{x}|\theta_m) &\sim \mathcal{N}(\mathbf{X};\mu_m, \mathbf{A}_m\mathbf{A}_m^\top + \psi^{-1}I_{d_\mathbf{s}}), && (5.59)\end{aligned}$$

As described before for spherical precision $\mathbf{\Psi}$ the model is referred to as *Mixture of Probabilistic PCA* (MPCA) [43], if the noise precision is elliptical the model is then called a *Mixture of Factor Analysers* [18]. These mixtures can be interpreted as a mixture of constrained Gaussians in which the number of parameters can be controlled through the dimension of the latent space d_S without putting too strong constraints on the flexibility of the model, that is, on the form of the precision matrix.
Bayesian net for VBMFA is illustrated in figure 5.7. All the needed quantity for VB learning are presented earlier in either FA, for the factor loading $\mathbf{A}$, the hidden states $\mathbf{S}$, the ARD parameter α, or in VBGMM for the mixing proportion π and the hidden states (indicators) $\mathbf{Z}$.
Due to the similarity to the previous simpler model model FA and GMM, I will just write down the complete data likelihood $p(\mathbf{X}, \mathcal{H}|\mathcal{M})$ needed for evaluating the VB learning.

$$\begin{aligned} p(\mathbf{X},\mathcal{H}|\mathcal{M}_j) &= p(\mathbf{X},\mu,\mathbf{\Psi}_x,\mathbf{Z},\pi,\mathbf{A},\alpha,\mathbf{S}|\mathcal{M}_j)\\ &= [\prod_i^N p(\mathbf{x}_i|\mu,\mathbf{\Psi}_x,\mathbf{z}_i)p(\mathbf{z}_i|\pi)p(\mathbf{s}_i)]\; p(\pi|\mathbf{u})\; [\prod_m^M p(\mu_m)p(\mathbf{A}_m|\alpha_m)p(\alpha_m)]p(\mathbf{\Psi})\\ \Rightarrow \ln p(\mathbf{X},\mathcal{H}|\mathcal{M}_j) &= \sum_i^N\sum_m^M z_{i,m}[\ln p(\mathbf{x}_i|z_{i,m},\mu_m,\mathbf{\Psi}^{-1},\mathbf{s}_i) + \ln p(z_{i,m}|\pi_m) + \ln p(\mathbf{s}_i)]\\ &\quad + \sum_m \ln p(\pi_m|u_m) + \ln p(\mathbf{A}_m|\alpha_m)\\ &\quad + \sum_m^M[\ln p(\mu_m|\mathbf{m}_m,\mathbf{V}_m) + \ln p(\alpha|a,\mathbf{b})] + \sum_k^{d_X} p(\psi_k). && (5.60)\end{aligned}$$

All the above needed quantities such as priors, are given in sections for the simpler models. The priors should be plugged in, and the expectation is then taken wrt. to all the variables except the one of interest as given in equation (5.31). Since all the priors are conjugate priors, the variational posterior $\mathcal{Q}(\mathcal{H}_k)$ will have the same form as the prior $p(\mathcal{H}_k)$, thus when solving the expectation, quantities are gathered together so as the result will look like, the logarithm of the distribution of interest $\mathcal{Q}(\mathcal{H}_k)$, all the constants and terms (even variables) independent on the one of interest, i.e., $\mathcal{H}_{l \neq k}$ $\forall$ l are then regarded as the normalization constant.

CHAPTER 6

Applications

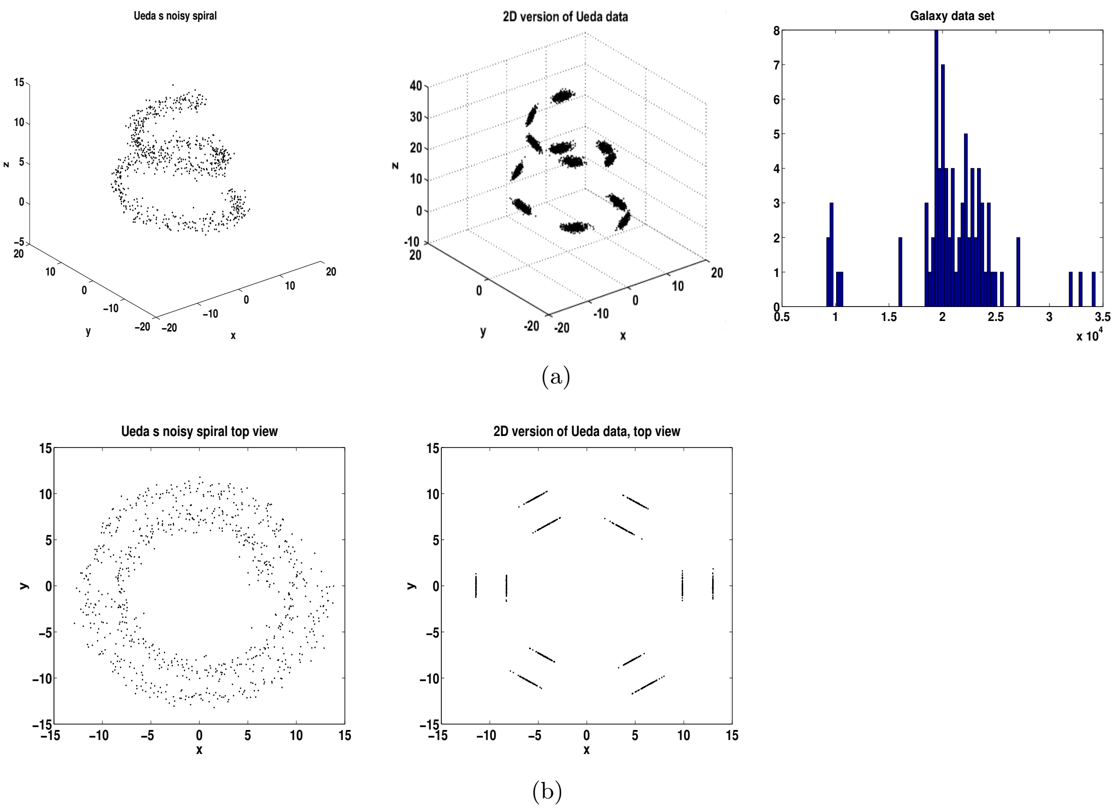


Figure 6.1: This figure shows the artificial (a) data, and real data (b) used to compare the models in the thesis. (a)(left) is a 3D view of Ueda's 1D noisy spiral described by equation (6.1). (a)(right) is a modified version of Ueda's data with intrinsic dimension of 2. (b) is the histogram of Galaxy data set

The aim of this chapter is to see how the models described in the course of thesis works, for both ML and VB learning. These models will be also compared to each other, and their performance will be tested on real and artificial data. Since ML suffers from overfitting as shown in example (5.2), a BIC penalized ML will be used. The data sets used in this chapter will be described next.

6.1 Artificial data

One of the used artificial data to compare mixture models is *Ueda*'s noisy shrinking spiral figure 6.1(left). This data set is used by many researchers for the purpose of testing mixture models e.g., [32, 16]. Ueda's spiral is generated as follows:

$$\begin{aligned} \mathbf{x}_i &= [(13-0.5t_i), \quad -(13-0.5t_i)\sin t_i, \quad t_i] + \varepsilon_i \\ \textit{where } t_i &\in [0, 4\pi], \qquad \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}([.5, .5, .5])), \end{aligned} \tag{6.1}$$

the parameter t_i determines the point along the spiral in one dimension. Ueda's data described above, can be seen in figure 6.1. This is a nice data set for testing mixture models, GMM, MFA and MPPCA, since the data has a 1D line folded as a spiral. Thus a good model will be the one that is able to find the hidden dimensionality. However this data does not contain a true number of clusters to make a decision about how good is the estimated model size inferred by a model of interest.

Figure 6.1(mid.) shows my version of this data, where the hidden dimensionality is increased to two, and where M distinct 2D mixture components are embedded in the shrinking spiral.

To generate such data I followed the same steps as in equation (6.1), where I first generate the mean of the mixtures as:

$$\begin{aligned} \mu_m &= [(13-0.5t_m), \quad -(13-0.5t_m)\sin t_m, \quad t_m] \\ \textit{where } \mathbf{t} &= \{t_m\}_{m=1}^{M} = \frac{[0:M-1]}{M-1}4\pi. \end{aligned} \tag{6.2}$$

Each of these mean μ_m will represent a centroid of a rotated 2D Gaussian components, the rotation is in 3D, i.e., around x, y and z axis, by the orthonormal rotation matrices $Q = A_x A_y A_z$, where:

$$A_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(A_x) & \sin(A_x) \\ 0 & -\sin(A_x) & \cos(A_x) \end{pmatrix} \tag{6.3}$$

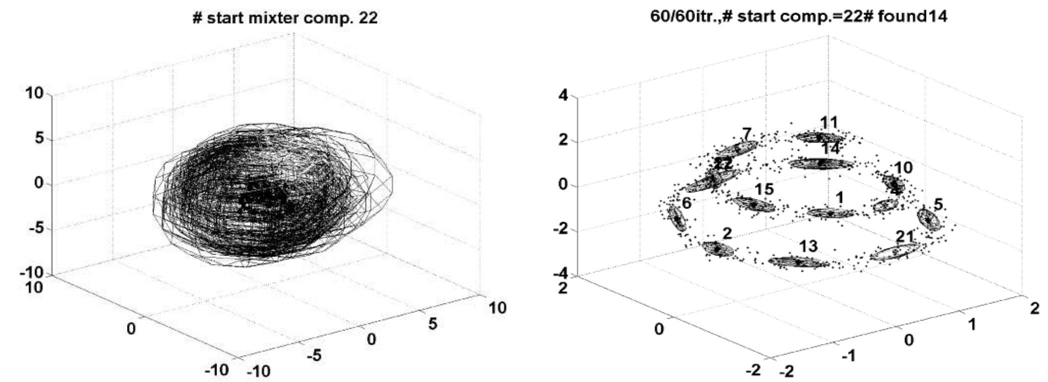
$$A_y = \begin{pmatrix} \cos(A_y) & 0 & \sin(A_y) \\ 0 & 1 & 0 \\ -\sin(A_y) & 0 & \cos(A_y) \end{pmatrix} \tag{6.4}$$

$$A_z = \begin{pmatrix} \cos(A_z) & \sin(A_z) & 0 \\ -\sin(A_z) & \cos(A_z) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{6.5}$$

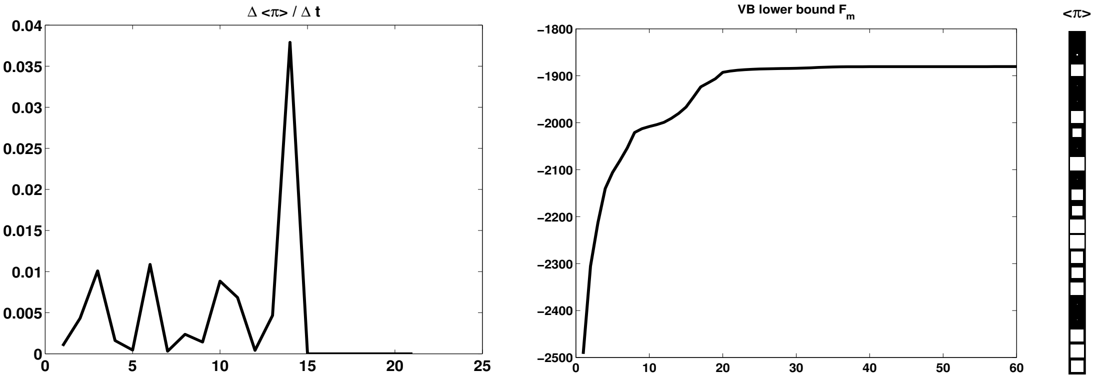
To perform the multiplication $Q\mathbf{X}_m$ of each single 2D Gaussian a row of for instance zeros should be appended to $\mathbf{X}$, i.e., $\mathbf{X}_m \leftarrow [\mathbf{X}_m^\top, \mathbf{0}^\top]^\top$, the resulting set is a "3D" data, and can then be multiplied by the rotation matrix.

6.2 Real Data

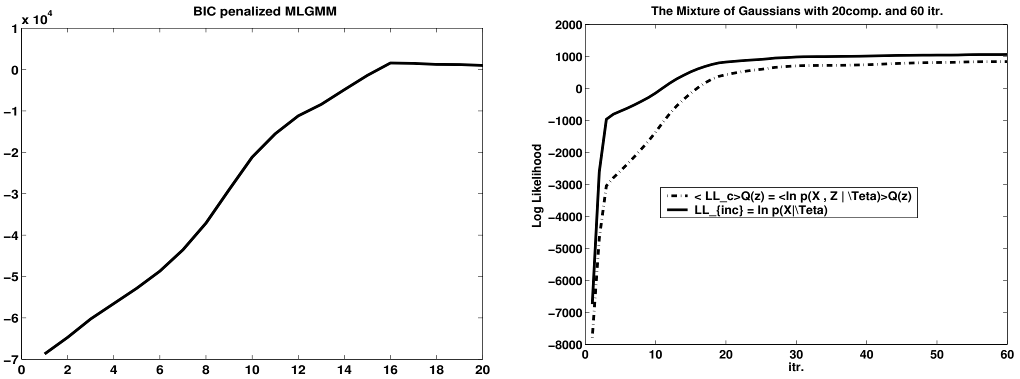
For real data I choose a simple univariate 'Galaxy data', which was first described by Roeder [2]. The data set consists of the velocity of 82 distant galaxies, diverging from our own galaxy. This data set was subsequently analyzed under different mixture models. Figure 6.4 shows the histogram of these data.



(a)



(b)



(c)

Figure 6.2: This figure shows how VBGMM determines the number of components. (a) (left) shows the initialization, where 22 components were used. (a)(right) VBGMM after convergence 14 components were left. (b) (right) shows hinton diagram of the resulting mixing proportions, where large squares correspond to large positive values, and there are 14 of them, corresponding to the number of inferred components. (b)(mid) shows the lower bound $\mathcal{F}_m$ during learning, the model converges already after the first 20 iterations. (b)(left) is just an easier way to visualize the inferred number of components, where the $< \pi >$ is sorted and differentiated ($-\frac{\Delta\langle\pi\rangle}{\Delta t}$). Note that 14 was found by many researchers [16, 32]. (c)(left) shows the expected complete data log likelihood (lower bound) vs. the incomplete data log likelihood for the case of 20 components. (c)(right) shows the BIC penalized MLGMM, runs for several components, and the highest score is assigned to a mixture of 16 components.

6.2.1 Determining the number of components using BIC and VB.

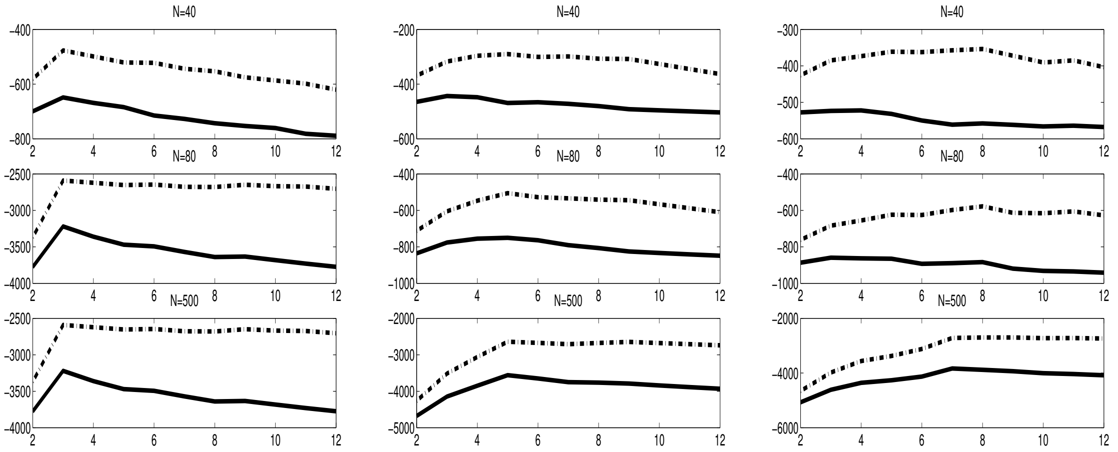
As a first test VBGMM and penalized MLGMM will be tested to find the number of mixtures in ueda's spiral, note that in this data there is no true number of mixtures. BIC criterion for GMM's can be seen in equations (5.57) and (5.56).
Figure 6.2 shows VBGMM applied to Ueda's spiral data, and the initial number of mixture is 22. After convergence the inferred number of components is $M = 14$, which is similar to, e.g., [16, 32]. A good thing about VB is that there is no need for discrete search of the model size. The unneeded components will simply die out as they don't get support from data. And as we will see shortly, this depends not only on data but also on our choice of the prior. BIC as seen in figure 6.2(c)(left) needs to "try" the appropriate model before deciding whether it should be chosen, by compering its performance with that of other models. Since the appropriate model is unknown, a large search in the model space should be made, which is computationally costly.

Since there is no true number of components, testing the model using new unseen data is almost a must. The Ueda spiral contains $N = 800$ data point. I used 70% of them as training data, and the rest as test data, the results are shown below, where GMM, MPCA and MFA are tested.

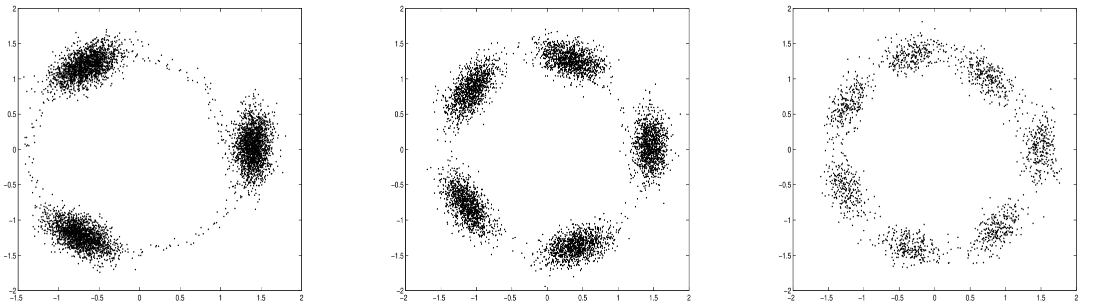
	GMM				
	spherical	diagonal	full	MFA	MPCA
Train	2.47	2.4	-0.99	-0.2	-056
Test	2.64	2.2	1.92	0.45	0.92

In the above table each data represent the average over 20 runs of ML learning. As it can be seen and expected, MFA and PCA outperform the non constrained or poorly constrained GMM. This expectation is based on the fact that the data has an intrinsic dimension, which is impossible for GMM to estimate without getting singularity problems, and then break down. MPCA or MFA has the ability to deal with those problems, due to their structure (see chapter 4).

In the next experiment VB and BIC are performed on 3 Gaussian mixture models with $M = \{3, 5, 7\}$ components, corresponding to (a), (b) and (c) in figure 6.3 respectively. Each of these mixtures (d), gave rise to 3 independently drawn data sets, with $N = \{40, 80, 500\}$ corresponding to the 1st. 2nd. and 3rd. rows in figure 6.3 respectively. In this experiment α was set such that, the data term is dominant (5.49), this is done by assigning a small value to α (.01) in this experiment. For the case of 3 mixtures (a), $N = 40$ was enough for both BIC and VB to infer the number of components. Whereas given $N = 40$ for $M = 5$ mixtures VB fails, and assigns a higher score ($\mathcal{F}_m$) to a model with 3 components, which is of lower complexity. BIC however has selected the true one $\tilde{M}_{\mathrm{BIC}} = 5$, this is not what one should expected, since according to equations (5.56) and (5.57), for a small data set BIC over-penalizes large models. Moreover VB was supposed to satisfy with small amount of data, which is enough to compute sufficient statistics of the model. Also in the case of 7 mixtures (c) for both $N = 40$ and 500, BIC was closer to the true value of $M = 7$, $\tilde{M}_{\mathrm{BIC}} = 8$, while $\tilde{M}_{\mathrm{VB}} = 4$ and $\tilde{M}_{\mathrm{VB}} = 3$ for $N = 40$ and $N = 80$ respectively. A similar result for the case of linear FA, was achieved by my colleague F.B.Nielsen [34] when he deduced that VB tends to underestimate the model choice. The same phenomena can be seen here.



(a) $M_{true} = 3$ (b) $M_{true} = 5$ (c) $M_{true} = 7$

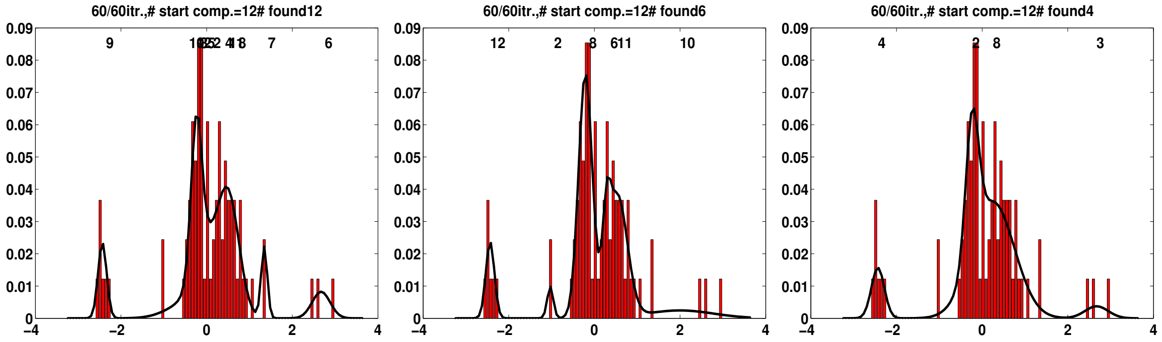


(d) Original mixtures

Figure 6.3: Data sets of different size $N = \{40, 80, 500\}$, drawn independently from each of the 3 Gaussian mixtures with number of mixtures $M = \{3, 5, 7\}$ (d). The 9 data sets are modelled by VBGMM and BIC penalized ML, each of the learning consist of 100 iterations. The resulting curves from BIC (dashed line) and VB-lower bound (solid line) are illustrated in (a), (b) and (c) for the nine data sets.

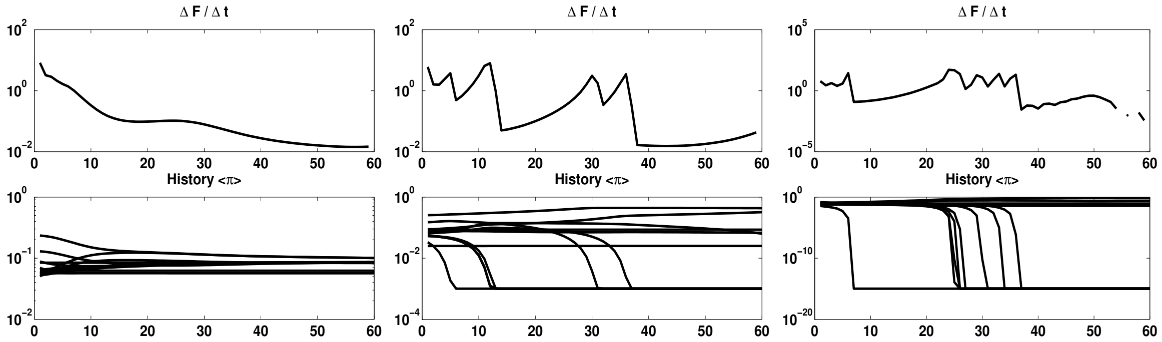
6.3 The effect of priors

Obviously the choice of prior has an effect on the computed posteriors and model selection. A close look at equations (5.39)-(5.49), one can easily see that updating is a sum of prior term and a data term. Allowing for weak priors is similar to allowing the data to dominate the updates. However for large priors, or equivalently small data set, the updates will be mainly based on the prior, which in this case will act as a regularizer. The effect of the prior $p(\pi|\alpha\ddot{\mathbf{u}})$ (see equation (5.12)) is investigated in a simple experiment of estimating the density distribution of the Galaxy data. In figure 6.4 (a), (b) and (c) 3 different values of α controlling the mixing proportions π are tested, $\alpha = \{100, 1, 10^{-2}\}$ corresponding to strong, noninformative and weak priors respectively and corresponding to left, middle and right figures respectively. For a strong prior all components will survive the learning, and are forced to represent the data equally, while for a weak prior only few will survive. This can be clearly seen in figure 6.4(c)(down), for weak prior ($\alpha = 10^{-2}$) (right), around 40 iterations 8 components die out very fast, which can be seen by the lines of $< \pi >$ falling down almost vertically, resulting in a density estimation with a mixture of 4 components. For $\alpha = 1$ (mid) the components die a bit

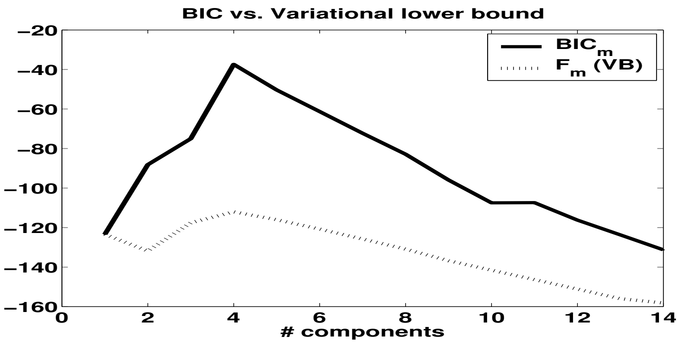


(a)

(b)



(c)



(d)

Figure 6.4: VBGMM and a BIC penalized ML applied to 'Galaxy' data. (a) shows the final result of density estimation using VBGMM with different $\alpha = \{100, 1, .01\}$, from left to right, respectively. (b) shows the hinton diagram of the final values of the mixing proportions at the end of the learning. (c)(up) shows the change in the lower bound $\Delta\mathcal{F} = \mathcal{F}_t - \mathcal{F}_{t-1}$. In (c)(down) one can see that for large α components do not 'die out' as fast as for small ones. A large factor ($\alpha = 100$) was used to compare how BIC will perform comparing to VB, (d) shows the scores for BIC and $\mathcal{F}$, both of them agreed for the same number of latent mixtures 4.

slower, ending with a mixture of 6 components. Finally for strong prior ($\alpha = 100$) all components survive, where the values of $< \pi >$ (left) remain almost the same as in the initialization and the density is then represented by a mixture of all the 12 components.

In figure 6.4 (d) the performance of BIC and VB to choose an appropriate model size (i.e., number of mixtures) to estimate the density of the Galaxy data is tested. For the comparison to make sense, the effect of π must be switched off, i.e., choose a strong prior ($\alpha = 100$ in this case), the best model is then the one which gives the highest score, the figure shows that both BIC and VB agreed for density estimation using a mixture of 4 components.

6.4 Image compression

In this section the performance of PPCA, EMFA and EMMFA, in image compression, will be tested. As the usual methods for image compression such as *Karhunen-Lòeve* transformation or PCA, an image is subdivided into nonoverlapping blocks of $K \times K$ pixels, usually $K = 8$ or $K = 16$ is used. Each block is transformed to a vector $\mathbf{x}$ $K^2 \times 1$. The whole image is then $\mathbf{X} = \{\mathbf{x}_i\}$, and the compression algorithm is then described as follows:

1 Choose a desired dimension q and number of mixture components M.

2 Estimate μ_m and $\mathbf{A}_m$, for all m, by fitting an MFA to $\mathbf{X}$.

3 For each $\mathbf{x} \in \mathbf{X}$

   (a) compute

$$\tilde{\mathbf{x}}_m = \mathbf{A}_m(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}_m^\top(\mathbf{x} - \mu_m) + \mu_m, \quad \forall\ m \tag{6.6}$$

   (b) the vector that minimizes $||\tilde{\mathbf{x}}_m - \mathbf{x}||^2$ is assigned to $\tilde{\mathbf{x}}_m^*$

4 Transform $\tilde{\mathbf{x}}_m^*$ back to a block image

for more detail see N. Ueda et.al [44]. Figure 6.5-(a)(right),-(b)(left) and (b)(right) shows an example of compressing an image using PPCA, EMFA and EMMFA respectively, where the original image is shown in (a)(left). As expected it can be seen that MFA performs best with an error of $e = 0.0015$, followed by FA with $e = 0.0029$ and at last comes PPCA with a slightly larger error $e = 0.003$. This is probably due to the fact that PPCA confuses noise with information in the K^2 dimensional $\mathbf{X}$, as shown in example (4.2), where I showed the particular problem of variance confusion PPCA suffers from. The performance can also be visually judged by looking at the quality of the compression, where $K = 4$, and $q = 3$, i.e., a compression ratio of $q/K^2 = 0.1875$.

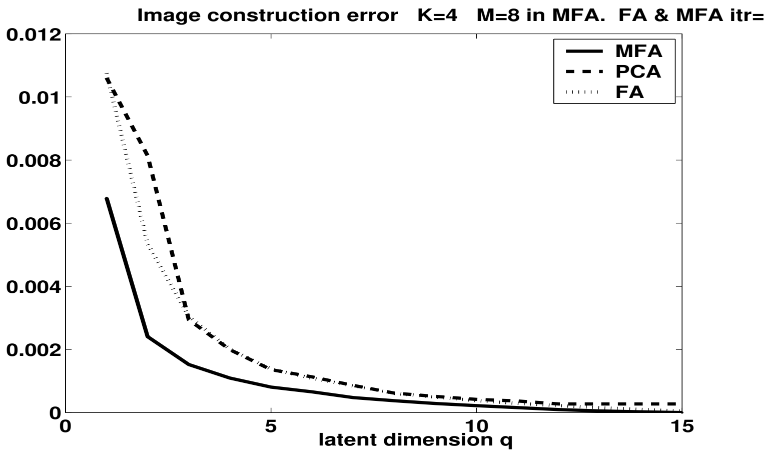
Figure 6.5(c) shows how the error (e=$||\tilde{\mathbf{x}}_m - \mathbf{x}||^2$) decreases to zeros as the latent dimensionality gets closer to the true dimension, for the three technics. The experiment confirms that MFA performs best followed by FA and PPCA.

6.5 Inferring Latent Dimensionality

As a last example, I will show here, how MFA finds the mixtures and the latent dimensionality. This is shown in figure 6.6, where it can be seen that the model finds actually the 1D hidden space, and manage to separate the noise from the information, which here the shrinking spiral. An other example is shown in (b), consisting of a rotated uniformly distributed 2D source. The reconstructed signal is a rotated version of the true source, and this is mainly because FA expects to model a Gaussian and therefore finds the direction of high variance and its quadratic to be its diagonals. This can be seen in the rotated way, the Gaussian estimate lies on the hidden source in (b)(left). This represents in fact a major drawback of FA models and their extension. And Attias however combined FA with GMM to get a better model [19].

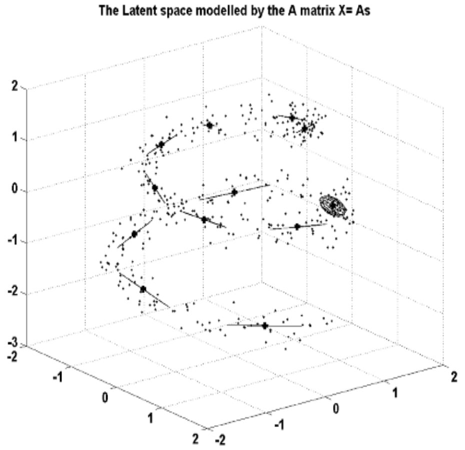
(a)

(b)

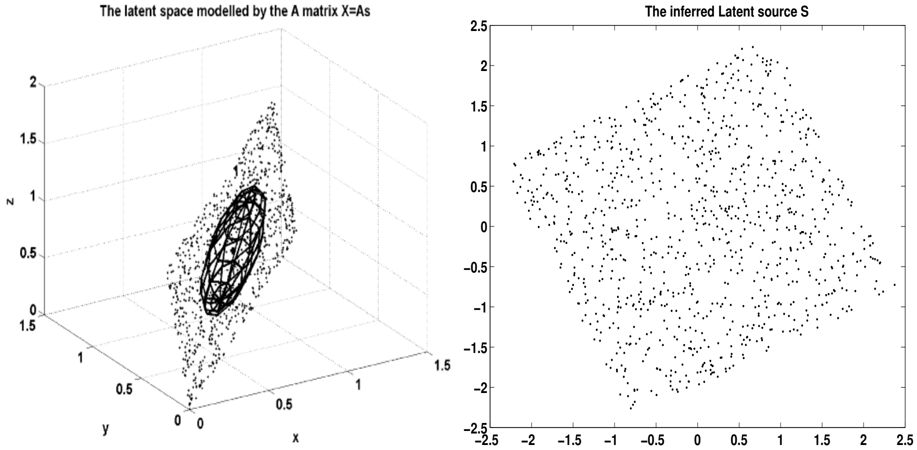


(c)

Figure 6.5: (a)(left) is the original image of Lenna (commonly used in image processing), the compression with a ratio of $q/K^2 = 0.1875$. (a)(right) is a reconstructed image after compression using PPCA where a texture appears on the image the square error .003. (b)(left) is the MLFA .002. (b) (right) EMMFA .001. It is visually clear that a MFA model performs better, followed by FA and at last come PCA. This is supported by the error function plotted as function of the latent dimensionality q (c).



(a)



(b)

Figure 6.6: Figure(a) shows how VBMFA is able to find the mixtures in a manifold, and also their latent dimension. Figure (b)(left) shows that VBFA is able to find the hidden dimensionality. Note, the signal it self is non-Gaussian, therefore the reconstructed latent source in (b)(right) is rotated according to correlation and not independency, higher moments than are therefore needed in order to solve the problem. This figure shows in fact a drawback of FA in general, because these models assumes the sources to be Gaussian distributed, which is not the case in most practical cases.

CHAPTER 7

Conclusion

The principle of building complex models based on simpler ones, has been proved to be a fruitful idea throughout this thesis. An important advantage is that it leads to learning algorithms which are analytically and computationally tractable. In fact at the heart of the learning algorithm of a complex model, lies the algorithm of a simpler one. The EM algorithm for maximum likelihood estimation embodies this idea in a probabilistic way. The idea of building more complex models by an extension to simpler or basic models, gave rise to a unified framework of many well known techniques in machine learning. Such framework is the generative model or latent variable model discussed in the thesis.

A detailed derivation of latent variable models, where techniques such as FA, PPCA are their special cases. Two learning methods for the linear latent variable models were given in chapter 4. The first one was ML learning section 3.1, which was made tractable by the EM algorithm. The other discussed learning method was Bayesian learning chapter 3.2. Because of its intractable derivation, approximation methods were discussed, with the focus on variational Bayes technique section 3.3, where EM can be regarded as a special case.

A further extension to basic models was achieved by introducing mixture of latent variable models chapter 5, which includes GMM, MFA and MPCA. A detailed derivation of the simplest case GMM was made in section 5.1. Based on GMM and linear latent variable models, it is easier to derive the learning algorithm for the other two models, MFA and MPCA, which were briefly mentioned in section 5.2.
Within the pervious mentioned sections and chapters small examples were derived, to compare models, or to justify some made assumptions. One of the assumptions was to use separate priors for means and covariance in a linear model, which from example (4.1) proved to be worse than the joint prior, but still gave nice result due to the form of minimized KL divergence. Another example was the failure of PPCA to separate information from noises.

In chapter 6 VB vs. BIC was discussed in form of experiments, which leads to unexpected results, such as BIC can outperform VB in small data set.
However all the discussed models are Gaussian, which are not always the case in real data. Non-Gaussian extension to the discussed models could be more general. Another way could be to use nonlinear transformation of the linear models, while keeping the nice derivation achieved by the simplicity of Gaussian model. Furthermore all the discussed model are static models where the time factor is neglected, which certainly limits the use of such models.

APPEDNDIX A
Important derivations

A.1 Lower bound derivation

A more detailed derivation of the lower bound equation (3.37).

$$\begin{aligned}
\mathcal{L}_{inc}(\boldsymbol{\Theta}) &= \sum_i \ln p(\mathbf{x}_i|\boldsymbol{\Theta}) & \text{(A.1)}\\
&= \sum_i \ln p(\mathbf{x}_i|\boldsymbol{\Theta}) \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)\ d\mathbf{s}_i & \text{(A.2)}\\
&= \sum_i \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln p(\mathbf{x}_i|\boldsymbol{\Theta})\ d\mathbf{s}_i & \text{(A.3)}\\
&= \sum_i \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln[\frac{p(\mathbf{x}_i,\mathbf{s}_i|\boldsymbol{\Theta})}{p(\mathbf{s}_i|\mathbf{x}_i,\boldsymbol{\Theta})}\ \frac{\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)}{\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)}]\ d\mathbf{s}_i & \text{(A.4)}\\
&= \sum_i \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{p(\mathbf{x}_i,\mathbf{s}_i|\boldsymbol{\Theta})}{\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)}\ d\mathbf{s}_i + \sum_i \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)}{p(\mathbf{s}_i|\mathbf{x}_i,\boldsymbol{\Theta})}\ d\mathbf{s}_i & \text{(A.5)}\\
&= \sum_i \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{p(\mathbf{x}_i,\mathbf{s}_i|\boldsymbol{\Theta})}{\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)}\ d\mathbf{s}_i + \sum_i \mathrm{KL}(\mathcal{Q}_{\mathbf{s}_i}||p(\mathbf{s}_i|\mathbf{x}_i,\boldsymbol{\Theta})) & \text{(A.6)}\\
&\geq \sum_i \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{p(\mathbf{x}_i,\mathbf{s}_i|\boldsymbol{\Theta})}{\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)}\ d\mathbf{s}_i & \text{(A.7)}\\
&= \mathcal{F}(\{\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)\}_{i=1}^N, \boldsymbol{\Theta}) & \text{(A.8)}
\end{aligned}$$

Researchers adopt different ways to get the same above result, depending on where they start the derivation, most of them use directly the *jensen's inequality* which is based on the concavity property of the logarithmic function, see e.g.[17, 46, 40].
An explanation of the different steps of the above derived relation can be explained as follows:

- In equation (A.3) the integral over the hidden variables does not affect the likelihood since $\int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)\ d\mathbf{s}_i = 1$.
- Since $\ln p(\mathbf{x}_i|\boldsymbol{\Theta})$ is independent on the hidden variable $\mathbf{s}$, it can be, with no harm, be placed inside the integral in equation (A.4).
- In equation (A.5) the Bayes' rule has been used.
- The quantity $\mathrm{KL}(\mathcal{Q}_{\mathbf{s}_i}||p(\mathbf{s}_i|\mathbf{x}_i,\boldsymbol{\Theta})) = \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)}{p(\mathbf{s}_i|\mathbf{x}_i,\boldsymbol{\Theta})}\ d\mathbf{s}_i$ in equation (A.7) is the kullback-Liebler divergence presented in section 3.3.1.
- The inequality in equation (A.8) is usually referred to as *Jenssen's inequality* that is based on the concavity property of the logarithmic function. It can be explained here by the fact that $\mathrm{KL} \geq 0$ (see section 3.3.1).

- $\mathcal{F}(\mathcal{Q}_{\mathbf{s}_i}, \mathbf{\Theta}) = \int \mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{p(\mathbf{x}_i, \mathbf{s}_i|\mathbf{\Theta})}{\mathcal{Q}_{\mathbf{s}_i}(\mathbf{s}_i)}\, \mathrm{d}\mathbf{s}_i$ in the last equation (A.8) is the *negative* of a quantity known in statistical physics as the *free energy*, and represent the *lower bound.*

APPEDNDIX B

Distributions, sufficient statistics and KL

In the following table, gives the formula of the used distributions, their sufficient statistics and KL divergences.

B.0.1 Multivariate Gaussian

Notation & Parameters

$\theta \sim \mathcal{N}(\mu, \mathbf{\Psi}^{-1})$ μ mean vector; $\mathbf{\Psi}$ precision matrix

density function

$p(\theta|\mu, \mathbf{\Psi}) = (2\pi)^{-d/2}|\mathbf{\Psi}|^{1/2} \exp(-1/2\text{Tr}[\mathbf{\Psi}(\theta-\mu)(\theta-\mu)^\top])$

sufficient statistics & KL

$\langle\theta\rangle = \mu$, $\langle\theta\theta^\top\rangle = \mathbf{\Psi}^{-1}$ $\text{KL}(\tilde{\mu}, \tilde{\mathbf{\Psi}}||\mu, \mathbf{\Psi}^{-1}) = \frac{-1}{2}(\ln|\mathbf{\Psi}^{-1}\tilde{\mathbf{\Psi}}| + \text{Tr}[I - [\tilde{\mathbf{\Psi}}^{-1} + (\tilde{\mu}-\mu)(\tilde{\mu}-\mu)^\top]\mathbf{\Psi}] \ln e)$

B.0.2 Gamma

Notation & Parameters

$\tau \sim \mathcal{G}(\alpha, \beta)$ $\alpha > 0$ shape; $\beta > 0$ inv. scale

density function

$p(\tau|\alpha, \beta) = \frac{\beta}{\Gamma(\alpha)}\tau^{\alpha-1}\exp^{-\beta\tau}$

sufficient statistics & KL

$\langle\tau^n\rangle = \frac{\Gamma(\alpha+n)}{\beta^n\Gamma(\alpha)}$ $\langle\tau\rangle = \frac{\alpha}{\beta}$
$\text{KL}(\tilde{\alpha}, \tilde{\beta}||\alpha, \beta) = \tilde{\alpha}\ln\tilde{\beta} - \alpha\ln\beta - \ln\frac{\Gamma(\tilde{\alpha})}{\Gamma(\alpha)}$
$+(\tilde{\alpha}-\alpha)(\psi(\tilde{\beta}) - \ln\tilde{\beta}) - \tilde{\alpha}(1 - \frac{\beta}{\tilde{\beta}})$

B.0.3 Dirichlet

$\pi \sim \mathcal{D}(\alpha)$, prior sample sizes $\alpha = \{\alpha_k\}_1^K$; $\alpha_k > 0$; $\alpha_0 = \sum_1^K \alpha_k$

density function

$p(\pi|\alpha) = \frac{\Gamma(\alpha_0)}{\prod_k \Gamma(\alpha_k)} \prod_k \pi_k^{\alpha_k - 1}$

sufficient statistics & KL

$\langle\pi\rangle = \alpha/\alpha_0$
$\mathrm{KL}(\tilde{\alpha}||\alpha) = \ln\frac{\Gamma(\tilde{\alpha_0})}{\Gamma(\alpha_0)} - \sum_k[\ln\frac{\Gamma(\tilde{\alpha}_k)}{\Gamma(\alpha_k)} - (\tilde{\alpha}_j - \alpha_k)(\psi(\tilde{\alpha}_k) - \psi(\tilde{\alpha}_0))]$

Bibliography

[1] H. Attias (2000). **A variational Bayesian framework for graphical models**. In *Adv. Neur. Info. Proc. Sys. 12 (Ed. by Leen, T. et al). MIT Press, Cambridge, MA.*, pages 209–215.

[2] Peter J. Green and Sylvia Richardson. **On Bayesian analysis of mixtures with an unknown number of components**. *the Royal Statistical Society, B, 59*, pages 731–792, 1997.

[3] Frey B.J. **Turbo factor analysis**. *In Adv. Neural Information Processing Systems (submitted)*, December 1999.

[4] David M. Blei and Michael I. Jordan. **Modeling annotated data**. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134. ACM Press, 2003.

[5] Bishop C. M. ***Neural Networks for Pattern Recognition***. Clarendon Press, Oxford, 1995.

[6] Bishop C. M. **Latent variable models.** *In M. I. Jordan (Ed.), Learning in Graphical Models, MIT Press*, pages 371–403, 1999.

[7] Bishop C. M. and Corduneanu A. **Variational Bayesian model selection for mixture distributions**. *In T. Richardson and T. Jaakkola (Eds.), Proceedings Eighth International Conference on Artificial Intelligence and Statistics, pp. Morgan Kaufmann*, pages 27–34, 2001.

[8] Bishop C. M. and Tipping. **Probabilistic principal component analysis**. *Journal of the Royal Statistical Society, Series B, 16(3)*, pages 611–622, 1999.

[9] Heckerman David. **A tutorial on learning with bayesian networks**. Technical report, MSR-TR-95-06, Microsoft Research, Redmond, Washington, 1995. Revised June 96.

[10] MacKay David J.C. **Bayesian Interpolation**. *Neural Computation*, 4(3):415–447, 1992.

[11] MacKay David J.C. **Ensemble Learning and Evidence Maximization**. submitted to NIPS*95, 1995.

[12] MacKay David J.C. **Probable Networks and Plausible Predictions — A Review of Practical Bayesian Methods for Supervised Neural Networks**. *Network: Computation in Neural Systems*, 6:469–505, 1995.

[13] MacKay David J.C. **Hyperparameters: Optimize, or Integrate out?** In G. Heidbreder, editor, *Maximum Entropy and Bayesian Methods, Santa Barbara 1993*, pages 43–60, Dordrecht, 1996. Kluwer.

[14] MacKay David J.C. ***Information theory, inference and learning algorithms***. Cambridge Press, 2003.

[15] Draper Denise. **Clustering Without (Thinking About) Triangulation**. In *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pages 125–133, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

[16] Z. Ghahramani and M. Beal. **Variational Inference for Bayesian Mixture of Factor Analysers**. *In Advances in Neural Information Processing Systems 12*, pages 449–455, 2000. eds. S. A. Solla, T.K. Leen, K. Müller, MIT Press.

[17] Z. Ghahramani and M. Beal. **Graphical Models and Variational Methods**. In *M. Opper and D. Saad (eds.), Advanced Mean Field Methods — Theory and Practice, MIT Press*, 2001.

[18] Zoubin Ghahramani and Geoffrey E. Hinton. **The EM Algorithm for Mixtures of Factor Analyzers**. Technical Report CRG-TR-96-1, 21 1996.

[19] Attias Hagai. **Independent Factor Analysis**. *Neural Computation*, 11(4):803–851, 1999.

[20] Attias Hagai. **Inferring Parameters and Structure of Latent Variable Models by Variational Bayes**. *In Proc. 15th Conf. on Uncertainty in Articial Intelligence*, pages 21–30, 1999.

[21] L. Hansen, S. Sigurdsson, T. Kolenda, F. Nielsen, U. ﬃ, and J. Larsen. **Modeling text with generalizable gaussian mixtures**. *In Proc. of IEEE ICASSP'2000*, 6:3494–3497, 2000.

[22] P. A. Højen-Sørensen, O. Winther, and L. K. Hansen. **Mean Field Approaches to Independent Component Analysis**. *Neural Computation*, 14(4):889–918, April 2002.

[23] Bilmes J. **A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models**. Technical report, University of Berkeley, ICSI-TR-97-021, 1998.

[24] Jackson J. Edward. ***A user's guide to Principal Components***. II. Wiley interscience, 1991.

[25] Tommi S. Jaakkola and Michael I. Jordan. **Bayesian parameter estimation via variational methods**. *Statistics and Computing*, 10(1):25–37, 2000.

[26] Michael I. Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence K. Saul. **An Introduction to Variational Methods for Graphical Models**. *Machine Learning*, 37(2):183–233, 1999.

[27] Conradsen knut. ***En introduction til statistik***, volume 2. IMM, Kgs Lyngby, 6 edition, 2002.

[28] Buntine Wray L. **Operations for Learning with Graphical Models**. *Journal of Artificial Intelligence Research*, 2:159–225, 1994.

[29] Harri Lappalainen and J. Miskin. **Ensemble Learning**. In *M. Girolami (Ed.), Advances in Independent Component Analysis, Springer, Berlin, 2000 (in press).*

[30] Wentian Li and Dale R Nyholt. **Marker Selection by AIC and BIC**. *Laboratory of Statistical Genetics, The Rockefeller University, New York, NY.*

[31] Geoffrey McLachlan. **Mixtures of Factor Analyzers**. In *Proc. 17th International Conf. on Machine Learning*, pages 599–606. Morgan Kaufmann, San Francisco, CA, 2000.

[32] Beal M.J. ***Variational Algorithms for Approximate Bayesian Inference***. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, May 2003.

[33] Perry Moerland. **Mixtures of latent variable models for density estimation and classification**. IDIAP-RR 25, IDIAP, 2000. Submitted for publication.

[34] F. B. Nielsen. **Variational Approach to Factor Analysis and Related Models**. Master's thesis, IMM, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, 2004.

[35] Moerland Perry. ***Mixture Models for Unsupervised and Supervised Learning***. PhD thesis, EPFL, Martigny, Swiss, June 2000.

[36] Neal R. M. **Bayesian mixture modeling by Monte Carlo simulation**. Technical Report CRG-TR-91-2, Department of Computer Science, University of Toronto, 1991.

[37] Sam Roweis and Zoubin Ghahramani. **A Unifying Review of Linear Gaussian Models**. Technical report, 6 King's College Road, Toronto M5S 3H5, Canada, 1997.

[38] Sam Roweis and Ruslan Salakhutdinov. **Relationship between gradient and EM steps in Latent Variable Models**. Unpublished Report, Sep 2002.

[39] Mardia K. V. Kent J. T. and Bibby J. M. ***Multivariate Analysis***. IV. Academic press, 1995 edition, 1979.

[40] Minka T. P. **Expectation-Maximization as lower bound maximization**. Technical report, MIT, November 4 1998.

[41] Minka T. P. **Bayesian inference, entropy and the multinomial distribution**. Technical report, Media Lab, 2000.

[42] Minka T. P. **Variational Bayes for mixture models: Reversing EM**. Technical report, MIT, 2000.

[43] Michael E. Tipping and Christopher M. Bishop. **Mixtures of Probabilistic Principal Component Analysers**. *Neural Computation*, 11(2):443–482, 1999.

[44] Naonori Ueda, Ryohei Nakano, Zoubin Ghahramani, and Geoffrey E. Hinton. **SMEM Algorithm for Mixture Models**. *Neural Computation*, 12(9):2109–2128, 2000.

[45] H. Valpola and Petteri Pajunen. **Fast Algorithms For Bayesian Independent Component Analysis**. *In Proceedings of ICA2000*, pages 233–237, 2000.

[46] John Winn. ***Variational Message Passing and its Applications***. Ph.d, Department of Physics, University of Cambridge, 2003.

[47] O. Winther. **Noter til Variational Technique**. IMM, DTU, 2001.

[48] Carl Edward Rasmussen Zoubin Gahramani. **Occam's Razor**. Technical report, Advances in Neural Information Processing Systems 13, 294-300. (Eds.) Todd Leen, Thomas G. Dietterich and Volker Tresp, MIT Press, 2001.