

---

**A feature selected clustered connectivity for DCMs  
in a unified framework**

---

by

Gustaf GILLANDER

A thesis submitted in partial fulfilment of the requirements  
for the degree of  
Master of Science in Engineering Physics  
FACULTY OF ENGINEERING, LUND UNIVERSITY

November 2014

### **Abstract**

In recent years dynamic causal models (DCMs) have been developed to analyse the connectivity between brain regions and their response to stimuli. Discriminative methods have been applied to the estimates of the model parameters. In general, the combination of DCMs and the discriminative methods have been applied step-by-step. Recently an unified framework has been developed for generative embedding in the context of DCMs for multi-subject analysis. The unified framework combines the DCM with the generative method. Here a generative method in the form of a Gaussian mixture model with feature selection is proposed. The unified framework is extended with the presented feature selection to improve clustering performance and interpretability. In the context of Markov chain Monte Carlo the problem of label switching arises, due to non-identifiability of the mixture components. The problem of label switching is reviewed and solutions presented. Experiments shows that the inclusion of the proposed feature selection improves the clustering performance. The interpretation of cluster differences is aided by the measure of the features relative importance in discriminating the clusters.

**KEY WORDS:** Dynamic causal models; Feature selection; Gaussian mixture models; Generative embedding; Label switching; Markov chain Monte Carlo.

# Acknowledgements

This master's thesis has been carried out at the Translational Neuromodeling Unit, University of Zurich & ETH Zurich under the supervision of Dr. Sudhir Shankar Raman and Prof. Dr. Klaas Enno Stephan. This thesis is part of my MSc in Engineering Physics from the Faculty of Engineering, Lund University.

Foremost, I would like to thank my supervisor Dr. Sudhir Shankar Raman who has been patient and allowed me a great deal of freedom. I am grateful for his knowledgeable advice and guidance throughout the course of this thesis.

Further, I am grateful to Prof. Dr. Klaas Enno Stephan and his team at TNU, University of Zurich & ETH Zurich for their hospitality. I am glad that I got the possibility to work with them and learn more about their research. They have all inspired me with their scientific curiosity and knowledge.

I want to thank my family for always supporting and encouraging me. They help me go above and beyond and never stop dreaming. Without them I would not be where I am today.

Thank you to grandma for my persistence, it has come of great use.

Finally, I want to thank Laura for just being herself. I am ever so grateful for her constant support and great patience.

---

The Translational Neuromodeling Unit (TNU) is a division of the Institute of Biomedical Engineering at the University of Zurich and the Swiss Federal Institute of Technology (ETH Zurich). Their research is focused on mathematical models that infer subject-specific mechanisms of brain disease from non-invasive measures of behaviour and neuronal activity.



University of  
Zurich UZH

**ETH**

Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich



Translational Neuromodeling Unit

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Motivation . . . . .	5
1.2	Objective . . . . .	5
1.3	Outline . . . . .	6
<b>2</b>	<b>Bayesian Inference</b>	<b>7</b>
2.1	Introduction . . . . .	7
2.2	Bayesian statistics . . . . .	7
2.3	Methods of inference . . . . .	10
2.3.1	Laplace approximation . . . . .	10
2.3.2	Variational Bayes . . . . .	12
2.3.3	Monte Carlo . . . . .	12
2.4	Markov chain Monte Carlo . . . . .	14
2.4.1	Markov chains . . . . .	14
2.4.2	Metropolis-Hastings . . . . .	16
2.4.3	Gibbs sampling . . . . .	18
2.4.4	Inversion sampling . . . . .	18
2.4.5	Burn-in . . . . .	19
<b>3</b>	<b>Mixture Models</b>	<b>20</b>
3.1	Introduction . . . . .	20
3.2	Finite mixture of Gaussians . . . . .	21
3.2.1	Model . . . . .	21
3.2.2	Posterior conditional distributions . . . . .	22
3.2.3	Sampling . . . . .	24
3.2.4	Choosing $K$ . . . . .	25
3.3	Infinite mixture of Gaussians . . . . .	25
3.3.1	Limit of $K$ . . . . .	26
3.3.2	Sampling . . . . .	27
<b>4</b>	<b>Label Switching</b>	<b>29</b>
4.1	Introduction . . . . .	29
4.1.1	Exchangeability of random variables . . . . .	29

4.1.2	Inference for mixture models . . . . .	30
4.1.3	Efficient sampling . . . . .	31
4.2	Relabelling . . . . .	32
4.2.1	Artificial identifiability constraints . . . . .	32
4.2.2	Loss functions . . . . .	33
<b>5</b>	<b>Feature Selection</b>	<b>36</b>
5.1	Introduction . . . . .	36
5.1.1	Supervised & unsupervised learning . . . . .	36
5.1.2	Approaches of feature selection . . . . .	38
5.2	Model . . . . .	38
5.2.1	Feature distribution . . . . .	39
5.2.2	Global distribution . . . . .	41
5.2.3	Cluster assignment . . . . .	41
5.3	Sampling . . . . .	42
5.3.1	Feature sampling . . . . .	42
5.3.2	Mixture & global distribution sampling . . . . .	43
5.3.3	Assignment sampling . . . . .	44
<b>6</b>	<b>Dynamic Causal Models</b>	<b>45</b>
6.1	Introduction . . . . .	45
6.2	Dynamic causal models . . . . .	46
6.2.1	Neural state equations . . . . .	47
6.2.2	Haemodynamic model . . . . .	47
6.2.3	BOLD signal model . . . . .	48
6.2.4	Bayesian framework and priors . . . . .	49
6.3	Generative embedding . . . . .	49
6.3.1	Generative model . . . . .	49
6.3.2	Generative method . . . . .	50
6.3.3	Unified framework . . . . .	50
<b>7</b>	<b>Results</b>	<b>52</b>
7.1	Evaluation measures . . . . .	52
7.1.1	Normalised mutual information . . . . .	52
7.1.2	Balanced purity . . . . .	53
7.2	Synthetic data set . . . . .	54
7.3	Iris flower data set . . . . .	57
7.4	Label switching . . . . .	60
7.5	Dynamic causal models . . . . .	67
<b>8</b>	<b>Conclusions</b>	<b>69</b>
8.1	Further research . . . . .	69
<b>A</b>	<b>Data Sets</b>	<b>75</b>

# Nomenclature

$\bar{y}$	mean of $y$
$\delta$	Kronecker delta
$\mathbb{1}_A$	indicator function of the set $A$
$\mathbb{E}[X]$	expected value of $X$
$A^c$	complement of the set $A$
$F(x)$	distribution function of $X$
$L(a; \theta)$	loss function
$P(A)$	probability of $A$
$p(x)$	probability density function of $X$
$X \sim F$	$X$ is distributed as $F$
$\Gamma(p, a)$	gamma distribution
$\mathcal{L}(x)$	Likelihood of $x$
$\mathcal{N}(\mu, \sigma^2)$	normal distribution
$\mathcal{U}(a, b)$	uniform distribution between $a$ and $b$
$\mathcal{W}_D(\mathbf{W}, v)$	Wishart distribution of dimension $D$
$t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	multivariate t-distribution of dimension $v$
$\text{Be}(p)$	Bernoulli distribution
$\text{Dir}(\alpha, \dots)$	Dirichlet distribution
BOLD	blood-oxygen-level dependent
BP	balanced purity
DCM	dynamic causal model
dHB	deoxyhaemoglobin
ECR	equivalence classes representatives
EEG	electroencephalography
FGMM	finite Gaussian mixture model
FGMMFS	finite Gaussian mixture model with feature selection
fMRI	functional magnetic resonance imaging
gcd	greatest common divisor
IC	identifiability constraint
IGMM	infinite Gaussian mixture model
IGMMFS	infinite Gaussian mixture model with feature selection

MCMC	Markov chain Monte Carlo
MH	Metropolis-Hastings
VB	variational Bayes

# Chapter 1

## Introduction

Exploring data sets can be the first step to new discoveries. Finding new patterns and causes, leading to scientific breakthroughs. Investigating these patterns is of relevance in fields ranging from socioeconomics to image analysis and neuroscience. With these patterns could knowledge about neurological differences between healthy and diagnosed subjects be obtained. Leading to an improved understanding of the underlying causes of psychiatric disorders and how subjects can be diagnosed.

### 1.1 Motivation

Gaining knowledge and understanding of the system of the brain is important for the interpretation of responses with respect to stimulus. A better understanding of the system and differences between subjects can lead to an improved knowledge of psychiatric disorders. Dynamic causal models were presented in [Friston et al., 2003] as a way of modelling the dynamics of the brain on a subject by subject basis. The dynamic causal model presents a structure of how brain regions are interconnected and respond to input. By applying generative or discriminative methods to the estimated parameters from the dynamic causal model, dissimilarities and structures can be explored. This, so called generative embedding [Brodersen et al., 2011], provides the foundation for exploring the differences, in the functioning of the brain, between groups. These groups could for example be diagnosed subjects and healthy subjects, leading to a better understanding of the underlying causes of psychiatric disorders.

### 1.2 Objective

The objective is to extend a previous method for multi-subject analysis in the context of dynamic causal models and generative embedding developed at the Translational Neuromodeling Unit, University of Zurich & ETH Zurich [Raman and Stephan].



The extension of the method will be a feature selected clustering. Based on the assumption of a Gaussian mixture model, a model for the feature selected clustering will be constructed. The method is extended with feature selection in hope of improved interpretability and clustering performance. The objective for the feature selection can be summarised as

*An embedded approach which includes all informative features with respect to discriminating between clusters as well as providing a measure for their relative importance. The method should work in a Markov chain Monte Carlo framework with a finite or infinite mixture model.*

### 1.3 Outline

**Chapter 2** The fundamentals upon which the following chapters are based is presented. Some basic statistical concepts are explained in a Bayesian context. Methods for inference are presented, with a more thorough description of Markov chain Monte Carlo.

**Chapter 3** The Gaussian mixture model is introduced as a finite model. The limitations of a finite model is discussed and the steps to extend it to an infinite model is described.

**Chapter 4** The application of Markov chain Monte Carlo on a mixture model can result in label switching. The phenomena of label switching and the complications that arises are described. The solution to the label switching problem is to apply relabelling. Here, three relabelling algorithms, using two approaches, are reviewed.

**Chapter 5** The concept of feature selection is introduced. A method for feature selection is presented and used to extend the Gaussian mixture models from chapter 3.

**Chapter 6** Modelling of brain region interactions is introduced as dynamic causal models. The dynamic causal model and how it is used in generative embedding is described. A unified model for generative embedding is extended with the feature selection from chapter 5.

**Chapter 7** The performed experiments and the data sets used are presented. The obtained result is analysed and reviewed.

**Chapter 8** The conclusions of the work is presented with a brief comment on further research and extensions.

## Chapter 2

# Bayesian Inference

### 2.1 Introduction

Inference is the process of drawing conclusions based on reasoning and the evidence at hand. From a mathematical point of view, statistical inference is the process of drawing conclusions based on some observations under uncertainty, i.e. under random variation. For instance, could a probability distribution or a confidence interval for a variable be obtained. The work is based on Bayesian inference and in this chapter the basic idea of the Bayesian approach as well as methods for inference will be explained.

### 2.2 Bayesian statistics

In statistical inference there are two main approaches, (i) frequentist inference and (ii) Bayesian inference. Frequentists are only interested in the observations and do not incorporate any prior information or beliefs into the analysis. The variable of interest is seen as having a fixed value. For example, could a confidence interval for the variable be obtained. In Bayesian statistics the prior information is seen as something beneficial and is included in the analysis. In the Bayesian context the parameter of interest can be described with a probability distribution, which also includes the prior beliefs.

An example of the difference between the two methods can be seen in figure 2.1. The example shows the result of clustering observations from three different distributions. As seen in figure 2.1 (a) the assignments are hard. The observations only belong to one of the distributions, corresponding to the frequentist approach. Figure 2.1 (b) shows the result from a Bayesian approach. Instead of the observations being assigned to one of the three distributions, they have a probability of being assigned to each of the three distributions. In some areas there are a mix of the three colours. The colour of each observation is defined by its probability of being assigned to each of the three different distributions.

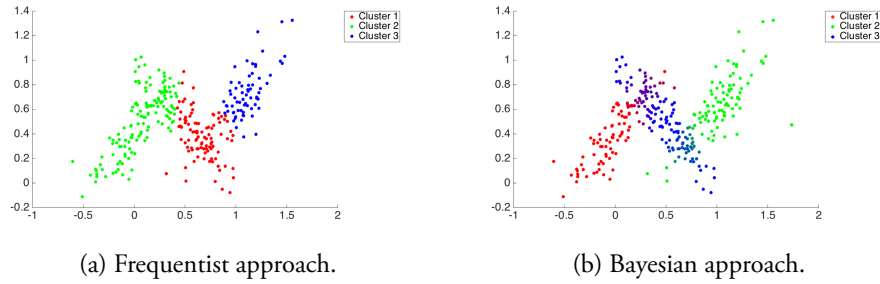


Figure 2.1: The difference between frequentist and Bayesian approach for inference. In (a) the result from categorising observations into three different distributions using a frequentist approach is presented and in (b) using a Bayesian approach.

Much has been said about the two approaches and why one is preferable to the other. It can be concluded that both are useful in their own context. Here, a Bayesian approach will be used for inference, assuming some prior knowledge about the problem.

In Bayesian inference the interest lays in the posterior distribution, the distribution of the parameter of interest given the observations. This can be expressed using Bayes' theorem.

#### Bayes' theorem

$$P(\theta | x) = \frac{P(x | \theta)P(\theta)}{P(x)} \quad (2.1)$$

$$P(x) = \int P(x | \theta)P(\theta)d\theta. \quad (2.2)$$

$x$  is the observation and  $\theta$  the parameter of interest. The probability of  $\theta$  given  $x$  equals the probability of  $x$  given  $\theta$  multiplied with the probability of  $\theta$ , the prior, divided by the probability of  $x$ .

**Example: Smoking and cancer** *Assumed a study is made where the smoking habits of 10,000 individuals are observed and they are screened for lung cancer. In the study it is found that the conditional probabilities of having cancer given being a smoker and non-smoker are*

$$P(\text{cancer} | \text{smoker}) = 0.34, \quad P(\text{cancer} | \text{non-smoker}) = 0.03. \quad (2.3)$$

*The probability of having cancer, being a smoker and non-smoker are*

$$P(\text{cancer}) = 0.0641, \quad P(\text{smoker}) = 0.11, \quad P(\text{non-smoker}) = 0.89.$$

The interest lays in the probability of being a smoker or non-smoker given that one has cancer. Using Bayes' theorem this can be express as

$$P(\text{smoker} | \text{cancer}) = \frac{P(\text{cancer} | \text{smoker})P(\text{smoker})}{P(\text{cancer})}$$

$$P(\text{non-smoker} | \text{cancer}) = \frac{P(\text{cancer} | \text{non-smoker})P(\text{non-smoker})}{P(\text{cancer})}.$$

Insert the probabilities obtained from the study into the equations above and obtain

$$P(\text{smoker} | \text{cancer}) = \frac{0.34 \cdot 0.11}{0.0641} = 0.5835$$

$$P(\text{non-smoker} | \text{cancer}) = \frac{0.03 \cdot 0.89}{0.0641} = 0.4165.$$

It is found that the probabilities of being a smoker or non-smoker given that one has cancer are fairly similar even though the probability of having cancer given one is a smoker or non-smoker differs greatly.

The variable of interest  $\theta$  will be a class of distributions and is specific to each individual problem. Bayesian inference can be seen as consisting of two parts, (i) the likelihood  $p(\mathbf{x} | \theta)$  and (ii) the prior distribution as the probability  $p(\theta)$ . The first part is the likelihood function of  $\theta$ ,  $\mathcal{L}(\theta | \mathbf{x}) = p(\mathbf{x} | \theta)$ . The likelihood gives the probability of the observations coming from the distribution specified by  $\theta$ . The second part, the prior distribution, gives the probability of  $\theta$ . With Bayes' theorem the posterior distribution is expressed as

$$p(\theta | \mathbf{x}) = \frac{p(\mathbf{x} | \theta)p(\theta)}{p(\mathbf{x})}.$$

In a more compact form it is expressed as

$$\underbrace{p(\theta | \mathbf{x})}_{\text{posterior}} \propto \underbrace{\mathcal{L}(\theta | \mathbf{x})}_{\text{likelihood}} \underbrace{p(\theta)}_{\text{prior}},$$

where the normalising constant has been removed.

**Example: Graphical model** A graphical model is an illustration of the structure of a model. Here, an example of a graphical model of a Bayesian model is given. The observations  $\mathbf{x}$  are given. The prior is that  $\mathbf{x}$  is normally distributed with known mean  $\mu$  and variance  $\sigma^2$ . The graphical model can be seen in figure 2.2.  $\mu$  and  $\sigma^2$  are the hyperparameters of  $\theta$ . In this case the interest is in the posterior distribution of  $\theta$  given the observations  $\mathbf{x}$  and hyperparameters  $\mu$  and  $\sigma^2$ .

$$p(\theta | \mathbf{x}, \mu, \sigma^2) \propto \mathcal{L}(\theta | \mathbf{x})p(\theta | \mu, \sigma^2). \quad (2.4)$$

The posterior distribution has the two components as described above, the likelihood and a prior distribution. In this case the prior is dependent on the parameters  $\mu$  and  $\sigma^2$ , since

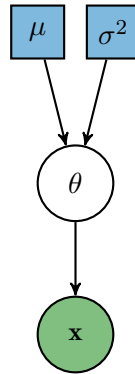


Figure 2.2: A graphical model of a Bayesian model with normal distributed prior with known mean and variance.

*they define the distribution. In the graphical model this is described by the arrows from  $\mu$  and  $\sigma^2$  to  $\theta$ . The blue squares are parameters set by the user, the white circles are parameters to be inferred on and the green circle is the observations. What is described here is a simple model which can be extended, as will be shown in coming chapters.*

The more observations there are the more weight is put on the likelihood. In cases with many observations the likelihood has a greater effect on the posterior than the prior. As the number of observations tends to infinity the posterior distribution will peak sharply around the likelihood. In cases when the observations are sparse the prior plays a greater role in obtaining meaningful results. The choice of prior will depend on the problem and the available information. Broadly it can be distinguished into two classes of priors, (i) informative and (ii) non-informative. Informative priors can be restrictions on a parameter or when there is strong evidence of its characteristics. Non-informative priors reflect uncertainty and lack of information about the characteristics.

## 2.3 Methods of inference

In cases where the posterior distribution is intractable, approximation schemes are used. These methods can broadly be categorised into two groups, (i) deterministic and (ii) stochastic approximations. This section introduces two deterministic approximation methods and one stochastic approximation method. In this section the main ideas of the methods and their strengths and weaknesses will be presented.

### 2.3.1 Laplace approximation

The Laplace method is an approximation method based on a Gaussian approximation of the distribution. The Gaussian approximation gives an integral of an exponential.

A simple one-dimensional example is

$$p(x) = \frac{1}{X} f(x), \quad X = \int f(x) dx, \quad (2.5)$$

where the aim is to approximate the distribution of  $p(x)$  with a Gaussian distribution centered at the mode of  $p(x)$ .  $X$  being the normalising constant for  $f(x)$ . The mean of the Gaussian approximation is expressed as

$$\left. \frac{df(x)}{dx} \right|_{x=x_0} = 0. \quad (2.6)$$

A Gaussian distribution is a quadratic function of its variables. Therefore a Taylor expansion of second order of the logarithm centred on the mode of  $p(x)$ ,  $x_0$ , is considered.

$$\ln f(x) \approx \ln f(x_0) - \frac{1}{2} \left. \frac{d^2}{dx^2} \ln f(x) \right|_{x=x_0} (x - x_0)^2. \quad (2.7)$$

By taking the exponential,

$$f(x) \approx f(x_0) \exp\left(-\frac{1}{2} \left. \frac{d^2}{dx^2} \ln f(x) \right|_{x=x_0} (x - x_0)^2\right) \quad (2.8)$$

is obtained. Now the approximative distribution of  $p(x)$  using the form of the Gaussian distribution can be obtained as

$$p(x) \approx \underbrace{\left(\frac{1}{2\pi} \left. \frac{d^2}{dx^2} \ln f(x) \right|_{x=x_0}\right)^{1/2}}_{1/X} \underbrace{\exp\left(-\frac{1}{2} \left. \frac{d^2}{dx^2} \ln f(x) \right|_{x=x_0} (x - x_0)^2\right)}_{f(x)}. \quad (2.9)$$

The same scheme can be applied to complex conditional distributions. To bear in mind when applying Laplace approximation is that the approximated distribution will be Gaussian, regardless of the true distribution. For the approximation to be representative, the distribution has to be similar to a Gaussian distribution. Otherwise global properties of the true distribution can be lost. The Laplace approximations performance is also sensitive to the number of observations available. The approximation of a Gaussian will become increasingly better the more observations available, as a result of the central limit theorem. Another downside of the Laplace approximation is that, since it is based on a Gaussian approximation, it is only applicable to real variables. This can in some cases be handled by applying the Laplace approximation to a transformation of the variable. [Bishop, 2007] The Laplace approximation is easily implemented, has a low computational cost and if used in the right context gives satisfying results.

### 2.3.2 Variational Bayes

Variational Bayes is an approximation method that adopts a more global approach than the Laplace approximation to solve some of its shortcomings. Variational Bayes (VB), as the Laplace approximation, is a deterministic approximation method used to approximate intractable posterior distributions.

When the distribution of interest is  $p(z|y)$ , it is approximated with the distribution  $q(z) \approx p(z|y)$ , where  $q(z)$  is restricted to a family of distributions being simpler than the true distribution.  $q(z)$  is called the variational distribution. To restrict  $q(z)$  to be simpler than  $p(z|y)$  the elements of  $z$  is partitioned into groups such as

$$q(z) = \prod_{i=1}^M q_i(z_i). \quad (2.10)$$

Apart from the partition there are no further assumptions made. The validity of the method depends on the partitioning of  $q(z)$ . The stronger the assumptions made upon  $q(z)$  are, the stronger are the restrictions introduced. Depending on the validity of these assumptions the approximation of the true posterior distribution can be more or less accurate. In many cases it is needed to make assumptions to obtain a tractable expression.

Using calculus of variations the optimal solution can be expressed as follows for each factor of the partition

$$q_j^*(z_j) = \frac{\exp(\mathbb{E}_{i \neq j}[\ln p(y, z)])}{\int \exp(\mathbb{E}_{i \neq j}[\ln p(y, z)]) dZ_j}. \quad (2.11)$$

For example, the factors can be parameters of a model. Equation 2.11 shows that there is a dependency between the factors. The factors are updated by first initialising them and then updating them one by one in a cycle. To obtain the optimal solution the update cycle is iterated until a convergence criteria is reached.

The drawbacks of VB are that the derivation of the update equations can be cumbersome and the assumptions made affects the validity of the result. It is also possible that the solution obtained is a local optima. This can to some extent be overcome by running the optimisation several times with different initialisations. This will increase the computational time. Still, the variational Bayes approach is relatively computational inexpensive and with reasonable assumptions obtains satisfying approximations of the true distribution. [Bishop, 2007]

### 2.3.3 Monte Carlo

Monte Carlo methods is a group of methods for stochastic approximations. The methods repeatedly draws random samples to obtain a numerical result. In comparison to the two mentioned deterministic approximation methods, the Monte Carlo approach

could obtain the true posterior distribution, i.e. no approximation, given unlimited computational power and time. Since unlimited computational power and time is not available, the obtained result will be an approximation of the truth.

The obtained estimate of the posterior rely on the law of large numbers

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=1}^n f(x_j) = \mathbb{E}[f(x)]. \quad (2.12)$$

The more samples (larger  $n$ ), the better the approximation. One difficulty in using Monte Carlo methods is to decide how many samples are needed for a satisfying approximation. The number of samples needed often increases with the complexity of the model. This leads to the methods becoming computationally costlier.

The advantage of the Monte Carlo methods over Laplace approximation and variational Bayes is that it can be used even for problems where it is intractable to obtain the true posterior. No assumptions are made and there is great flexibility in the design of the model. The downside is the computational cost, since it can take many times the computational time needed for the deterministic methods. A simplified comparison to VB is that with VB one spends more time deriving the equations and with Monte Carlo methods one spends more time on the simulation of samples.

**Example: Ice cream in the dark** *Charles and Lucy are living together and both love ice cream. Charles' favourite flavour is chocolate while Lucy's is vanilla. They store the ice cream in their fridge in the basement and unfortunately the lightbulb is broken so it is pitch black and impossible to differentiate between chocolate and vanilla ice cream. Charles and Lucy wonder what flavour they will get when randomly taking a scoop of ice cream in the dark. The posterior distribution of flavour given the ice cream in the fridge is*

$$p(\text{flavour} | \text{in fridge}) \propto \mathcal{L}(\text{flavour} | \text{in fridge})p(\text{flavour}). \quad (2.13)$$

*By letting Charles and Lucy sample the ice cream in the fridge during two weeks, observations of what is in the fridge is obtained.  $p(\text{flavour})$  is a distribution over the flavours, which is decided based on who has done the shopping lately. 10,000 samples were randomly drawn from the posterior to obtain an approximation of the distribution. The result from the samples is*

Chocolate	Vanilla
3289	6711

*The probability of getting a scoop of chocolate ice cream is 32.89% and 67.11% of getting vanilla. It looks like Lucy will be the happy one.*



## 2.4 Markov chain Monte Carlo

Within Monte Carlo methods there is a subgroup of methods called Markov chain Monte Carlo methods. These, as the Monte Carlo, are sampling methods but the sampling can be described as a random walk. The random walk is based on a Markov chain, hence the name, and its properties. Markov chain Monte Carlo is the method of inference used in this thesis and in this section the foundations of these methods, as described in [Gamerman and Lopes, 2006; Chib and Greenberg, 1995], will be reviewed.

### 2.4.1 Markov chains

A Markov chain is a sequence of stochastic variables that has the property of only being dependent on the previous value,

$$P(\theta^n | \theta^{n-1}, \dots, \theta^0) = P(\theta^n | \theta^{n-1}). \quad (2.14)$$

**Example: Markov chain** *If flying across the globe the destination of the moment does not depend on any other destination than the previous one. The connections from the previous destinations defined the possible destinations. The destinations one visited before that are independent of ones current destination. Figure 2.3 shows an example of this Markov chain.*

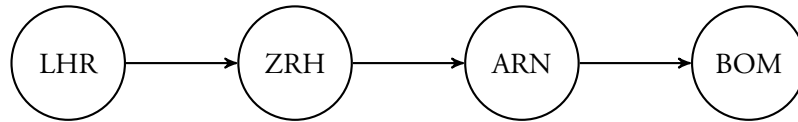


Figure 2.3: Example of Markov chain.

Depending on its design, the chain can have different characteristics. Some of those properties are crucial for Markov chain Monte Carlo (MCMC) and will be described below.

#### Periodic

A state of the chain is periodic if it must occur with a specific periodicity  $k$ .

$$k = \gcd\{n \in \mathbb{N}_+ : P(\theta_n = i | \theta_0 = i) > 0\}, \quad (2.15)$$

where  $\gcd(\cdot)$  is the greatest common divisor. If  $k = 1$  the state is said to be aperiodic and can be visited irregularly. If every state of the chain is aperiodic the Markov chain is also aperiodic.

**Irreducible**

If the probability of transition from state  $\theta$  to state  $\phi$  is non-zero, state  $\phi$  is said to be irreducible,

$$P(\phi | \theta) > 0. \quad (2.16)$$

In the case of positive probability of reaching every state from every state, the Markov chain is said to be irreducible.

**Recurrent**

A recurrent Markov chain is a chain where the probability of returning to state  $i$ , the starting state, is non-zero. Let  $T_i$  be the time until the chain returns to state  $i$  for the first time.

$$T_i = \{n \geq 1 : \theta_n = i | \theta_0 = i\}. \quad (2.17)$$

The probability of the returning time being less than infinity equals one for a recurrent chain.

$$P(T_i < \infty) = 1. \quad (2.18)$$

**Transition kernel**

The transition kernel is the probability of the transition from state  $\phi$  to state  $\theta$ ,  $K(\theta, A) := P(\phi \in A | \theta)$  where  $A$  is a subset of the state space. The marginal likelihood of state  $\phi$  is

$$\pi^n(\phi) = \int_{\Omega} \pi^{n-1}(\theta) K(\theta, \phi) d\theta. \quad (2.19)$$

**Invariant**

For a chain  $\pi$  with transition kernel  $K(\theta, \phi)$  to be invariant, the following equation must hold

$$\pi(\phi) = \int_{\Omega} \pi(\theta) K(\theta, \phi) d\theta. \quad (2.20)$$

This implies,

$$\pi(\phi) K(\phi, \theta) = \pi(\theta) K(\theta, \phi). \quad (2.21)$$

In the case of the chain being both irreducible and recurrent the chain is also invariant and  $\pi$  is unique. In the case of the chain being irreducible and aperiodic the transition kernel tends to the invariant distribution  $\pi$ .

### Properties for Markov chain Monte Carlo

The properties that need to be met for MCMC is the chain to be (i) invariant and (ii) aperiodic. If these properties are met, the chain will have a stationary distribution. The chain having a stationary distribution means that regardless of starting point it will tend to the stationary distribution, the equilibrium distribution of the chain.

#### 2.4.2 Metropolis-Hastings

The Metropolis-Hastings (MH) algorithm describes a sampling scheme useful in the case of the distribution of interest  $\pi$  being expensive or impossible to sample from. Given the distributions it is needed to define the transition kernel. This can be done in several ways and will be discussed later.

The transition kernel  $K(\theta, \phi)$ , a transition from  $\theta$  to  $\phi$ , can be seen as two parts, an arbitrary proposal kernel  $q(\theta, \phi)$  and a probability  $\alpha(\theta, \phi)$ . The probability of reaching a new state is

$$K(\theta, \phi) = q(\theta, \phi)\alpha(\theta, \phi), \quad \theta \neq \phi \quad (2.22)$$

and the probability of reaching the same state is

$$K(\theta, \theta) = 1 - \int_{\Omega} q(\theta, \phi)\alpha(\theta, \phi)d\phi. \quad (2.23)$$

It is possible to express the probability of reaching any state,  $A$  of the parameter space as

$$K(\theta, A) = \int_A q(\theta, \phi)\alpha(\theta, \phi)d\phi + \mathbf{1}_{\theta \in A}[1 - \int_{\Omega} q(\theta, \phi)\alpha(\theta, \phi)d\phi]. \quad (2.24)$$

From equation 2.24 the acceptance probability  $\alpha(\theta, \phi)$  can be derived as

$$\alpha(\theta, \phi) = \min\left(1, \frac{\pi(\phi)q(\phi, \theta)}{\pi(\theta)q(\theta, \phi)}\right), \quad (2.25)$$

where  $\pi$  is the equilibrium distribution and the following expression holds

$$\pi(\theta)K(\theta, \phi) = \pi(\phi)K(\phi, \theta), \quad \forall(\theta, \phi). \quad (2.26)$$

The MH algorithm does not accept all draws as coming from the distribution  $\pi$ . For the draw to be accepted as being the next step in the chain and coming from the distribution  $\pi$  the probability of  $\alpha$  has to be computed. For a draw being more probable than the current state results in an acceptance probability of 1, a draw less probable than the current state results in an acceptance probability of less than 1.

The algorithm is constructed as seen in algorithm 1.

---

**Algorithm 1** Metropolis-Hastings algorithm

---

**Require:** Initialise  $\theta^0$ . Set  $N$ , the number of iterations,

```

1: for  $n = 1:N$  do
2:   Move the chain to  $\phi$  generated from the density  $q(\theta^{n-1}, \cdot)$ 
3:   Compute the probability of acceptance  $\alpha(\theta^{n-1}, \phi)$ 
4:   if  $\alpha > u \sim \mathcal{U}(0, 1)$  then
5:      $\theta^n = \phi$ 
6:   else
7:      $\theta^n = \theta^{n-1}$ 
8:   end if
9: end for

```

---

**Transition kernel**

The transition kernel defines the walk across the sample space. There are different approaches to how to construct the transition kernel. The basic approaches to transition kernels will be described and other approaches discussed briefly.

**Symmetric chains** A chain is symmetric if its transition kernel is symmetric,  $K(\theta, \phi) = K(\phi, \theta)$  for all  $(\theta, \phi)$ . The symmetry leads to that the probability of acceptance can be reduced to  $\alpha = \min\left(1, \frac{\pi(\phi)}{\pi(\theta)}\right)$ . In the context of the MH algorithm the symmetry is applied to the proposal kernel  $q$ .

**Random walk chains** The random walk transition is when the new state equals a random step from the current state,  $\theta^n = \theta^{n-1} + \epsilon_n$  where  $\epsilon_n$  is the random step.  $\epsilon$  can take many forms, for example, normal or Student's-t distribution. In the case of  $\epsilon$  being symmetric around 0, the proposal kernel is symmetric and  $\alpha$  can be reduced. In this case the proposal kernel only depends on the distance between the two states,  $q(\theta, \phi) = q_R(\phi - \theta)$ .

**Independence chains** For independence chains the proposal kernel is independent of the current state,  $q(\theta, \phi) = q_I(\phi)$ . Although the proposal kernel is independent, the transition kernel is still dependent on the current state via  $\alpha(\theta, \phi)$ .

**Other chains** Beyond the three methods shown above there are many ways to construct the proposal kernel, for instance, using an autoregressive chain of order one or letting the transition proposal have the same shape as  $\pi$ . The different kernels all have their strengths and weaknesses which has to be considered when choosing a transition kernel for a specific application.

### Step size & acceptance rate

The efficiency and accuracy of the MH algorithm is greatly affected by how the state space is explored. The step size, the distance between current state and a new draw, can be modified to obtain a more efficient exploration of the state space. The step size affects the acceptance rate, number of accepted draws divided by the number of iterations, and thereby the performance of the algorithm.

When selecting the step size the aim is to allow large enough steps to explore the state space adequately in an efficient manner while retaining an acceptable acceptance rate for the sake of computational cost. Both a too small and a too large step size leads to inefficient exploration of the state space and poor performance of the MH algorithm. As a general rule of thumb, an acceptance rate in the span of 20 % – 50 % yields a balance between efficiency and accuracy [Besag et al., 1995; Bennett et al., 1996].

### 2.4.3 Gibbs sampling

Gibbs sampling is a special case of MH where every sample is accepted. In comparison to MH, Gibbs sampling can only be applied when the conditional distributions are fully known. For the Gibbs sampling the transition kernel is formed by the full conditional distribution and every draw is accepted as a part of the chain. Assume the distribution of interest is  $\pi(\boldsymbol{\theta})$ , where  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_d)$ , and that the conditional distributions  $\pi_i(\theta_i) = \pi_i(\theta_i | \boldsymbol{\theta}_{-i})$ , where  $\boldsymbol{\theta}_{-i} = \boldsymbol{\theta} \setminus \theta_i$ , are known. When draws directly from  $\pi(\boldsymbol{\theta})$  are costly or impossible, while draws from  $\pi(\boldsymbol{\theta}_{-i})$  are simple and possible, Gibbs sampling provides a scheme to generate draws from  $\pi(\boldsymbol{\theta})$ . The method takes the following steps presented in algorithm 2.

---

#### Algorithm 2 Gibbs sampling

---

**Require:** Initialise  $\boldsymbol{\theta}^0 = (\theta_1^0, \dots, \theta_d^0)$ . Set N, the number of iterations.

- 1: **for** n = 1:N iterations **do**
  - 2:    $\theta_1^n \sim \pi(\theta_1 | \boldsymbol{\theta}_{-1}^{n-1})$
  - 3:    $\theta_2^n \sim \pi(\theta_2 | \theta_1^n, \theta_3^{n-1}, \dots, \theta_d^{n-1})$
  - 4:    $\vdots$
  - 5:    $\theta_d^n \sim \pi(\theta_d | \boldsymbol{\theta}_{-d}^n)$
  - 6: **end for**
- 

### 2.4.4 Inversion sampling

According to the probability integral transform it is known that a continuous random variable  $X$  has a cumulative distribution  $F_X$  that is uniformly distributed between 0 and 1,  $F_X \sim \mathcal{U}(0, 1)$ . Given that the inverse of the cumulative distribution exists,  $F_X^{-1}$ , it is possible to use the inverse to sample from the distribution of  $F_X$  [Gamerman and Lopes, 2006].

Stating that

$$F_X^{-1}(u) = \inf\{x \mid F_X(x) \geq u\} \quad 0 < u < 1, \quad (2.27)$$

which can be expressed as

$$P(F_X^{-1}(u) \leq x) = P(u \leq F_X(x)) \quad (2.28)$$

$$= F_X(x). \quad (2.29)$$

This equality is used to sample from the distribution described by  $F_X$ , as seen in algorithm 3.

---

**Algorithm 3** Inversion sampling

---

- 1: Generate  $u \sim \mathcal{U}(0, 1)$
  - 2:  $x = F_X^{-1}(u)$
- 

### 2.4.5 Burn-in

When the samples are drawn they should come from the equilibrium distribution, which is the target distribution. To do so the chain has to be at its steady state. Since the sampling starts by drawing a sample with respect to a initialised state it might take a while until the chain converges and reaches the steady state. This period until reaching the equilibrium distribution is called *burn-in*. A certain number of iterations is set as burn-in, a number chosen by the user. The chosen number of iterations should be large enough to allow convergence. When convergence is reached the samples are drawn from the equilibrium distribution of the chain and can be used for inference. The number of samples needed for burn-in varies depending, amongst other things, on the complexity of the model and the initialisation.

## Chapter 3

# Mixture Models

### 3.1 Introduction

In real world applications it is not unusual that the data consists of a mixture of several distributions, for instance, representing healthy and diagnosed subjects or different species within a genus<sup>1</sup>. The data can be seen as a weighted sum of independent distributions from different sources. A way to model the data under these circumstances is a mixture model, as in figure 3.1. A mixture model decomposes a complicated distribution into several simpler distributions. Each distribution describing a subgroup of the dataset.

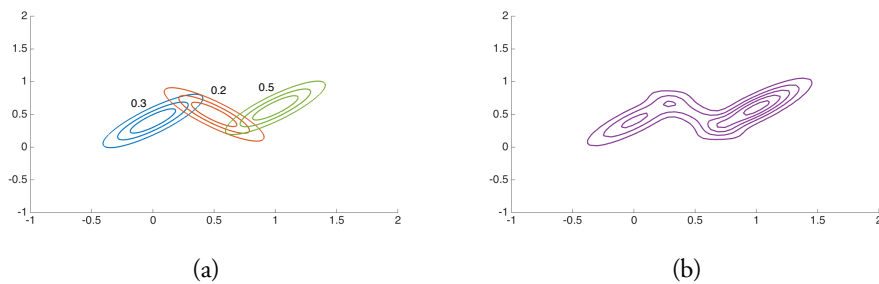


Figure 3.1: An example of a Gaussian mixture distribution in two dimensions. In (a) the contours of the three distributions are shown in red, blue and green with their respective mixing proportion. In (b) the contour of the joint probability of the three mixture distributions is shown.

A mixture model is a sum of weighted distributions where the weights sum to one.

$$p(X) = \sum_k w_k p_k(X), \quad (3.1)$$

---

<sup>1</sup>Genus is a taxonomic rank used in the biological classification of living and fossil organisms. A genus consists of several species and is a part of a family.

where  $w_k$  is the weight of distribution  $p_k$  with  $\sum_k w_k = 1$ ,  $0 < w_k \leq 1$ , and  $X$  a random variable. In this constellation there is no restriction on  $p_k$ , except it being a probability distribution. The individual mixture distributions could be of different distribution families or from the same family but having different parameter values.

Making the assumption that the mixtures are multivariate Gaussian distributed equation 3.1 can be expressed as

$$p(\mathbf{y} | \boldsymbol{\mu}, \mathbf{S}, \boldsymbol{\pi}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{y} | \boldsymbol{\mu}_k, \mathbf{S}_k^{-1}), \quad (3.2)$$

where  $\boldsymbol{\mu}_k$  is the mean,  $\mathbf{S}_k$  the precision and  $\pi_k$  the mixture proportion, the weight, for distribution  $k$  belonging to the sets  $\boldsymbol{\mu}$ ,  $\mathbf{S}$  and  $\boldsymbol{\pi}$ .

While using a Gaussian mixture model the goal is to *infer* on the parameters of the different distributions as well as the assignment of each observation. In this chapter two models and their implementations will be introduced. First a (i) finite mixture model will be proposed, where the number of distributions in the mixture are finite. The finite model will later be extended to an (ii) infinite mixture model, following [Rasmussen, 2000; Neal, 2000].

## 3.2 Finite mixture of Gaussians

The finite mixture of Gaussians model has a fixed number  $K$  of Gaussian distributions. The observations are clustered into one of the  $K$  mixtures. The finite model is constructed in such a way that it is easily extended to the infinite case, which will highlight the addition of inference on the number of clusters. A graphical model for the finite mixture model can be seen in figure 3.2. The number of clusters is indicated by  $K$  and the number of observations is indicated by  $N$ .  $\boldsymbol{\mu}$ ,  $\boldsymbol{\pi}$  and  $\mathbf{S}$  are the parameters of the mixture distributions with hyperparameters  $\boldsymbol{\lambda}$ ,  $r$ ,  $\mathbf{W}$ ,  $v$ .  $\mathbf{c}$  is a latent class with parameter  $\alpha$ .

### 3.2.1 Model

The model is build upon the assumption of the mixtures being multivariate Gaussian distributed with unknown mean and precision. The mixture proportions are distributed as a symmetric Dirichlet distribution

$$\boldsymbol{\pi} | \alpha \sim \text{Dir}(\alpha/K, \dots, \alpha/K). \quad (3.3)$$

Each observation is assigned to one of the mixture distributions. The assigned mixture is indicated by the latent class  $\mathbf{c}$ .  $\mathbf{c}$  takes a integer value of  $(1, \dots, K)$ , where  $K$  is



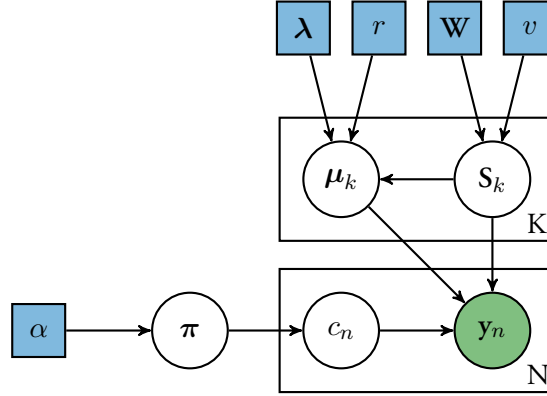


Figure 3.2: A graphical model of the finite Gaussian mixture model with normal-Wishart prior.

the number of mixtures, for each observation and is distributed as

$$p(\mathbf{c} | \boldsymbol{\pi}) = \prod_{k=1}^K \pi_k^{n_k}, \quad n_k = \sum_{n=1}^N \delta_{c_n, k}, \quad (3.4)$$

with  $n_k$  being the number of observations assigned to cluster  $k$ ,  $\delta_{c_n, k}$  the Kronecker delta and  $\pi_k$  the mixing proportion.

The priors of the mixture distributions are chosen as the conjugate priors. Conjugate priors simplifies the equations and reassures the posterior distribution is of the same family as the prior. The conjugate prior for a multivariate Gaussian distribution with unknown mean and precision is the normal-Wishart distribution

$$\boldsymbol{\mu}_k, \mathbf{S}_k | \boldsymbol{\lambda}, r, \mathbf{W}, v \sim \text{NW}(\boldsymbol{\lambda}, r, \mathbf{W}, v). \quad (3.5)$$

The normal-Wishart distribution can be decomposed into two conditional distributions for the mean and the precision. The conditional distribution for the mixture precision is a Wishart distribution and can be expressed as

$$\mathbf{S}_k | \mathbf{W}, v \sim \mathcal{W}_D(\mathbf{W}, v), \quad (3.6)$$

where  $\mathbf{W}$  is the scale matrix,  $v$  the degrees of freedom and  $D$  the dimensionality. The conditional distribution for the mixture mean is a Gaussian distribution,

$$\boldsymbol{\mu}_k | \boldsymbol{\lambda}, r, \mathbf{S}_k \sim \mathcal{N}(\boldsymbol{\lambda}, (r\mathbf{S}_k)^{-1}), \quad (3.7)$$

with mean  $\boldsymbol{\lambda}$  and precision  $r\mathbf{S}_k$ .

### 3.2.2 Posterior conditional distributions

The interest lies within the posterior distributions of the parameters of the model. In this section the posterior distributions will be derived for each parameter. The posterior

distributions obtained will be used for sampling. The posterior distributions for  $\mathbf{c}$ ,  $\boldsymbol{\mu}_k$  and  $\mathbf{S}_k$  are easily derived given equation 3.2-3.7.  $\boldsymbol{\mu}_k$  and  $\mathbf{S}_k$  will be of the same distribution family as their priors since conjugate priors are used.

### Collapsed- and instantiated-weights

In the case at hand, a finite mixture model, where the mixture proportion  $\boldsymbol{\pi}$  is Dirichlet distributed there are two approaches to choose from. The two approaches are (i) instantiated-weights and (ii) collapsed-weights. With instantiated-weights the mixture proportions will be explicitly represented and samples drawn to obtain an estimate. In the case of collapsed-weights the mixture posteriors is marginalised out. The mixture proportions are not sampled in this case but it is possible to infer on the mixture proportion from the assignment samples. In some occasions the probability of the draws from the prior of the mixture proportions fitting the data is low. For instantiated-weights this results in slow convergence. The advantage compared to collapsed-weights is that it can be parallelised easily which greatly affects the computational time. With the collapsed-weights approach the mixture proportion  $\boldsymbol{\pi}$  is marginalised over which simplifies the sampling in the case of an infinite mixture model. [Chang and Fisher III, 2013]

To easily be able to extend the finite mixture of Gaussians model to an infinite version the collapsed-weights approach is chosen. Hence an expression for the posterior of the mixture proportion will not be derived and the mixture proportion will not be part of any posterior distribution.

### Conditional posterior of $c_n$

The posterior of  $c_n$  can be expressed as

$$p(c_n = k | \mathbf{y}, \boldsymbol{\mu}_k, \mathbf{S}_k, \alpha, \mathbf{c}_{-n}) \propto p(\mathbf{y}_n | \boldsymbol{\mu}_k, \mathbf{S}_k, \mathbf{c}) p(c_n = k | \mathbf{c}_{-n}, \alpha) \quad (3.8)$$

By integrating out the mixing proportions one obtains

$$\begin{aligned} p(\mathbf{c} | \alpha) &= \int p(\mathbf{c} | \boldsymbol{\pi}) p(\boldsymbol{\pi}) d\boldsymbol{\pi} \\ &= \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \int \prod_{k=1}^K \pi^{n_k + \alpha/K - 1} d\boldsymbol{\pi}_k \\ &= \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \frac{\Gamma(\alpha/K + n_k)^K}{\Gamma(\alpha + n)} \\ &= \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)} \prod_{k=1}^K \frac{\Gamma(\alpha/K + n_k)}{\Gamma(\alpha/K)} \end{aligned} \quad (3.9)$$

with respect to  $\int \prod_{k=1}^K \pi_k^{\alpha/K-1} d\pi_k = \Gamma(\alpha/K)^K / \Gamma(\alpha)$  since the integral over the whole distribution equals 1. With equation 3.9 the conditional distribution for  $c_n$  is obtained as

$$p(c_n = k | \mathbf{c}_{-n}, \alpha) = \frac{P(\mathbf{c} | \alpha)}{p(\mathbf{c}_{-n} | \alpha)} = \frac{n_{k,-n} + \alpha/K}{N + \alpha - 1} \quad (3.10)$$

The conditional posterior can now be expressed as

$$p(c_n = k | \mathbf{y}_n, \boldsymbol{\mu}_k, \mathbf{S}_k, \mathbf{c}, \alpha) \propto \frac{n_{k,-n} + \alpha/K}{N + \alpha - 1} |\mathbf{S}_k|^{1/2} \exp\left(-\frac{1}{2}(\mathbf{y}_n - \boldsymbol{\mu}_k)^T \mathbf{S}_k (\mathbf{y}_n - \boldsymbol{\mu}_k)\right) \quad (3.11)$$

### Conditional posterior of $\mathbf{S}_k$

The posterior distribution of  $\mathbf{S}_k$  is a Wishart distribution. With equation 3.2 and 3.6 one obtains

$$\mathbf{S}_k | \mathbf{y}_k, \mathbf{W}, v, \boldsymbol{\lambda}, r \sim \mathcal{W}_D(\mathbf{W}', v + n_k), \quad (3.12)$$

where

$$\mathbf{W}' = \left( \mathbf{W}^{-1} + \sum_{n=1}^{n_k} (\mathbf{y}_n - \bar{\mathbf{y}}_k)(\mathbf{y}_n - \bar{\mathbf{y}}_k)^T + \frac{rn_k}{r + n_k} (\boldsymbol{\lambda} - \bar{\mathbf{y}}_k)(\boldsymbol{\lambda} - \bar{\mathbf{y}}_k)^T \right)^{-1}.$$

Here  $\mathbf{y}_k$  are the observations assigned to cluster  $k$  and  $\bar{\mathbf{y}}_k$  is their mean.

### Conditional posterior of $\boldsymbol{\mu}_k$

The posterior of  $\boldsymbol{\mu}_k$  has a Gaussian distribution. With equation 3.2, 3.7 together with 3.6 one obtains

$$\boldsymbol{\mu}_k | \mathbf{y}_k, \mathbf{S}_k, \boldsymbol{\lambda}, r \sim \mathcal{N}\left(\frac{n_k \bar{\mathbf{y}}_k + r \boldsymbol{\lambda}}{n_k + r}, ((n_k + r) \mathbf{S}_k)^{-1}\right). \quad (3.13)$$

### 3.2.3 Sampling

With the posterior distributions from section 3.2.2 the sampling is set up as seen in algorithm 4. The sampling is straight forward and can easily be implemented using the methods described in section 2.4.  $\boldsymbol{\mu}_k$  and  $\mathbf{S}_k$  can be sampled using Gibbs sampling, as described in section 2.4.3, thanks to their posterior distributions being Gaussian and Wishart respectively. The assignments  $c_n$  is sampled using inversion sampling, as described in section 2.4.4.

**Algorithm 4** Finite mixture model

**Require:** Initialise  $\boldsymbol{\mu}^0$ ,  $\mathbf{S}^0$ ,  $\mathbf{c}^0$ . Set  $M$ , the number of iterations.

```

1: for  $l = 0 : M - 1$  do
2:   for  $n = 1 : N$  do
3:      $c_n^{l+1} \sim p(c_n = k | \mathbf{y}_n, \mathbf{c}_{-n}^l, \boldsymbol{\mu}^l, \mathbf{S}^l)$ 
4:   end for
5:   for  $k = 1 : K$  do
6:      $\mathbf{S}_k^{l+1} \sim p(\mathbf{S}_k | \mathbf{y}_k, \mathbf{c}^{l+1})$ 
7:      $\boldsymbol{\mu}_k^{l+1} \sim p(\boldsymbol{\mu}_k | \mathbf{y}_k, \mathbf{c}^{l+1}, \mathbf{S}_k^{l+1})$ 
8:   end for
9: end for

```

**3.2.4** Choosing  $K$ 

The choice of  $K$  is not trivial. In some cases a comparison, for example, between gender might lead to a natural choice of  $K$ . In other cases the goal might be to fully explore the data and then the choice is not as easy. To find the most suitable  $K$  one wants to compare models with different values of  $K$ . A standard way of doing so is Bayesian model comparison. In Bayesian model comparison the marginal likelihood of the different models are compared. The marginal likelihood is computed as

$$p(\mathbf{y} | m) = \int_{\Omega_m} p(\mathbf{y} | \boldsymbol{\theta}_m, m) p(\boldsymbol{\theta}_m | m) d\boldsymbol{\theta}_m. \quad (3.14)$$

The integral for each model is computed. To find the best model the Bayes factor is computed as

$$B = \frac{p(\mathbf{y} | m)}{p(\mathbf{y} | m')}, \quad (3.15)$$

where  $m$  and  $m'$  are two different models.

A more complex model will often have a better fit to the data. To avoid overly complex models complexity will be penalised. Bayesian model comparison does this via the distribution of  $p(\mathbf{y} | \boldsymbol{\theta}_m, m)$ . As the complexity increases the model's fit to the data will improve, hence the model will have a good fit to a broader set of observations. Since the fit improves for a broader set of observations the distribution will be flatter. The value of  $p(\mathbf{y} | \boldsymbol{\theta}_m, m)$  will therefore decrease, penalising the increase in complexity. [Bishop, 2007]

**3.3** Infinite mixture of Gaussians

The disadvantage of having finite number of mixtures is the need to define the number of clusters, which is not always trivial and reduces the exploratory strength of the

method. The infinite mixture model does not require the number of clusters to be defined, instead the number of clusters can be inferred on. The difficulty of choosing  $K$  is overcome by, instead of relying on model comparison, using a suitable prior. To expand the finite model to an infinite model there are two changes that need to be made, (i) taking the limit of  $K \rightarrow \infty$  and (ii) modifying the sampling of  $c$ . A graphical model for the infinite mixture model can be seen in figure 3.3.

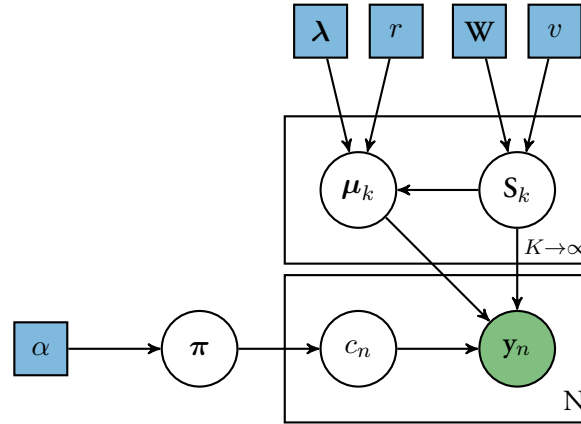


Figure 3.3: Graphical model of the infinite Gaussian mixture model with normal-Wishart prior.

### 3.3.1 Limit of $K$

The infinite model does not depend on  $K$ . An unlimited number of clusters are allowed in theory by taking the limit of  $K \rightarrow \infty$ . However, in practice the number of clusters with assigned observations is limited by the number of observations  $N$ . The first step is to take the limit of  $K$  of equation 3.10. The infinite counterpart is obtained as

$$p(c_n = k | \mathbf{c}_{-n}, \alpha) = \frac{n_{k,-n}}{N + \alpha - 1} \quad n_{k,-n} > 0 \quad (3.16)$$

$$p(c_n \neq c_{n'} \forall n \neq n' | \mathbf{c}_{-n}, \alpha) = \frac{\alpha}{N + \alpha - 1} \quad \text{otherwise.} \quad (3.17)$$

Given equations 3.16 and 3.17, the two possible cases are

$n_{k,-n} > 0$ : There are several observations assigned to cluster  $k$  and the posterior distribution is given by equation 3.18.

$n_{k,-n} = 0$ : There are no observations assigned to cluster  $k$ , except for observation  $n$ . This scenario will be seen as an unrepresented class as described in [Neal, 2000]. An unrepresented class is when there are no observations assigned to the cluster and the cluster parameters will be sampled from the prior distribution.

Here the scheme presented by [Neal, 2000] as *algorithm 2* is applied. For the unrepresented class the likelihood is integrated with respect to the prior distribution, obtaining

the posterior predictive distribution of the single observation  $\mathbf{y}_n$ . The posterior distributions are updated, equation 3.11, with the modified distributions of  $c_n$  above. The posterior distribution for the two cases are

$$\begin{aligned} p(c_n = k | \mathbf{y}_n, \boldsymbol{\mu}_k, \mathbf{S}_k, \mathbf{c}_{-n}, \alpha) &\propto p(c_n = k | \mathbf{c}_{-n}, \alpha) \mathcal{L}(\boldsymbol{\mu}_k, \mathbf{S}_k | \mathbf{y}_n) \\ &= \frac{n_{k,-n}}{N + \alpha - 1} \\ &\quad \mathcal{N}(\mathbf{y}_n | \boldsymbol{\mu}_k, \mathbf{S}_k^{-1}). \end{aligned} \quad (3.18)$$

$$\begin{aligned} p(c_n = k | \mathbf{y}_n, \boldsymbol{\mu}_k, \mathbf{S}_k, \mathbf{c}_{-n}, \alpha) &\propto \frac{\alpha}{N + \alpha - 1} \\ &\quad \int \mathcal{L}(\boldsymbol{\mu}_k, \mathbf{S}_k | \mathbf{y}_n) d\text{NW}(\boldsymbol{\lambda}, r, \mathbf{W}, v) \\ &= \frac{\alpha}{N + \alpha - 1} \\ &\quad t_{v-D+1} \left( \mathbf{y}_n | \boldsymbol{\lambda}, \frac{r+1}{r(v-D+1)} \mathbf{W}^{-1} \right) \end{aligned} \quad (3.19)$$

where  $t_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  is the multivariate t-distribution with  $d$  degrees of freedom, location  $\boldsymbol{\mu}$  and scale matrix  $\boldsymbol{\Sigma}$ .

### 3.3.2 Sampling

With equations 3.18 and 3.19 the posterior distributions needed for the sampling is given. Besides updating the equation the algorithm has to be modified to handle a varying number of clusters. This is done by first controlling the value of  $n_{k,-n}$ . In the case of  $n_{k,-n} = 0$  the cluster is seen as unrepresented. The cluster parameters for cluster  $k$  is then removed and  $K$  updated to the new number of clusters. The existing assignments are relabelled, assignments to cluster  $(1, \dots, k-1)$  keeps their labels and  $(k+1, \dots, K)$  are reduced by one to meet the new number of clusters. Then, regardless of  $n_{k,-n}$ , an assignment is sampled, which ranges between  $(1, \dots, K+1)$ . If the assignment is  $c_n = K+1$ ,  $K$  is updated to  $K = K+1$  and  $\mathbf{S}_K$  and  $\boldsymbol{\mu}_K$  are sampled. The rest of the algorithm remains unchanged. See algorithm 5 for the pseudocode.

---

**Algorithm 5** Infinite mixture model
 

---

**Require:** Initialise  $\boldsymbol{\mu}^0, \mathbf{S}^0, \mathbf{c}^0$ . Set  $M$ , the number of iterations.

```

1: for  $l = 0 : M - 1$  do
2:   for  $n = 1 : N$  do
3:     if  $n_{k,-n} = 0$  then
4:       Remove cluster & relabel remaining clusters.
5:        $K = K - 1$ .
6:     end if
7:      $c_n^{l+1} \sim p(c_n = k | \mathbf{y}_n, \mathbf{c}_{-n}^l, \boldsymbol{\mu}^l, \mathbf{S}^l)$ 
8:     if  $c_n$  is a new cluster then
9:        $K = K + 1$ .
10:       $\mathbf{S}_K^{l+1} \sim p(\mathbf{S}_K | \mathbf{y}_K, \mathbf{c}^{l+1})$ 
11:       $\boldsymbol{\mu}_K^{l+1} \sim p(\boldsymbol{\mu}_K | \mathbf{y}_K, \mathbf{c}^{l+1}, \mathbf{S}_K^{l+1})$ 
12:    end if
13:  end for
14:  for  $k = 1 : K$  do
15:     $\mathbf{S}_k^{l+1} \sim p(\mathbf{S}_k | \mathbf{y}_k, \mathbf{c}^{l+1})$ 
16:     $\boldsymbol{\mu}_k^{l+1} \sim p(\boldsymbol{\mu}_k | \mathbf{y}_k, \mathbf{c}^{l+1}, \mathbf{S}_k^{l+1})$ 
17:  end for
18: end for

```

---

## Chapter 4

# Label Switching

### 4.1 Introduction

When dealing with complicated models, MCMC provides a convenient way to draw inference. One example of a complicated model is a mixture model. When applying MCMC to draw inference on a mixture model there are some issues that might arise due to the characteristics of the model and MCMC sampling. With the MCMC samples the ergodic average is taken to obtain the parameter estimations. For the estimate to be accurate the sample trace should be stable. This indicates that the draws are made from a stationary distribution, the equilibrium distribution.

Mixture models are useful when the data comes from a mixture of simple distributions. The labelling of the different mixtures is said to be exchangeable. The exchangeability of the labels in combination with the characters of MCMC is what might create issues.

#### 4.1.1 Exchangeability of random variables

Random variables are exchangeable, or interchangeable, when their joint density function is symmetric and does not depend on their relative order.

**Definition 1** A sequence of random variables  $(X_1, X_2, \dots)$  are called exchangeable if

$$(X_1, X_2, \dots) \triangleq (X_{i_1}, X_{i_2}, \dots) \quad (4.1)$$

for each finite permutation  $i$  of  $\{1, 2, \dots\}$ .

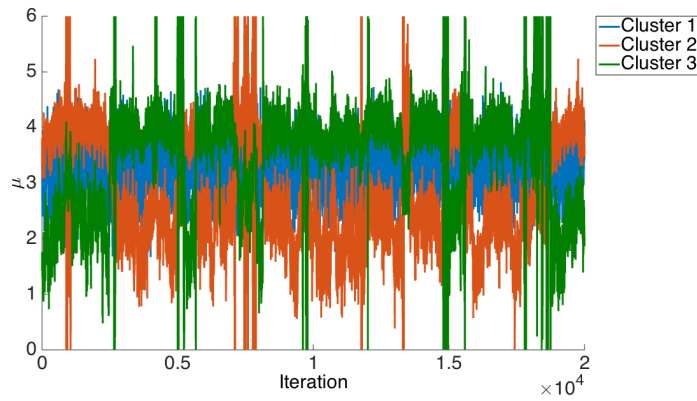
In the context of sampling assignments this means that the probability of assigning observation  $n$  to mixture  $k$  only depends on the current assignments, and not on the order in which the assignments are sampled. Any permutation of the sampling order does not effect the probability function. [Aldous, 1985] As seen in equation 3.11 and 3.18, 3.19, the distribution of observation  $n$  being assigned to mixture  $k$  does not



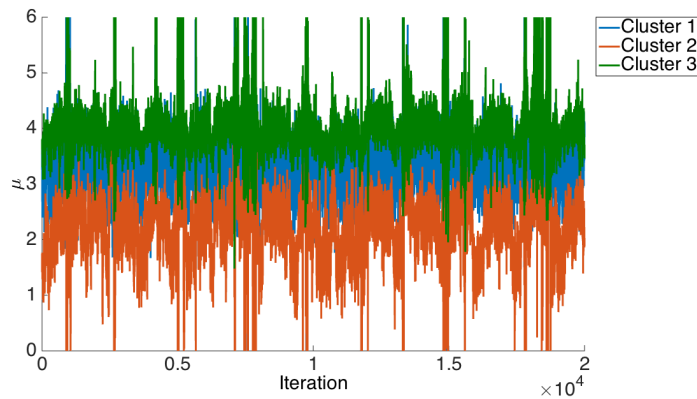
depend on the order of the assignments. Also the likelihood is invariant to the order of the assignments. Hence the exchangeability is fulfilled.

#### 4.1.2 Inference for mixture models

As mentioned above, when drawing inference with MCMC the ergodic average of the sample trace is taken. For the inference to be meaningful the samples must come from a stationary Markov chain. In the case of mixture models and interchangeable mixture labels this is not always the case. Throughout the sampling the labels of two or more mixtures might be interchanged, leading to the samples from the chains being mixed. The interchange of the mixture labels are called *label switching*. An example of this can be seen in figure 4.1 (a).



(a)



(b)

Figure 4.1: Figure (a) shows the occurrence of label switching. It is clearly visible that cluster 1 and cluster 2 are interchanged. In figure (b) relabelling has been applied to the same sample traces and the label switching has been removed.

The result of the label switching is that the ergodic average of the trace becomes meaningless since the average will be over several mixtures. To address this problem and obtain a meaningful estimate one needs to make sure there is no label switching. In the case of label switching *relabelling* has to be applied to obtain a stable sample trace. In figure 4.1 (b), the result of applying relabelling to the samples in 4.1 (a) can be seen. Figure 4.2 shows the sample distributions of the traces from figure 4.1. In figure (a) there is clearly multimodality for cluster 1 and 2. It is clear that for these clusters the symmetry in the posterior is present. Figure (b) shows the distributions after relabelling has been applied. The signs of multimodality are now gone and the estimates will be representative for the mixture parameters.

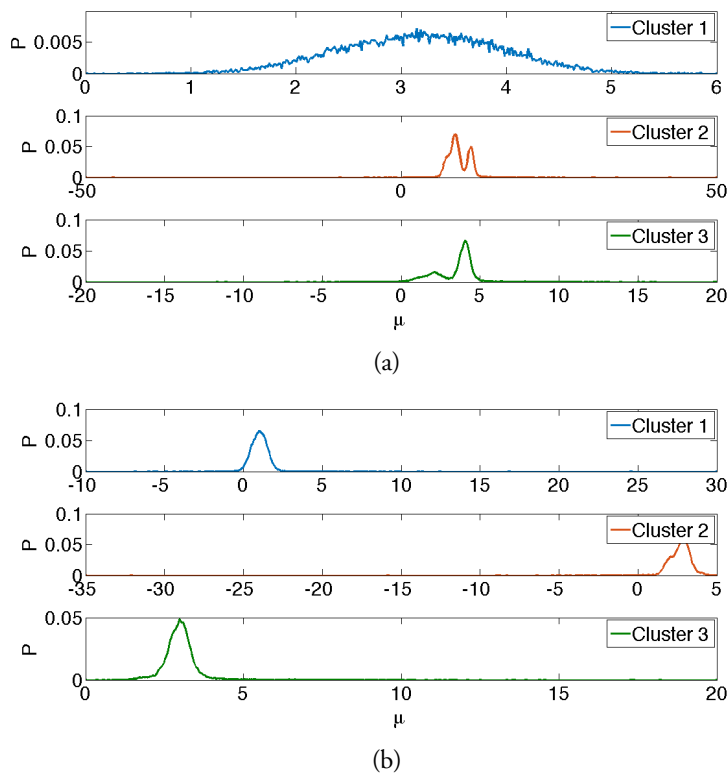


Figure 4.2: The distributions of the respective traces shown in figure 4.1.

### 4.1.3 Efficient sampling

The efficiency of the MCMC sampler depends on how the state space is explored, as mentioned in section 2.4. The samples should be drawn from the whole state space. In that sense the obtain samples covers the whole support of the distribution. For a continuous distribution it is not feasible to cover the whole support, but it can still be extensively explored. In the case of a discrete distribution, as in the assignment

sampling, it is possible to sample from all possible states. Therefore each observation should be assigned to each of the  $K$  assignments throughout the sampling.

One way of looking at the label switching is as an indicator of convergence. For a finite mixture model there are  $K!$  possible permutations of the labels. If all states have not been visited, convergence has not yet been reached. [Jasra et al., 2005] The work at hand has not gone to the extent of expecting  $K!$  permutations, although its effect on the convergence when sampling has been considered.

If convergence is not reached or the label switching is not on par with the desired level, there are methods to enhance label switching and the efficiency of the MCMC sampler. Some of these methods are tempering MCMC [Neal, 1996], Split-Merge MCMC [Jain and Neal, 2004] and evolutionary Monte Carlo [Liang and Wong, 2001]. None of these methods are applied and will therefor not be reviewed here.

## 4.2 Relabelling

As seen in the previous section label switching is an issue when working with MCMC sampling for a mixture model. To solve the issue relabelling needs to be applied. Relabelling can be implemented in two ways, (i) embedded in the sampling algorithm or (ii) as a post-processing step. Existing methods for post-processing has been studied. This approach does not intervene with the sampling in any way and it is easy to compare different methods. For the relabelling to be successful, the non-stationary distribution shall be transformed into a stationary distribution. It is sought for the sample distributions to be unimodal. The methods have different approaches to the permutation of the labels and will find different “optimal” solutions with respect to their respective conditions.

### 4.2.1 Artificial identifiability constraints

An *identifiability constraint* is a constraint on the parameter space  $\Omega$  leading to only one possible permutation. An example of a constraint is to order the means

$$\mu_1 \leq \dots \leq \mu_K. \quad (4.2)$$

Identifiability constraints (IC) were introduced by Diebolt and Robert [1994]. When selecting the constraint it has to be in the context of the model and the observations. With the constraint an artificial constraint is introduced on the parameters. A suitable constraint is one that leads to the density estimates of the parameters being unimodal, or close to unimodal. An unwisely chosen constraint leads to exaggerated skewness and multimodal density estimates [Jasra et al., 2005].

In Diebolt and Robert [1994] the IC was introduced in the sampler, an embedded approach. This approach will not be used, instead the IC is applied to the sampler output as Fruhwirth-Schnatter [2001].

The choice of constraint is highly influential on the inference and should therefore be made with great care. One approach, as suggested by Fruhwirth-Schnatter [2001], is

“if the components of the state specific parameters have some physical meaning, then an expert in the field will have some idea in which way the groups or states differ and might be able to offer such an identifiability constraint.”

In the case of prior expert knowledge this could eventually be introduced in the model to avoid label switching altogether. When working with multivariate problems, finding a suitable constraint becomes increasingly harder and in some cases impossible.

#### 4.2.2 Loss functions

Finding the optimal permutation by some means can be seen in the light of minimising a loss function. This approach was introduced in [Stephens, 2000] and can be seen as a class of methods based on minimising the loss.

The objective is to find the action  $a$  from a set of possible actions  $A$ . The loss function is defined as  $L : A \times \Theta \rightarrow R$ , where  $L(a; \theta)$  is the loss induced by taking action  $a$  when the true value is  $\theta$ .  $\hat{a}$  should be chosen such that the posterior expected loss,

$$R(a) = \mathbb{E}[L(a; \theta)|\mathbf{x}], \quad (4.3)$$

is minimised. For a mixture model the likelihood is indifferent to permutations of  $\theta$ . Therefore a loss function that is invariant to these permutations is considered,

$$L(a; \theta) = \min_v [L_0(a; v(\theta))] \quad (4.4)$$

for some  $L_0 : A \times \Theta \rightarrow R$ . The loss  $R(a)$  can be approximated using the samples  $(\theta^1, \dots, \theta^N)$  as

$$\hat{R}(a) = \min_{v_1, \dots, v_N} \left[ \frac{1}{N} \sum_{n=1}^N L_0(a; v_n(\theta^n)) \right], \quad (4.5)$$

and choosing  $\hat{a}$  as to minimise  $\hat{R}(a)$ . A general algorithm can be constructed as in algorithm 6.

---

#### Algorithm 6 Relabelling: loss function

---

**Require:** initialise  $(v_1, \dots, v_N)$ .

- 1: **while** permutations are performed **do**
  - 2:   choose  $\hat{a}$  to minimise  $\sum_{n=1}^N L_0(a; v_n(\theta^n))$ .
  - 3:   for  $n = (1, \dots, N)$  choose  $v_n$  to minimise  $L_0(a; v_n(\theta^n))$ .
  - 4: **end while**
-

The accuracy and computational complexity of the algorithm depends on the choice of loss function. The algorithm is guaranteed to converge since for every iteration  $\hat{R}$  is decreased and the possible number of permutations is limited. It should be noted that the performance of the algorithm is dependent on the starting point and should therefore be run several times to obtain a result closer to the global optima.

Two loss functions will be reviewed, (i) Equivalence Classes Representatives and (ii) K-means.

### Equivalence Classes Representatives

The Equivalence Classes Representatives (ECR) can be seen as a loss function but deviates slightly from the scheme given in algorithm 6. The ECR method computes the loss with respect to a reference pivot vector. The idea behind the method is based on equivalence classes.  $\mathbf{z}_1$  and  $\mathbf{z}_2$  is equivalent if there exists a transformation  $\tau \in \mathcal{T}_k$  such that  $\mathbf{z}_1 = \tau \mathbf{z}_2$ .  $\Xi_{\mathbf{z}} = \{\tau \mathbf{z} : \tau \in \mathcal{T}_k\}$  denotes the equivalence class of  $\mathbf{z} \in \mathcal{Z}$ , where  $\mathcal{Z}$  is the set of possible assignments. Consider a set  $\mathcal{Z}_0$ , consisting of only one representative from each equivalence class. It can be shown that finding  $\mathcal{Z}_0$  can be reduced to finding a good allocation vector  $\mathbf{z}^*$  and use it as a pivot. With respect to  $\mathbf{z}^*$  the assignments are permuted as to minimise the loss function

$$S(\mathbf{z}^*, \mathbf{z}) = \sum_{n=1}^N \mathbb{1}_{z_n^* \neq z_n}, \quad (4.6)$$

where  $N$  is the number of observations and  $\mathbb{1}$  the indicator function. The difference between the ECR and the general loss function scheme above is that only one iteration is needed since the reference is not updated. [Papastamoulis and Iliopoulos, 2010]

The choice of  $\mathbf{z}^*$  greatly affects the performance of the relabelling. [Papastamoulis and Iliopoulos, 2010] suggest a maximum a posteriori estimate and notes that it is sufficient with a good pivot  $\mathbf{z}^*$  and the “best” is not needed. They show proof of promising qualities for the ECR, the relabelling is not dependent on the mixture distributions parameters and can therefore better handle similarities between distributions. The relatively low computational complexity and the ease of implementation are other qualities of the ECR. It should be noted that the choice of the pivot  $\mathbf{z}^*$  should be done with careful consideration.

### K-means

K-means minimises the distance between observations and their cluster’s mean. In the context of relabelling the observations are the samples and the cluster means are the mixture means [Yao, 2012]. The pseudocode can be seen in algorithm 7. For each iteration the loss decreases and an optima is obtained, with respect to the starting point.

---

**Algorithm 7** Relabelling: K-means

---

```
1: while permutations are performed do
2:   update the mixture means  $\theta_c^z = \frac{1}{N} \sum_{n=1}^N \theta_n^{z_n}$ 
3:   for n = 1:N do
4:     permute  $z_n = \arg \min_z (\theta_n^z - \theta_c^z)^T (\theta_n^z - \theta_c^z)$ 
5:   end for
6: end while
```

---

In comparison to IC with a constraint on the mean, both approaches permute with respect to the mean. In this sense both approaches, in general, result in well separated mixtures. In contrast the K-means is easily applicable to multivariate cases and is not dependent on any introduced constraint. Compared to the ECR, K-means is more costly. K-means will result in more well separated clusters since it tries to find dense clusters and separate the mixtures. The ECR does not base the permutations on the means and can thereby find structures that will be lost with K-means. For instance, mixtures with similar mean but varying variance. The greatest benefit of the K-means approach is that there is no need to construct a reference vector, as for ECR, or constraint, as for IC. This makes it easily applicable with minimal intrusion from the user.

## Chapter 5

# Feature Selection

### 5.1 Introduction

Feature selection is the process of selecting a subset of the feature set, which covers the relatively important features. Feature selection assumes that the features that are not included in the subset are redundant, they do not add any useful information with respect to the model. Some of the benefits with feature selection can be

- improved interpretability,
- lower computational cost,
- improved prediction performance.

Different feature selection methods differ in the sparsity of the obtained feature sets or the measure of relative importance. Due to this, different methods are more or less suitable in various situations. For instance, for DNA sequencing, which is a high dimensional problem, a method resulting in a sparse feature set is preferable.

Figure 5.1 shows an example of clustering in two dimensions. In figure (a) the clusters are only separated in feature  $x$ , hence feature  $y$  does not add any information about how to discriminate between the two clusters. In figure (b) the clusters are well separated in both features, both adding information about how to discriminate between the two clusters. Depending on the method used the obtained feature set might differ. A sparse method would exclude one feature, since only one is needed to distinguish the two clusters.

#### 5.1.1 Supervised & unsupervised learning

Both supervised and unsupervised learning attempt to find structures in the data. There is one crucial difference between the two approaches. In supervised learning there is access to class labels that can be used as references for the classifications, i.e. comparing the data to some given reference. For instance, trying to detect apples and bananas in

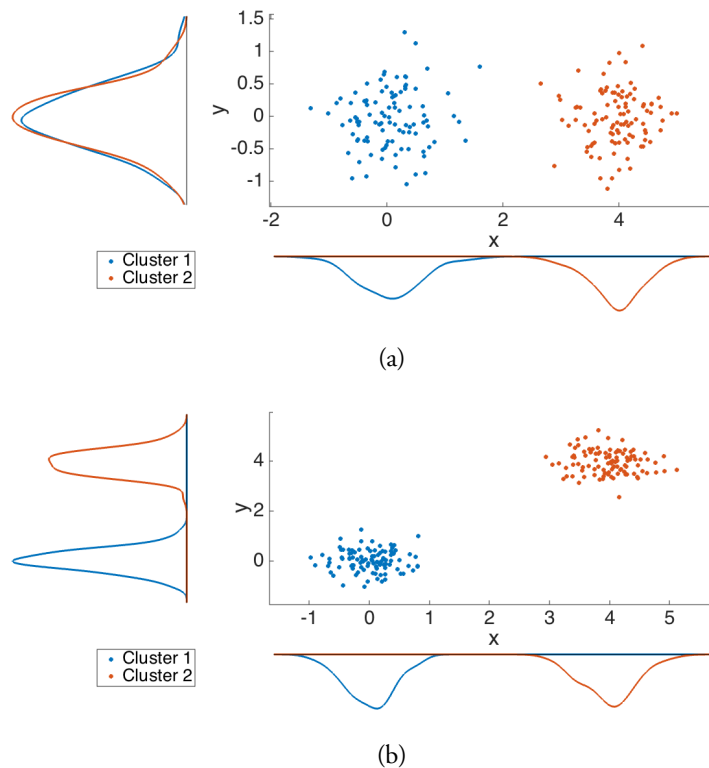


Figure 5.1: Figure (a) shows two clusters well separated in feature  $x$ , but not in feature  $y$ . In figure (b) the clusters are well separated in both feature  $x$  and  $y$ .

images with image analysis. The labels for banana is yellow and a rectangular shape, the label for apple is red and a circular shape. The images are searched for the patterns to classify the fruits according to these shapes and colours. In unsupervised learning the advantage of having labels is not given, instead one tries to cluster the data according to its structure. In unsupervised learning it is attempted to find the hidden structure of the data. Due to this difference, supervised and unsupervised learning are suitable for different applications.

Feature selection is usual in applications in supervised learning, it is a well researched field where the algorithms maximises a function to optimise the predictive accuracy [Fukunaga, 1990; Almuallim and Dietterich, 1991; Cardie, 1993; Kohavi and John, 1997]. Since there are no labels in unsupervised learning it gives the task of feature selection a different flavour. The research on feature selection for unsupervised learning is not yet as extensive as for supervised learning.



### 5.1.2 Approaches of feature selection

Methods of feature selection can be divided into three main approaches, (i) wrapper, (ii) filter and (iii) embedding [Guyon, 2003]. Each approach will be described to give a greater understanding of feature selection.

#### Wrapper

The wrapper approach wraps around the discriminative method. A subset of features are selected and passed on to the discriminative method. A score is obtained for the performance with the selected feature subspace and used to update the feature subspace and repeat the process. The wrapper approach can be summarised with the following steps,

1. Select a feature set.
2. Run the discriminative method.
3. Compute the score and update the feature set.
4. Repeat 1-3 until a criteria is met.

Wrapper methods can be used as a black-box and are not dependent on the discriminative method, although they are run together in order to obtain the optimal feature set.

#### Filter

Filtering can be seen as a pre-processing step or stand alone selection of a feature subspace. A subset of the features are selected and used for the discriminative method.

#### Embedding

Embedded methods are when the feature selection is embedded with the discriminative method. They are often tailored to the specific discriminative method. This approach can be computationally less costly than the other two since the feature selection and the discriminative method are run simultaneously. Therefore the iterative approach of wrapper method is avoided.

## 5.2 Model

As seen above the approach to feature selection can vary. The approach is based on the objective of the feature selection as described in chapter 1. The feature selection method is built around [Kim et al., 2006] and [Niu et al., 2012]. Kim et al. [2006] and Niu et al. [2012] have the similarity of creating a binomial vector for the features where

the features are Bernoulli distributed. Kim et al. [2006] select one subset of the feature set. The non-selected features are modelled as a global distribution, a distribution for all observations. Niu et al. [2012] can be seen as an extension of Kim et al. [2006]. The model is for multi-view clustering, i.e. perform clustering on several overlapping feature subsets. The non-selected features for each view are no longer modelled as a global distribution, they are simply excluded from the model for the specific view. Figure 5.3 illustrates the multi-view model.

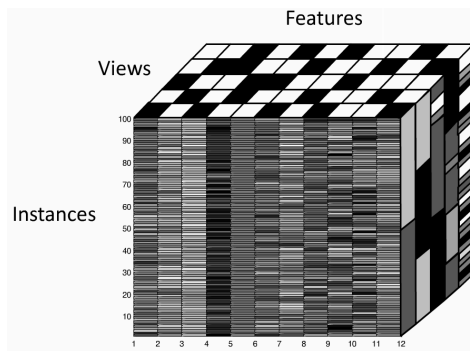


Figure 5.2

Figure 5.3: “Instances, features and views form a cube. The instances and features face is the data. The features and views face indicate the membership of features in views. The instances and views face indicate multiple cluster labelings.” [Niu et al., 2012]

The work at hand stays with the one view model and models the non-selected features as a global distribution. The graphical model can be seen in figure 5.4. If compared to the graphical model of the finite and infinite mixture of Gaussians, figure 3.2 and 3.3, it shows that the addition of feature selection is a simple extension of those models. The extension for the feature selection is the feature vector  $\mathbf{f}$  as well as a *global distribution* with parameters  $\mu_0$  and  $S_0$ .

The idea behind this model is that the informative features are sampled from the mixture distribution while the non-informative features are sampled from the global distribution. The global distribution is a single Gaussian distribution for all observations.

### 5.2.1 Feature distribution

The feature vector  $\mathbf{f}$  is a binary vector where 1 indicates the inclusion of a feature and 0 indicates the exclusion of a feature. The features are independent and identically distributed random variables with distribution

$$f_d \sim \text{Be}(\omega) \quad \forall d \in D. \quad (5.1)$$

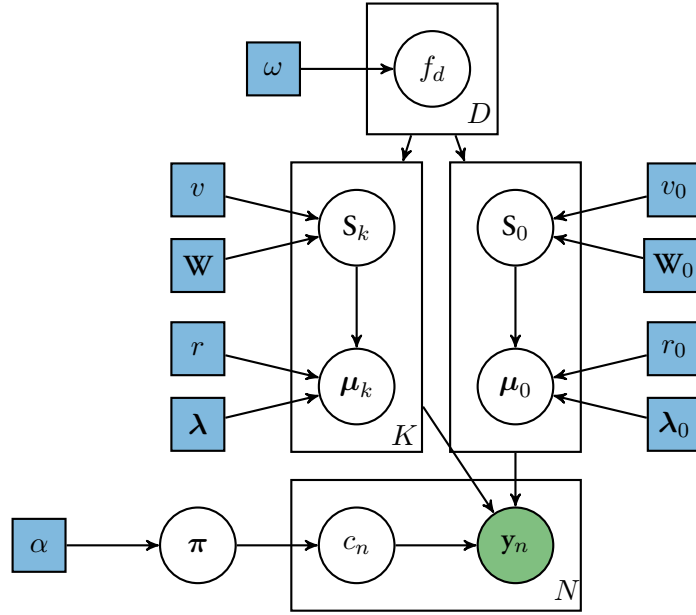


Figure 5.4: Graphical model of the finite Gaussian mixture model with feature selection.  $N$  the number of subjects and  $D$  the number of features. The extension of feature selection can also be added to the infinite mixture model, seen in figure 3.2

The conditional posterior distribution of the features are

$$p(f_d | \mathbf{f}_{-d}, \boldsymbol{\mu}, \mathbf{S}, \boldsymbol{\mu}_0, \mathbf{S}_0, \mathbf{c}, \mathbf{y}) \propto \mathcal{L}(\boldsymbol{\mu}, \mathbf{S}, \boldsymbol{\mu}_0, \mathbf{S}_0, \mathbf{c}, \mathbf{f} | \mathbf{y}) \omega^{n_f} (1 - \omega)^{D - n_f}, \quad (5.2)$$

where  $D$  is the dimension of  $\mathbf{f}$ ,  $n_f$  the dimension of the selected feature subset and  $\mathcal{L}(\cdot)$  is the likelihood

$$\begin{aligned} \mathcal{L}(\boldsymbol{\mu}, \mathbf{S}, \boldsymbol{\mu}_0, \mathbf{S}_0, \mathbf{c}, \mathbf{f} | \mathbf{y}) = & \left[ (2\pi)^{-(D-n_f)N/2} |\mathbf{S}_0|^{N/2} \right. \\ & \left. \exp \left\{ -\frac{1}{2} \sum_{n=1}^N (\mathbf{y}_{n,\mathbf{f}^c} - \boldsymbol{\mu}_0)^T \mathbf{S}_0 (\mathbf{y}_{n,\mathbf{f}^c} - \boldsymbol{\mu}_0) \right\} \right] \\ & \left[ \prod_{k=1}^K (2\pi)^{-n_f n_k / 2} |\mathbf{S}_k|^{n_k / 2} \right. \\ & \left. \exp \left\{ -\frac{1}{2} \sum_{n \in \mathcal{C}_k} (\mathbf{y}_{n,\mathbf{f}} - \boldsymbol{\mu}_k)^T \mathbf{S}_k (\mathbf{y}_{n,\mathbf{f}} - \boldsymbol{\mu}_k) \right\} \right]. \end{aligned}$$

The first part of the likelihood is the global distribution whereas the second part is the mixture distributions.

### 5.2.2 Global distribution

The global distribution is a Gaussian distribution with mean  $\boldsymbol{\mu}_0$  and precision  $\mathbf{S}_0$  for the non-informative features. The prior distributions for the global distribution are

$$\mathbf{S}_0 \sim \mathcal{W}_{D-n_f}(\mathbf{W}_0, v_0) \quad (5.3)$$

$$\boldsymbol{\mu}_0 | \mathbf{S}_0 \sim \mathcal{N}(\boldsymbol{\lambda}_0, (r_0 \mathbf{S}_0)^{-1}), \quad (5.4)$$

where  $D$  is the dimension of  $\mathbf{f}$  and  $n_f$  is the number of included features. The global distribution can be interpreted as the mixture only consisting of one cluster for the excluded features.

The conditional posterior distributions for the mean and precision of the global distribution are

$$\mathbf{S}_0 | \mathbf{W}_0, v_0, \mathbf{y}, \mathbf{f}^c \sim \mathcal{W}_{D-n_f}(\mathbf{W}'_0, v_0 + N), \quad (5.5)$$

where

$$\begin{aligned} \mathbf{W}'_0 = & \left( \mathbf{W}_0^{-1} + \sum_{n=1}^n (\mathbf{y}_{n,\mathbf{f}^c} - \bar{\mathbf{y}}_{\mathbf{f}^c})(\mathbf{y}_{n,\mathbf{f}^c} - \bar{\mathbf{y}}_{\mathbf{f}^c})^T \right. \\ & \left. + \frac{r_0 N}{r_0 + N} (\boldsymbol{\lambda}_0 - \bar{\mathbf{y}}_{\mathbf{f}^c})(\boldsymbol{\lambda}_0 - \bar{\mathbf{y}}_{\mathbf{f}^c})^T \right)^{-1}, \end{aligned} \quad (5.6)$$

and

$$\boldsymbol{\mu}_0 | \boldsymbol{\lambda}_0, r_0, \mathbf{S}_0, \mathbf{Y}, \mathbf{f}^c \sim \mathcal{N}\left(\frac{N\bar{\mathbf{y}}_{\mathbf{f}^c} + r_0\boldsymbol{\lambda}_0}{N + r_0}, ((N + r_0)\mathbf{S}_0)^{-1}\right) \quad (5.7)$$

### 5.2.3 Cluster assignment

The mixture distributions depend on the features included, as seen in figure 5.4. The mixture distributions are of dimension  $n_f$  and hence the cluster assignment will be dependent on the feature vector  $\mathbf{f}$ . Depending on  $\mathbf{f}$ , the dimensionality of the mixture distributions will change throughout the iterations. The posterior is therefore only changed in the sense of which features to include. The updated conditional posteriors are

$$p(c_n = k | \mathbf{y}_n, \boldsymbol{\mu}_k, \mathbf{S}_k, \mathbf{c}_{-n}, \alpha, \mathbf{f}) \propto \frac{n_{k,-n} + \alpha/K}{N + \alpha - 1} \mathcal{N}(\mathbf{y}_{n,\mathbf{f}} | \boldsymbol{\mu}_k, \mathbf{S}_k^{-1}) \quad (5.8)$$

for the finite version and for the infinite version

$$\begin{aligned} p(c_n = k | \mathbf{y}_n, \boldsymbol{\mu}_k, \mathbf{S}_k, \mathbf{c}_{-n}, \alpha, \mathbf{f}) &\propto \mathcal{L}(\boldsymbol{\mu}_k, \mathbf{S}_k, \mathbf{c}, \alpha | \mathbf{y}_{n,\mathbf{f}}) p(c_n = k | \mathbf{c}_{-n}, \alpha) \\ &= \frac{n_{k,-n}}{N + \alpha - 1} \\ &\quad \mathcal{N}(\mathbf{y}_{n,\mathbf{f}} | \boldsymbol{\mu}_k, \mathbf{S}_k^{-1}). \end{aligned} \quad (5.9)$$

$$\begin{aligned} p(c_n = k | \mathbf{y}_n, \boldsymbol{\mu}_k, \mathbf{S}_k, \mathbf{c}_{-n}, \alpha, \mathbf{f}) &\propto \frac{\alpha}{N + \alpha - 1} \\ &\quad \int \mathcal{L}(\boldsymbol{\mu}_k, \mathbf{S}_k, \mathbf{c}, \alpha | \mathbf{y}_{n,\mathbf{f}}) d\text{NW}(\boldsymbol{\lambda}, r, \mathbf{W}, v) \\ &= \frac{\alpha}{N + \alpha - 1} \\ &\quad t_{v-D+1} \left( \mathbf{y}_{n,\mathbf{f}} | \boldsymbol{\lambda}, \frac{r+1}{r(v-D+1)} \mathbf{W}^{-1} \right), \end{aligned} \quad (5.10)$$

where equation 5.9 is when  $n_{j,-i} > 0$  and equation 5.10 is when  $n_{j,-i} = 0$ .

## 5.3 Sampling

With the addition of feature selection the MCMC algorithm needs one modification and two additions. It is needed to (i) modify the sampling of the assignments as well as include the sampling of (ii) features and (iii) parameters of the global distribution. In algorithm 8 the pseudocode for the finite Gaussian mixture model with feature selection is represented, where the above mentioned modifications can be seen if compared to algorithm 4. The feature selection can also be added to the infinite Gaussian mixture model, the pseudocode will not be provided since the modifications are the same as in the finite case.

### 5.3.1 Feature sampling

The sampling of the features are carried out with MH-sampling, see section 2.4.2. A feature  $d$ ,  $d \in \{1, \dots, D\}$ , is selected at random for each iteration. The transition proposal for the selected feature  $d$  is 1 if the current value is 0 and vice versa. Hence the proposal is a simple switch of the binary value. This transition proposal is symmetric and the MH acceptance can be reduced to the ratio between the conditional posterior probability of the current state versus the candidate state. The conditional posterior distribution, equation 5.2, will only differ in the feature vector  $\mathbf{f}$  between the proposal and the current state. This implies that the mixture and global distributions will have different feature sets for the current and candidate state.

For improved performance of the feature selection the MH step is repeated  $h$  times, in accordance with [Kim et al., 2006]. This way the selected features have time to stabilise given the new mixture distributions and hence give a better prediction of which features

that are informative in the current state. The performance difference for different values of  $h$  have not been explored and is instead set according to the findings in [Kim et al., 2006].

---

**Algorithm 8** Finite Gaussian mixture model with feature selection
 

---

**Require:** initialise  $\boldsymbol{\mu}^0, \mathbf{S}^0, \mathbf{c}^0, \mathbf{f}^0$ . Set  $M$  and  $h$ , the number of iterations.

```

1: for  $l = 0 : M - 1$  do
2:   for  $f = 1 : F$  do
3:     Draw a random numbers  $d \in \{1, \dots, D\}$ 
4:     Switch feature  $d$ ,  $0 \rightarrow 1$  or  $1 \rightarrow 0$ 
5:     Metropolis-Hastings acceptance/rejection
6:   end for
7:   for  $n = 1 : N$  do
8:      $c_n^{l+1} \sim p(c_n = k | \mathbf{y}_n, \mathbf{c}_{-n}^l, \boldsymbol{\mu}_k^l, \mathbf{S}_k^l, \mathbf{f})$ 
9:   end for
10:  for  $k = 1 : K$  do
11:     $\mathbf{S}_k^{l+1} \sim p(\mathbf{S}_k | \mathbf{y}_k, \mathbf{c}^{l+1})$ 
12:     $\boldsymbol{\mu}_k^{l+1} \sim p(\boldsymbol{\mu}_k | \mathbf{y}_k, \mathbf{c}^{l+1}, \mathbf{S}_k^{l+1})$ 
13:  end for
14:   $\mathbf{S}_0^{l+1} \sim p(\mathbf{S}_0 | \mathbf{y}, \mathbf{W}_0, v_0)$ 
15:   $\boldsymbol{\mu}_0^{l+1} \sim p(\boldsymbol{\mu}_0 | \mathbf{y}, \mathbf{S}_0^{l+1}, \boldsymbol{\lambda}_0, r_0)$ 
16: end for

```

---

### 5.3.2 Mixture & global distribution sampling

Throughout the sampling the feature subspace is changing. The conditionality of the feature sampling and mixture assignment on the global and mixture distributions gives a situation where it is needed to have samples of varying dimensionality. To handle this the samples from the global and mixture distributions are drawn for all  $D$  features. In this way the sampled parameters for the extra dimensions are at hand and does not have to be sampled later on.

The mixture distributions are sampled in the same manner as described in the case without feature selection, section 3.2.3 and 3.3.2. By sampling for all features the complete mixture distributions are obtain while still having the advantage of excluding noisy features from the cluster assignments.

The global distribution is sampled by Gibbs sampling with the posterior distributions given in equation 5.7 and 5.5.

### 5.3.3 Assignment sampling

The assignment probabilities are affected by the feature vector in the sense of excluding the noisy features, hence modulating the dimension of the mixture distributions. When computing the probability of assigning subject  $n$  to mixture  $k$  only the informative features are included and the differentiation between the mixtures becomes clearer. Hopefully resulting in higher probability of assigning to the “correct” mixture. The assignment sampling scheme follows that in chapter 3 depending on the use of a finite or infinite model. For equations see equation 3.11 or 3.18, 3.19.

## Chapter 6

# Dynamic Causal Models

### 6.1 Introduction

Dynamic causal models (DCM) are used to model the connectivity between brain regions driven by inputs. A DCM models the brain as a linear or non-linear system, depending on the complexity sought. With a DCM one can evaluate a hypothesis of how the brain responds to inputs from the surroundings. The model considers the changes in activity over time depending on the inputs. From an explanatory point of view this means that, for instance, learning can be modelled in the form of plasticity<sup>1</sup>. It is a causal model where the states are perturbed by an input, i.e. the input is causing the responses of the system.

There are two ways in which input can evoke responses, (i) direct changes in the state variables and (ii) changes to the effective connectivity or interactions between states. The first type of input could be a visual or auditory stimulus. Something that affects the states directly, resulting in neuronal activity. The second type of input could come from attention modulation of the connections or changes corresponding to plasticity.

The objective is to estimate the connectivity parameters by perturbing the system and measuring the response. In practice this can be used to get better knowledge about the dynamics of the brain, what perturbs the system and which responses does it lead to. Questions like, are there differences in the dynamics of the brain between gender or between healthy subjects and subjects diagnosed with a psychiatric disorder, can be answered. Simply put, it is possible to compare the neuronal connections of subjects and perform analysis to characterise inter-subject differences.

---

<sup>1</sup> Plasticity refers to changes in neural pathways and synapses which are due to changes in behaviour, environment and neural processes, as well as changes resulting from bodily injury. [Pascual-Leone et al., 2011]



## 6.2 Dynamic causal models

The dynamic causal model was first presented in [Friston et al., 2003] and will be described in this section. The dynamic causal model links the connections of neuronal states with a measured signal of brain activity. This measured signal could be from electroencephalography (EEG) or functional magnetic resonance imaging (fMRI), for example. Depending on the source of the signal some parts of the model have to be modified, the model described will be applicable in the context of fMRI measurements. The model consists of a system with multiple inputs and outputs. The inputs, as described above, could be a stimulus and correspond to the experimental design. The outputs correspond to the observed blood-oxygen-level dependent (BOLD) signal.

Figure 6.1 shows an example of the system structure.  $x_1$ ,  $x_2$ ,  $x_3$  are the neuronal states with bidirectional connections, capturing the direction of the influence between the states.  $u_1$  and  $u_2$  are the input to the system.  $u_1$  directly affects state  $x_1$  whereas  $u_2$  has a modulatory affect on the connection from  $x_1$  to  $x_3$ . The input  $u_1$  activates the system which will result in a BOLD signal  $y_1$ ,  $y_2$ ,  $y_3$  corresponding to the three neuronal states. The neuronal states are connected to the BOLD signal via a forward model, which will be described below.

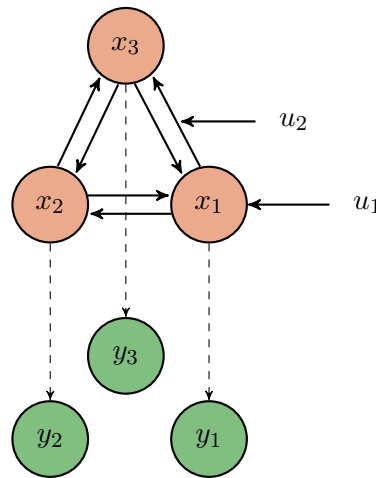


Figure 6.1: An illustration of the DCM. The neuronal states  $x_1$ ,  $x_2$ ,  $x_3$  are affected by input  $u_1$ ,  $u_2$  which results in a measurable signal  $y_1$ ,  $y_2$ ,  $y_3$ .

For each region a measured output ( $y_1$ ,  $y_2$ ,  $y_3$ ) is obtain in the form of a time series. Each region has five state variables, four related to variables of a haemodynamic model and one variable corresponding to the neuronal activity as a function of the neuronal states, ( $x_1$ ,  $x_2$ ,  $x_3$ ). The model can be divided into three parts, (i) neural state equations, (ii) haemodynamic model and (iii) BOLD signal model.

### 6.2.1 Neural state equations

Using figure 6.1 as a basis for the explanation, the neuronal states are  $x = (x_1, x_2, x_3)$ . The dynamics of these states can be expressed as

$$\dot{x} = F(x, u, \theta), \quad (6.1)$$

where  $F$  is a nonlinear function of the neuronal states, the input  $u$  and model parameters  $\theta$ . The function  $F$  can be approximated using Taylor expansion. The obtained bilinear form of the neural state equations for DCMs are

$$\begin{aligned} \dot{x} &\approx (A + \sum_{j=1}^J u_j B^j)x + Cu \\ A &= \frac{\partial F}{\partial x} = \frac{\partial \dot{x}}{\partial x} \\ B^j &= \frac{\partial^2 F}{\partial x \partial u_j} = \frac{\partial}{\partial u_j} \frac{\partial \dot{x}}{\partial x} \\ C &= \frac{\partial F}{\partial u}. \end{aligned} \quad (6.2)$$

It is referred to as bilinear due to the second order derivative  $B^j$ . A linear version can be obtained by satisfying with a lower order approximation. The linear form only consists of the matrices  $A$  and  $C$ . There is also a nonlinear form where the states can directly modulate the connections. Here the DCM will be explained using the bilinear form.

Matrix  $A$  represents the first order interactions between the states. A change in activity will influence the other states over the neuronal network, inducing a response of  $\frac{\partial \dot{x}}{\partial x}$ . The  $B^j$  matrix is the change to the connections induced by input  $u_j$ . It indicates the modulatory effects of input  $u_j$ . Matrix  $C$  is the strength of the inputs that directly affect the neuronal activity. Together the matrices,  $\theta = \{A, B^1, \dots, B^J, C\}$ , describe the connection of brain regions and inputs at a neuronal level.

### 6.2.2 Haemodynamic model

To connect the neural state equations with the BOLD model a haemodynamic model that describes the connection between activations of the neuronal states and the blood-oxygenation-level is needed. This is done with a haemodynamic forward model based on the work by [Buxton et al., 1998; Friston et al., 2000].

The haemodynamic state equations can be divided into two parts, (i) describing the neural activity and regional cerebral blood flow (rCBF) and (ii) the connection between BOLD signal and rCBF-induced changes of blood volume and deoxyhaemoglobin content. The link to rCBF models a dampened oscillator with linear differential equations [Friston et al., 2000]. The connection between BOLD signal and rCBF is described as the ‘‘Balloon model’’ [Buxton et al., 1998]. The Balloon model describes the

changes in blood volume  $v$  and deoxyhaemoglobin (dHB) content  $q$  with a non-linear dependency on the BOLD signal. The haemodynamic state equations are

$$\begin{aligned}\dot{s} &= x - \kappa s - \gamma(f - 1), \\ \dot{f} &= s, \\ \tau\dot{v} &= f_{in} - v^{1/\alpha}, \\ \tau\dot{q} &= f_{in}E(f_{in}, E_0)/E_0 - v^{1/\alpha}q/v.\end{aligned}\tag{6.3}$$

Here  $\dot{s}$  is the change in the vasodilatory signal and  $\dot{f}$  the change in flow induction (rCBF).  $\kappa$  is the rate constants of signal decay and  $\gamma$  the feedback regulation.

The Balloon model is comprised of the last two equations,  $\tau\dot{v}$  and  $\tau\dot{q}$ , where the first one is change in volume and the second one change in dHB. The outflow is related to the volume through Grubb's exponent  $\alpha$  as  $f_{out}(v) = v^{1/\alpha}$  [Grubb et al., 1974]. In brief, a neuronal activity  $x$  leads to an increase in a vasodilatory signal  $s$ . The inflow  $f$  responds to the signal  $s$  with associated changes in blood volume  $v$  and dHB content  $q$ .

### 6.2.3 BOLD signal model

In [Buxton et al., 1998] the BOLD signal at rest is modelled as

$$S_0 = (1 - V_0)S_E + V_0S_I.\tag{6.4}$$

The equation comprises a volume weighted sum of the extra- and intravascular signals,  $S_E$  and  $S_I$ . The interest lays in the changes in the BOLD signal during activation  $\Delta S$  in relation to the resting state  $S_0$ . This is expressed as a non-linear function of the volume and dHB. Here, the best model found by [Stephan et al., 2007] is followed.

$$\begin{aligned}\lambda(q, v) &= \frac{\Delta S}{S_0} \\ &\approx V_0[k_1(1 - q) + k_2(1 - \frac{q}{v} + k_3(1 - v))], \\ k_1 &= 4.3\vartheta_0 E_0 TE, \\ k_2 &= \epsilon r_0 E_0 TE, \\ k_3 &= 1 - \epsilon.\end{aligned}\tag{6.5}$$

$v$  and  $q$  are the venous blood volume and dHB content,  $E_0$  is the oxygen extraction fraction at rest.  $\vartheta_0$  is the frequency offset at the outer surface of the magnetised vessel for fully deoxygenated blood,  $TE$  is the echo time and  $r_0$  is the slope of the relation between the intravascular relaxation rate and oxygen saturation.  $E_0$ ,  $\vartheta_0$ ,  $TE$ ,  $r_0$  are all fixed parameter related to the experiment settings. In the literature there is an

uncertainty about the value of  $\epsilon$ , [Buxton et al., 1998; Obata et al., 2004; Silvennoinen et al., 2006; Lu and van Zijl, 2005]. To acknowledge this uncertainty  $\epsilon$  is not given a fixed value, instead the parameter is given a suitable prior density.

### 6.2.4 Bayesian framework and priors

DCMs are complex and may need a large number of free parameters since they are not restricted to linear or instantaneous systems. This makes them more plausible from a biological point of view. This leads to the need to introduce constraints on the parameters for the estimates. [Friston et al., 2003] This can easily be done in a Bayesian framework, where the constraints are introduced via the priors. Within a Bayesian framework the resulting estimates will be the posterior probability distributions of the parameters. For further information on the priors and implementation, see [Friston et al., 2000] and SPM8.

## 6.3 Generative embedding

Generative embedding, as described in [Brodersen et al., 2011], combines a generative model with a discriminative or generative method. Here a generative method for clustering will be used. A signal  $Y$  is mapped to a different feature set  $X$ . The feature set of  $X$  is more suitable for the generative method, improving performance, as well as interpretation.

The method can be described in four steps as seen in figure 6.2. A generative model maps the data from  $Y \mapsto M_{\Theta}$ . The mapping creates interpretable features which explains the mechanics behind the signal,  $Y$ . The model feature space  $M_{\Theta}$  is mapped to a feature space suitable for the generative method.  $M_{\Theta} \mapsto X$ , where  $X$  is the generative score space. The generative method is applied on the feature set  $X$  and the obtained result can be interpreted.

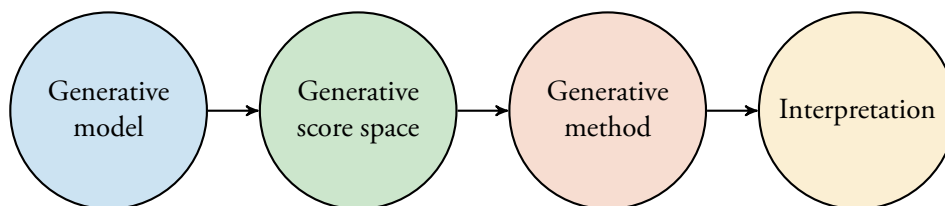


Figure 6.2: Structure of generative embedding.

### 6.3.1 Generative model

The generative model used is the DCM described in this chapter. The DCM explains the underlying mechanisms behind the BOLD signal. The DMC lets one infer on the

connectivity parameters, constructing the generative score space.

Discriminating on connectivity parameters, instead of high-dimensional fMRI time series, improves the neurobiological interpretability of the obtained result. The neuronal system are of interest in psychiatric spectrum diseases, such as schizophrenia [Stephan et al., 2009] and depression [Castren, 2005], validating the use of DCM as a generative model. With a DCM one can infer on the neuronal system from subject specific data. Applying the generative method to the inferred output from several subjects lets us analyse if the model can describe differences between, for example, diagnosed and healthy subjects.

The DCM projects the obtained BOLD signal  $Y$  to the posterior probability distribution  $p(\theta | y, m)$ ,  $y \in Y$ ,  $m \in M_{\Theta}$ . The model architecture  $m$  defines the model of the neuronal system. The model describes the brain regions of interest and their interconnections and response to input. With a given model  $m$  the connectivity parameters subject by subject is inferred, mapping the posterior distribution to the real valued  $X$ .

### 6.3.2 Generative method

The generative method will be the Gaussian mixture model with feature selection, described in chapter 5, performing clustering on the connectivity parameters inferred from the DCM.

With a feature selected clustered connectivity one can infer on groups of subjects with similar neuronal systems. It is also possible to infer on the relative importance of the connections with respect to discriminating the clusters. The addition of feature selection means more statistics will be available when interpreting the result of the generative method.

### 6.3.3 Unified framework

A method to unify the generative embedding in the context of DCMs for multi-subject analysis has been developed at the Translational Neuromodeling Unit, University of Zurich & ETH Zurich [Raman and Stephan]. The unified framework incorporates the DCM with the clustering on the connectivity parameters using a Gaussian mixture model. This method has been extended with the feature selection described in chapter 5. Figure 6.3 shows a graphical representation of the unified framework with the feature selected clustered connectivity for DCMs.

Here  $\theta_{d_n}$  is the subject specific connectivity parameters,  $\Lambda_n$  a subject specific Gaussian distributed noise parameter,  $\theta_{h_n}$  a subject specific haemodynamic parameter and  $\theta_q$  is model parameters for the DCM. The left part of the figure are the feature selection and the Gaussian mixture model. The parts marked with red are the additions of feature selection, being the additions made to the previous work.

The advantage of the unified framework is that it incorporates the generative model and the generative method in one unified model. By doing so it is possible to draw inference on the cluster assignments, feature probabilities and connectivity parameters for multiple subjects simultaneously.

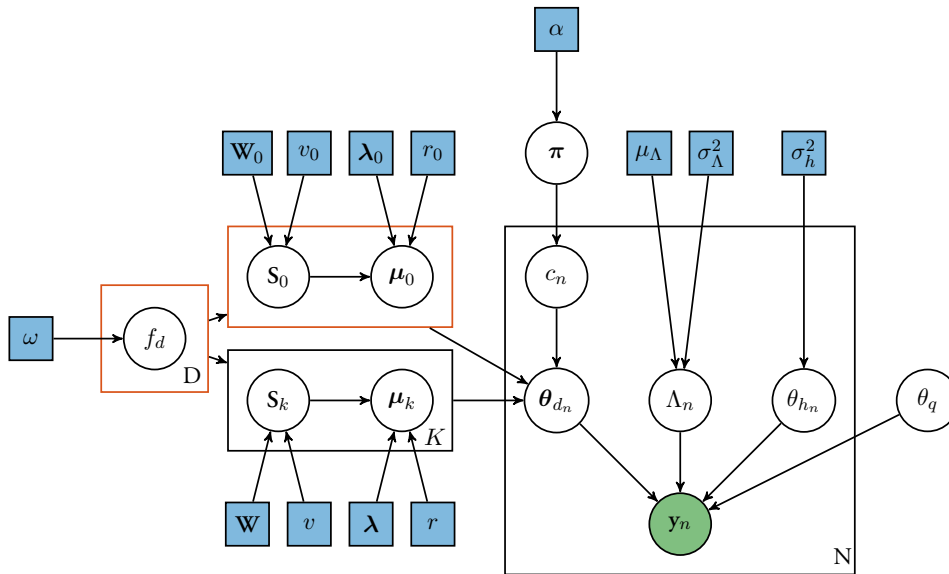


Figure 6.3: A graphical model of the unified model for DCMs with feature selected clustering.

# Chapter 7

## Results

The experiments are designed to (i) investigate the clustering performance of the different models, (ii) evaluate the accuracy of the feature selection and (iii) compare the relabelling algorithms. For the evaluation, both synthetic data sets as well as a real data set are used. The synthetic data set allows for a sensitivity analysis with respect to the number of noisy features to be performed. It will also be possible to evaluate the feature selection and its accuracy, since the ground truth is known. This chapter presents the datasets, the evaluation methods as well as the obtained results.

### 7.1 Evaluation measures

For evaluation of the mixture assignments two measures are used, (i) normalised mutual information and (ii) balanced purity. Both measures give a quantitative estimation of how accurate the assignments of the objects are. They complement each other since they penalise for faults differently. Balanced purity only penalise for impurities in the cluster, whereas normalised mutual information penalise for uncertainty in describing the classes with the obtained clusters.

#### 7.1.1 Normalised mutual information

Mutual information comes from information theory and lets us interpret the performance of the clustering in view of information theory. It measures the relative entropy between the joint distribution  $p(x, y)$  and the product distribution  $p(x)p(y)$ . The mutual information is expressed as

$$I(X; Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}. \quad (7.1)$$

The more information the two variables share the higher the mutual information will be. In the case of the variables being independent,  $p(x, y) = p(x)p(y)$ , the mutual

information will be zero. [Cover and Thomas, 2006]

To obtain a comparable measure, the mutual information is normalised so that the obtained score is in the range of  $[0, 1]$ . There are several ways of normalising the mutual information. The chosen procedure is described in [Strehl and Ghosh, 2002],

$$\text{NMI} = \frac{I(X; Y)}{\sqrt{H(X)H(Y)}}. \quad (7.2)$$

### 7.1.2 Balanced purity

Purity measures the purity of each cluster. The more homogenous the clusters are, with respect to the classes, the higher the score. It takes the cardinalities of the dominant class for each cluster and sums them up. The sum is divided with the number of objects as expressed in the following equation

$$\text{purity}(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{k=1}^K \max_j |\mathbf{x}_k \cap \mathbf{y}_j|, \quad (7.3)$$

where  $N$  is the number of objects and  $K$  the number of clusters.

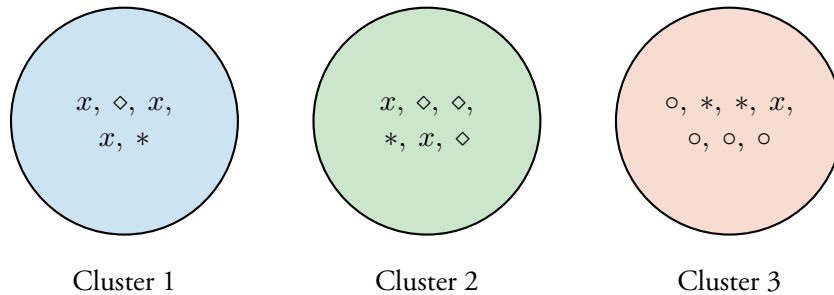


Figure 7.1: Purity as a measure of clustering performance. The cardinality of the dominant class in each cluster are 3 ( $x$  in cluster 1), 3 ( $\diamond$  in cluster 2) and 4 ( $\circ$  in cluster 3). The purity score equals  $\frac{1}{18}(3+3+4) = 0.5556$ .

Purity is a simple and easily interpretable measure of the clustering performance. Good results give a score close to 1 while bad results give a score close to 0. Purity may be a misleading measure when the data set is not well balanced. For instance, a data set with two classes where one is five times as big as the other. To correct this *balanced purity* as described in [Broderson et al., 2013] will be used,

$$\text{BP}(\mathbf{x}, \mathbf{y}) = \left(1 - \frac{1}{K}\right) \left(\frac{\text{purity}(\mathbf{x}, \mathbf{y}) - \epsilon}{1 - \epsilon}\right) + \frac{1}{K}. \quad (7.4)$$

Here  $\epsilon$  is the fraction of objects within the largest class and  $K$  the number of classes. The balanced purity (BP) can be interpreted as the probability of assigning an object



to a cluster where the objects class is dominating. The same example as in figure 7.1 would result in a BP score of 0.5050, which is slightly lower due to the imbalanced data set.

## 7.2 Synthetic data set

The synthetic data set has two relevant features and up to ten irrelevant features. The data set with all twelve features is visualised in figure A.1. It consists of three classes with 50 observations each, 150 observations in total. In the two relevant features the clusters are clearly separated, with mean  $(-10, -10)$ ,  $(0, 0)$  and  $(10, 10)$ . For the irrelevant features,  $(3, \dots, 12)$ , the classes have the same mean and variance. With this data set the performance of the feature selection will be evaluated. The dimension of the data set is varied by changing the number of included irrelevant features. By increasing the number of irrelevant features, a result of the performance with respect to the proportion of noisy features will be obtained. The sampler ran for 120,000 iterations, which of 40,000 was burn-in. The data set was normalised to remove any effects of scale differences between features.

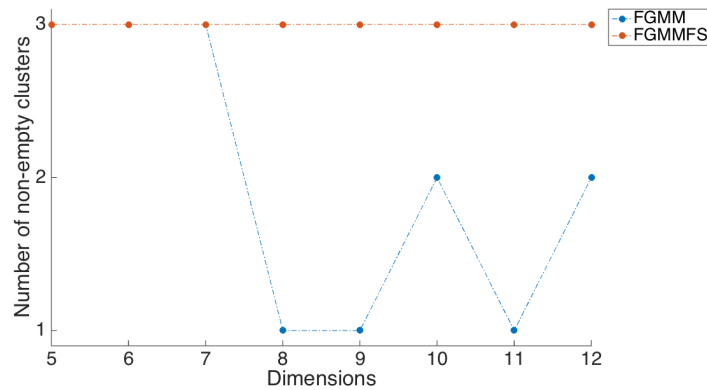
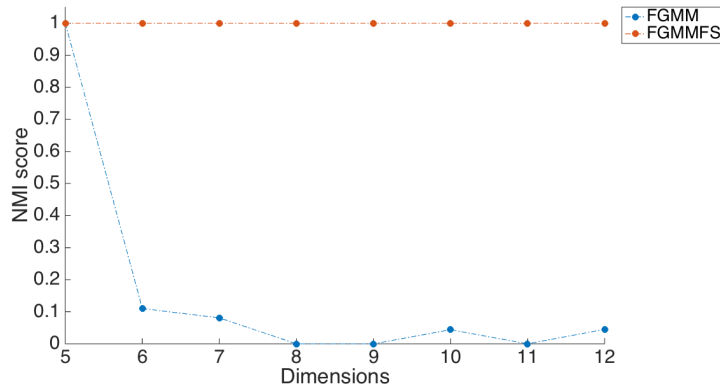


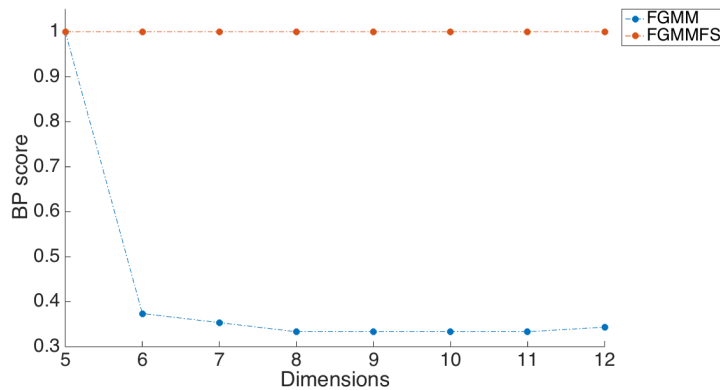
Figure 7.2: The number of non-empty clusters inferred using FGMM and FGMMFS for the synthetic data set with two relevant features. The dimensionality is indicated on the x axis.

Figure 7.2 shows the number of non-empty clusters obtained using the finite Gaussian mixture model with and without feature selection (FGMMFS, FGMM) for varying dimensionality of the data set. The FGMM correctly finds three clusters for the lower dimensional data sets. When the number of noisy features is increased it starts to struggle and the number of non-empty clusters decrease. The FGMMFS on the other hand manages to correctly find three non-empty clusters for up to ten irrelevant features included. Even though the number of non-empty clusters are correctly inferred up to 7 dimensions with the FGMM, the clustering performance is not satisfying. In figure 7.3 the NMI and BP scores are presented for the two models. Both the NMI and

BP score see a large drop for the FGMM when the dimensionality is increased from 5 dimensions. The scores for a dimensionality of 6 – 12 are close to or 0 for the NMI and close to or  $1/3$  for BP. Hence the clustering for these dimensionalities with FGMM does not say anything about the underlying structure of the data. The NMI and BP scores for using FGMMFS are equal to one for all dimensionalities, the clusters accurately represents the classes.



(a) NMI score.



(b) BP score.

Figure 7.3: NMI and BP score for FGMM and FGMMFS for varying dimensionality of synthetic data set.

The effect of including feature selection is very clear. The clustering performance increases drastically when noisy features are present. The FGMM gives satisfying results for a ratio of irrelevant to relevant features up to 1.5. With feature selection the ratio was increased to, at least, 5.

Figure 7.4 shows the probability of each feature being included. For all dimensionalities, the feature selection accurately detected the relevant features with a probability equal to 1. For the 5 – 9 dimensional data sets there are some additional features with a

probability above 0.5. In the 11 and 12 dimensional cases all irrelevant features have a probability close to 0.

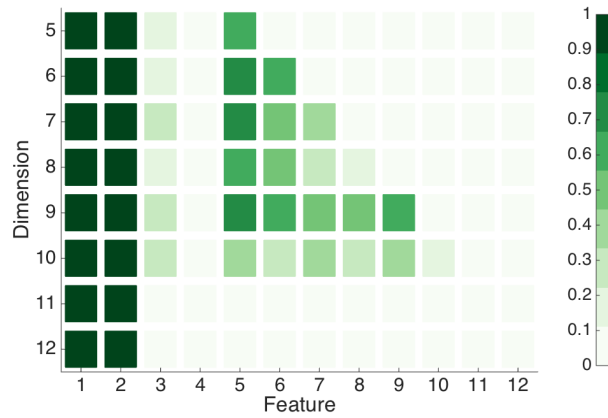


Figure 7.4: The colour indicates the probability of including the feature. The y axis indicates the dimensionality of the data set and the x axis which feature.

Figure 7.5 shows the trace of the number of included features for the 12 dimensional data set. Throughout the sampling the number of individual features varies from two to twelve. For the majority of the samples the number of included features is two, 94 % of the samples.

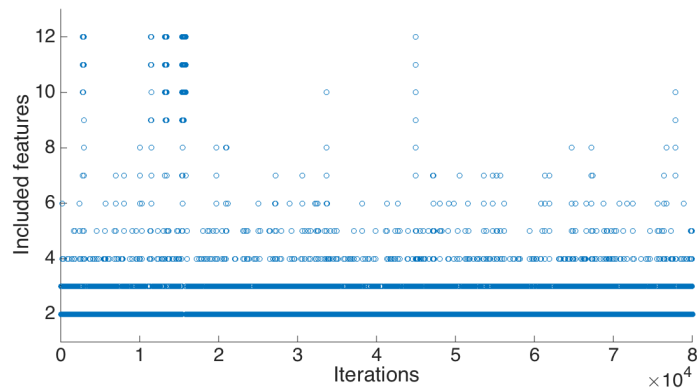
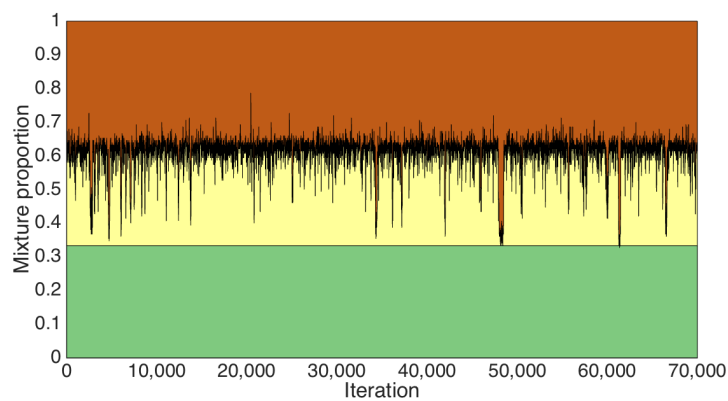


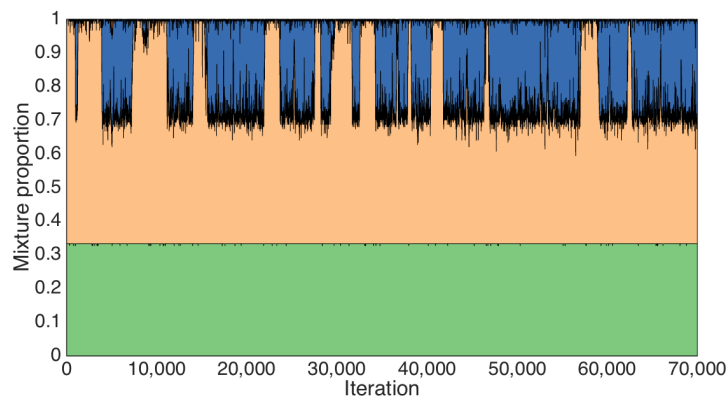
Figure 7.5: The number of included feature for each iteration with a twelve dimensional data set.

### 7.3 Iris flower data set

The Iris flower data set was introduced by Sir Ronald Fisher in 1936, [Fisher, 1936]. The data set consists of observations of three species of Iris, *Iris setosa*, *Iris virginica* and *Iris versicolor*. It consists of 150 observations in total, 50 observations for each species. Each observation has four features, corresponding to the length and the width of the sepals<sup>1</sup> and petals<sup>2</sup>. A scatterplot of the data set can be found in figure A.2. The Iris flower data set can be found at [Bache and Lichman]. *Iris setosa* is well separated from the other two species, which are partially mixed in all four features.



(a) Mixture proportion for FGMM.



(b) Mixture proportion for IGMM.

Figure 7.6: The trace of the mixture proportions for the FGMM and IGMM.

The simulations were run for 110,000 iterations, which of 30,000 was burn-in. The data set was normalised. Figure 7.6 shows the trace of the mixture proportion for the

<sup>1</sup>any of the separate parts of the calyx of a flower

<sup>2</sup>one of the often coloured segments of the corolla of a flower.

FGMM and infinite Gaussian mixture model (IGMM). With the FGMM the mixture proportions are relatively stable throughout the iterations. For the IGMM the trace is not as stable. It jumps between two and three major clusters, sometimes combining the overlapping classes *Iris virginica* and *Iris versicolor*. The mixture proportion for the infinite Gaussian mixture model with feature selection (IGMMFS) shows similar behaviour as the IGMM. The infinite models are more prone to change the number of clusters to which the majority of the observations are assigned. This behaviour persists when the number of iterations are increased, which is tested for up to 300,000 iterations.

From figure A.2 it is clear that all four features are relevant for the clustering. The result from using the FGMMFS and IGMMFS confirms this, as seen in figure 7.7. The probability of including each feature equaled 1 for the FGMMFS. For the IGMMFS feature 1, 3 and 4 had a probability of 1 and feature 2 had a probability of 0.9954.

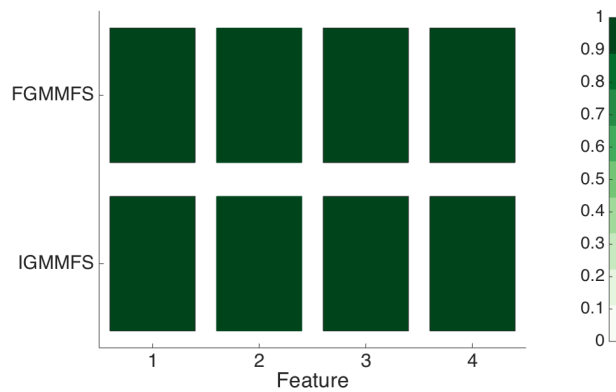


Figure 7.7: Probability of including feature 1 – 4 from Iris data set using FGMMFS and IGMMFS. The colour indicates the probability of including the feature.

Method	Number of mixtures	NMI	BP
FGMM	fixed at 3	0.8997	0.9667
FGMMFS	fixed at 3	0.8977	0.9667
IGMM	3	0.8622	0.9467
IGMMFS	3	0.8315	0.9267

Table 7.1: NMI & BP for the obtained cluster assignments using FGMM, FGMMFS, IGMM & IGMMFS on the Iris data set. For the infinite model the inferred number of clusters are presented.

The accuracy of the clustering can be seen in table 7.1. Both infinite models accurately found the three clusters. All four methods obtain high scores in both NMI and BP. The

incorrectly labeled observations are among the observations on the boarder between Iris virginica and Iris versicolor. There is a slight difference between the finite and infinite models in the score. The difference could be explained by the cluster assignment not being as stable for the infinite models as for the finite models.

The inferred mean of each cluster from the four models are presented in table 7.2. As a reference the class means from the Iris data set are presented. The means for cluster 1 are close to the class mean for all models. The means for cluster 2 and 3 differ slightly between the class reference, the finite and infinite models. The finite models estimate the means slightly better than the infinite models. This can be explained by figure 7.6. It is clear that cluster 2 and 3 are seen as one during sequences of the sampling for the infinite models. For the infinite models the estimated means for cluster 2 and 3 are closer to each other and the variance are higher than for the finite models.

	Cluster	Feature 1	Feature 2	Feature 3	Feature 4
Classes	1	5.0060	3.4180	1.4640	0.2440
	2	5.9360	2.7700	4.2600	1.3260
	3	6.5880	2.9740	5.5520	2.0260
FGMM	1	5.0074 ± 0.0492	3.4172 ± 0.0530	1.4685 ± 0.0300	0.2459 ± 0.0167
	2	5.9388 ± 0.1275	2.7950 ± 0.0642	4.2087 ± 0.1174	1.2994 ± 0.0444
	3	6.4996 ± 0.1070	2.9324 ± 0.0500	5.4031 ± 0.1444	1.9454 ± 0.0776
FGMMFS	1	5.0078 ± 0.0492	3.4175 ± 0.0527	1.4686 ± 0.0301	0.2459 ± 0.0167
	2	5.9332 ± 0.1041	2.7905 ± 0.0553	4.2064 ± 0.0919	1.2984 ± 0.0348
	3	6.5074 ± 0.1000	2.9347 ± 0.0488	5.4175 ± 0.1245	1.9528 ± 0.0671
IGMM	1	5.0078 ± 0.0494	3.4176 ± 0.0526	1.4686 ± 0.0302	0.2457 ± 0.0167
	2	5.9853 ± 0.3753	2.8320 ± 0.2101	4.2610 ± 0.6935	1.3240 ± 0.2920
	3	6.4342 ± 0.1440	2.9158 ± 0.0536	5.2672 ± 0.2545	1.8721 ± 0.1366
IGMMFS	1	5.0076 ± 0.0492	3.4173 ± 0.0528	1.4687 ± 0.0302	0.2460 ± 0.0168
	2	6.0420 ± 0.4903	2.8481 ± 0.2382	4.3476 ± 0.8664	1.3587 ± 0.3572
	3	6.4080 ± 0.1511	2.9104 ± 0.0542	5.2167 ± 0.2722	1.8474 ± 0.1461

Table 7.2: Cluster means with standard deviation for the FGMM, FGMMFS, IGMM and IGMMFS method for 80,000 iterations on the Iris data set.

## 7.4 Label switching

To compare the three relabelling algorithms, introduced in chapter 4, the output from FGMM on the Iris data set is permuted. By permuting the output from the FGMM the ground truth is known and the performance of the methods can easily be compared. Five permutations were made and the resulting trace of the mean of feature 1 can be seen in figure 7.8.

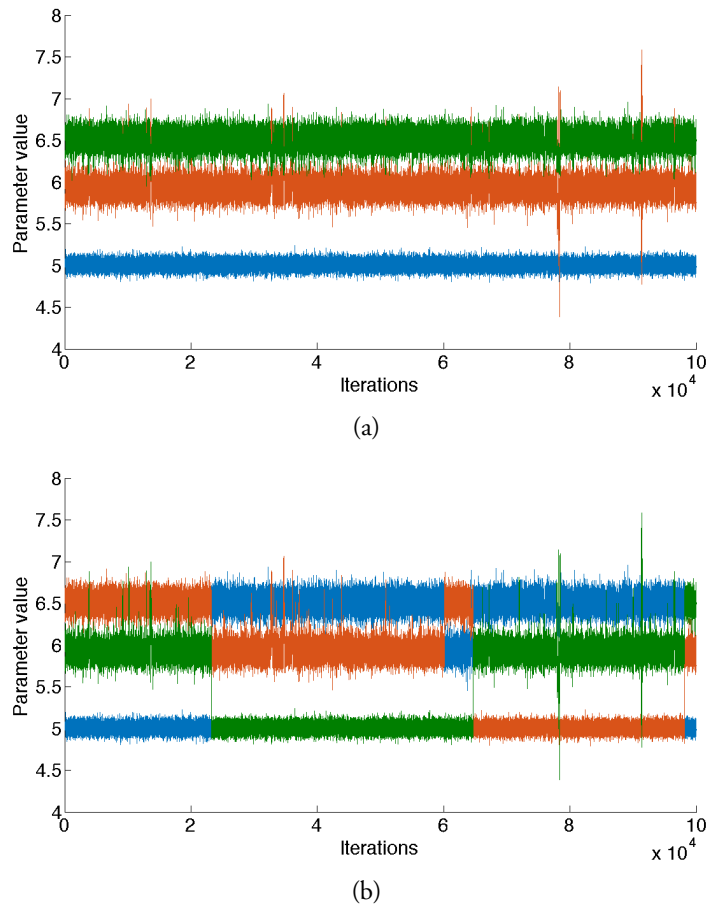


Figure 7.8: Trace for mean of feature 1 for the three clusters. Figure (a) is the trace before permutation and figure (b) after permutation.

For the IC the mean was chosen as the constraint. The mean of both feature 1 and 4 individually were used as constraints, providing a test of the robustness of the IC. As seen in figure A.2, the classes overlap in feature 1 and are almost separated in feature 4. For ECR several reference assignment vectors were compared, all resulting in similar results. The reference used for the presented results is the mode of the permuted trace.

Table 7.3 shows the NMI and BP score for the inferred cluster assignments after relabelling. The assignments were compared to the inferred assignments from the original un-permuted output from FGMM. All the methods perform well with respect to obtaining the correct cluster assignment for each observation. K-means sticks out by being the only method not getting all observations correctly labeled, mislabelling one observation.

Method	NMI	BP
IC(1)	1	1
IC(4)	1	1
K-means	0.9701	0.9930
ECR	1	1

Table 7.3: NMI & BP score for the inferred assignments after relabelling with respect to ground truth.

For the relabelling the removal of the multimodality is also of interest. Figure 7.9-7.12 shows the trace distributions after relabelling. There is a clear difference between the methods. K-means manages to remove almost all effects of multimodality satisfyingly. There are some tendencies of a second mode, especially in cluster 2, but overall the result must be seen as satisfying. The ECR did not manage to remove the multimodality, which is clearly visible in figure 7.10. Similar results were obtained when using different reference assignment vectors. The result obtained with ECR is at subpar with the other methods. Figure 7.11 shows the distributions when using IC with a constraint on feature 1 and figure 7.12 with a constraint on feature 4. Both results in satisfying distributions. There is a slightly stronger tendency of multimodality when the constraint is on feature 1 compared to 4. This is a result of the classes overlapping more in feature 1 than feature 4. Comparing IC with constraint on feature 4 with K-means, the results are similar. It should be noted that in this case the constraint was chosen with prior knowledge of the structure of the data.

The inferred means from the trace and its standard deviation for each method is presented in table 7.4. The result from IC(4) and K-means are on par. IC(1) has slightly higher variance whereas ECR has distinctively higher variance than the other methods. The increased variance for ECR is a result of the multimodality.



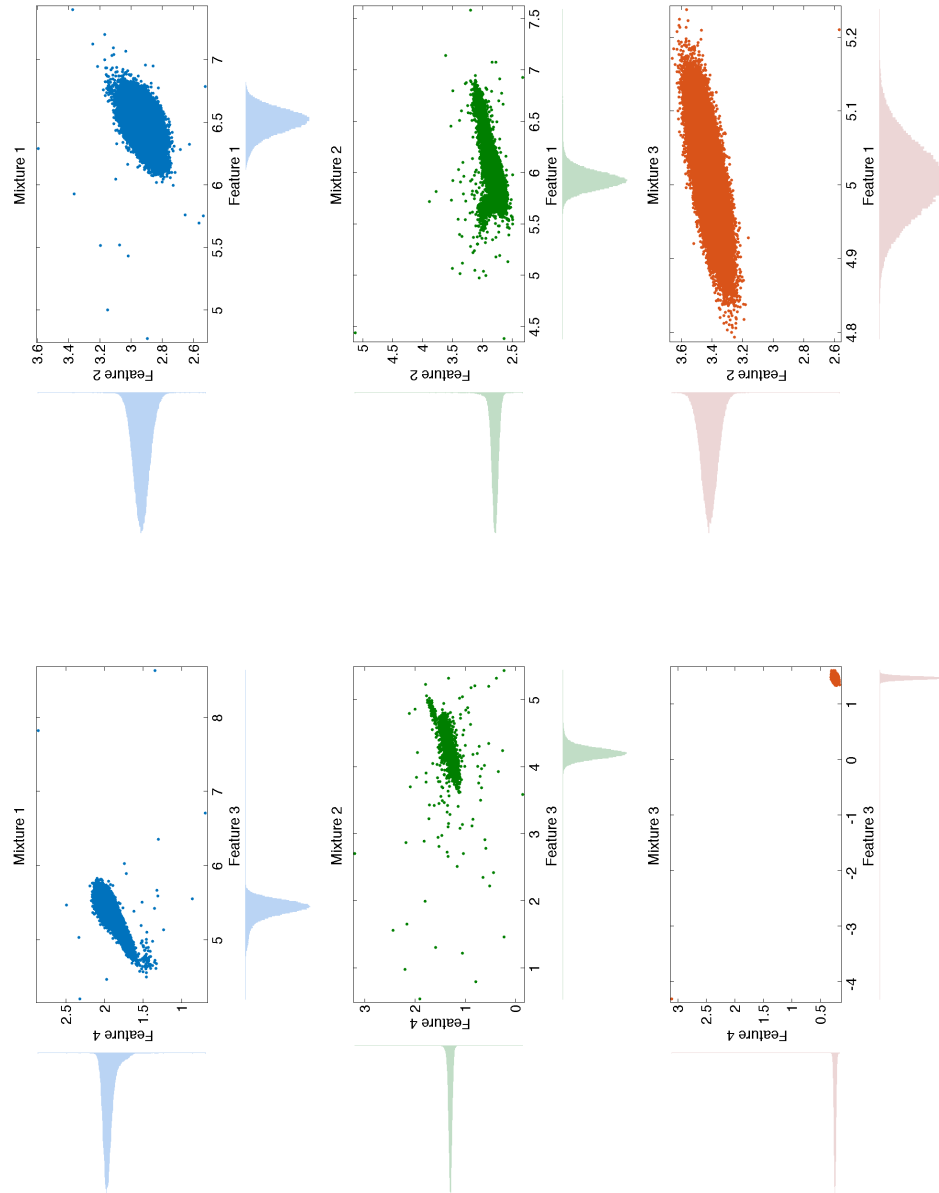


Figure 7.9: Trace distribution for the four features for each cluster after relabelling with K-means.

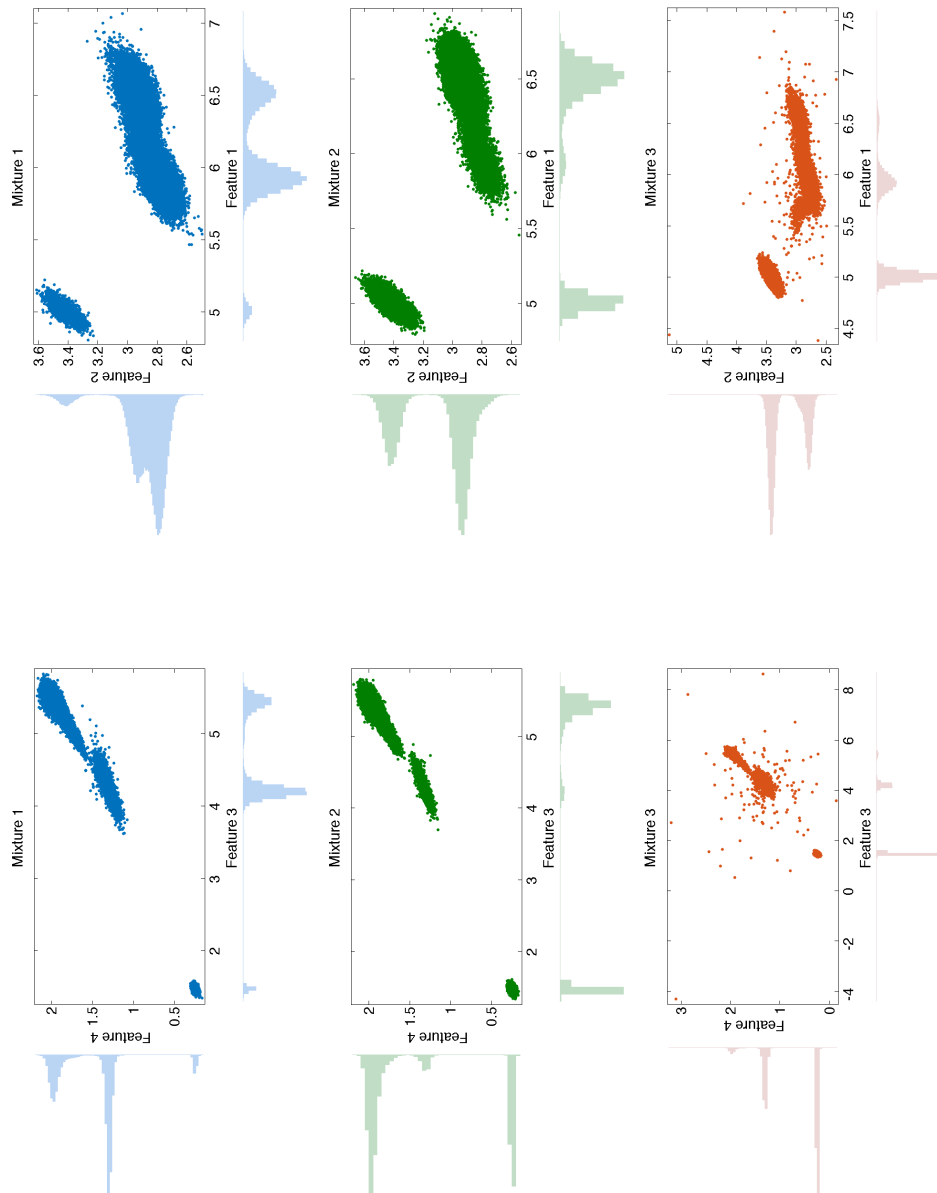


Figure 7.10: Trace distribution for the four features for each cluster after relabelling with ECR.

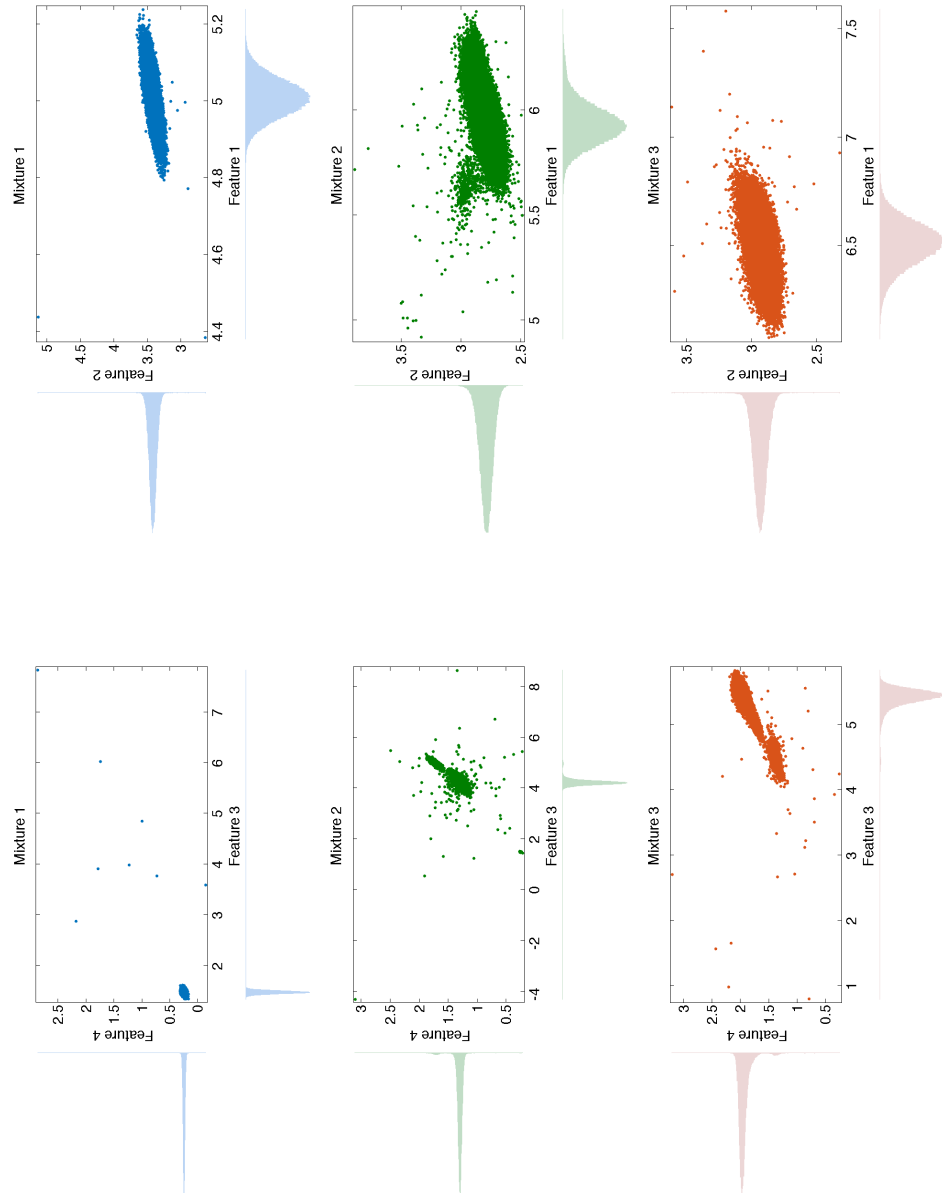


Figure 7.11: Trace distribution for the four features for each cluster after relabelling with IC. The constraint was upon the mean of feature one.

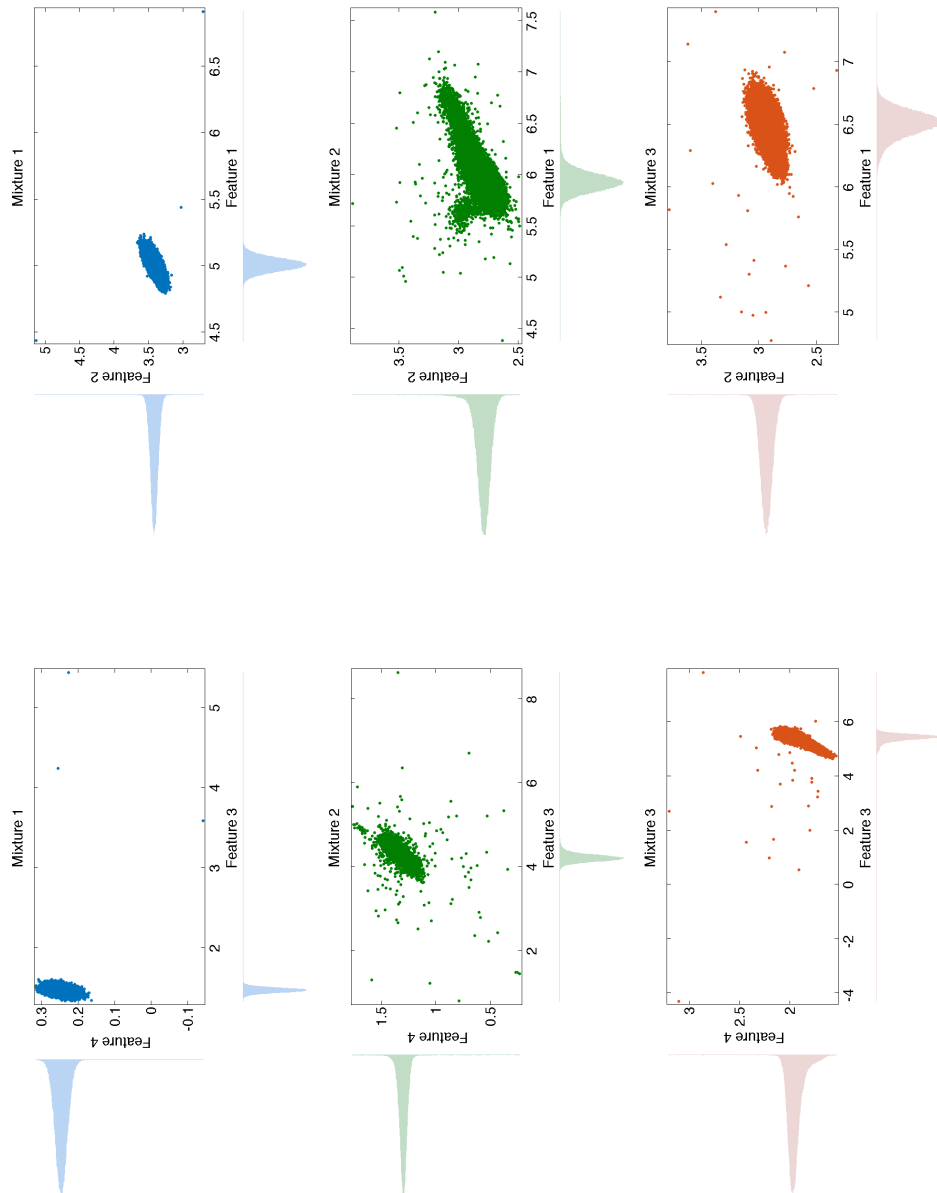


Figure 7.12: Trace distribution for the four features for each cluster after relabelling with IC. The constraint was upon the mean of feature four.

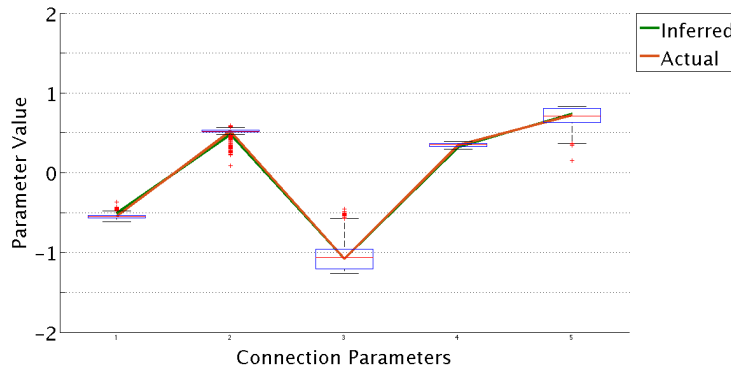
	Cluster	Feature 1	Feature 2	Feature 3	Feature 4
Ref.	1	$5.0075 \pm 0.0494$	$3.4174 \pm 0.0529$	$1.4685 \pm 0.0301$	$0.2459 \pm 0.0167$
	2	$5.9385 \pm 0.1245$	$2.7942 \pm 0.0625$	$4.2089 \pm 0.1111$	$1.2993 \pm 0.042$
	3	$6.5014 \pm 0.1057$	$2.9330 \pm 0.0498$	$5.4066 \pm 0.1406$	$1.9472 \pm 0.0755$
IC(1)	1	$5.0075 \pm 0.0495$	$3.4174 \pm 0.0533$	$1.4687 \pm 0.0432$	$0.2460 \pm 0.0212$
	2	$5.9338 \pm 0.1043$	$2.7919 \pm 0.0561$	$4.2169 \pm 0.1417$	$1.3048 \pm 0.0660$
	3	$6.5061 \pm 0.1021$	$2.9353 \pm 0.0505$	$5.3984 \pm 0.1791$	$1.9417 \pm 0.1015$
IC(4)	1	$5.0075 \pm 0.0498$	$3.4174 \pm 0.0533$	$1.4685 \pm 0.0344$	$0.2459 \pm 0.0168$
	2	$5.9386 \pm 0.1240$	$2.7941 \pm 0.0618$	$4.2092 \pm 0.1060$	$1.2992 \pm 0.0396$
	3	$6.5013 \pm 0.1067$	$2.9330 \pm 0.0501$	$5.4063 \pm 0.1483$	$1.9474 \pm 0.0757$
K-means	1	$5.0075 \pm 0.0494$	$3.4174 \pm 0.0530$	$1.4684 \pm 0.0352$	$0.2460 \pm 0.0190$
	2	$5.9381 \pm 0.1222$	$2.7940 \pm 0.0618$	$4.2088 \pm 0.1059$	$1.2995 \pm 0.0420$
	3	$6.5018 \pm 0.1062$	$2.9332 \pm 0.0501$	$5.4067 \pm 0.1416$	$1.9471 \pm 0.0766$
ECR	1	$5.3857 \pm 0.5033$	$3.1870 \pm 0.3011$	$2.5638 \pm 1.4213$	$0.6756 \pm 0.5648$
	2	$5.9748 \pm 0.6986$	$3.0859 \pm 0.2411$	$4.0334 \pm 1.8290$	$1.3456 \pm 0.7903$
	3	$6.0869 \pm 0.3774$	$2.8716 \pm 0.1506$	$4.4870 \pm 0.8928$	$1.4712 \pm 0.4158$

Table 7.4: Cluster means after relabelling with K-means, IC(1), IC(4) and ECR.

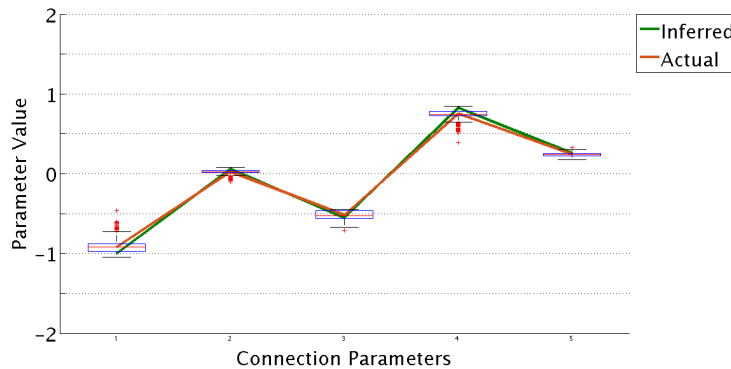
## 7.5 Dynamic causal models

For the unified framework a simulation to infer on the connectivity parameters of ten subjects as well as cluster these ten subjects were performed. The number of clusters was set to two, one simulation with feature selection and one without was performed. A linear DCM was used.

Figure 7.13 shows the inferred connectivity parameters and figure 7.14 the same when feature selection was used. Both models resulting in satisfying estimates of the connectivity parameters, reflecting the true values. Both models obtained a score of 1 for the NMI and the BP. Overall, both models result in accurate estimates on par with each other.

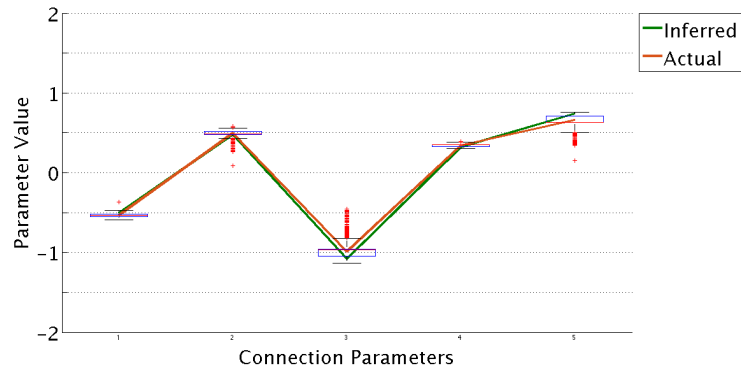


(a) Cluster 1.

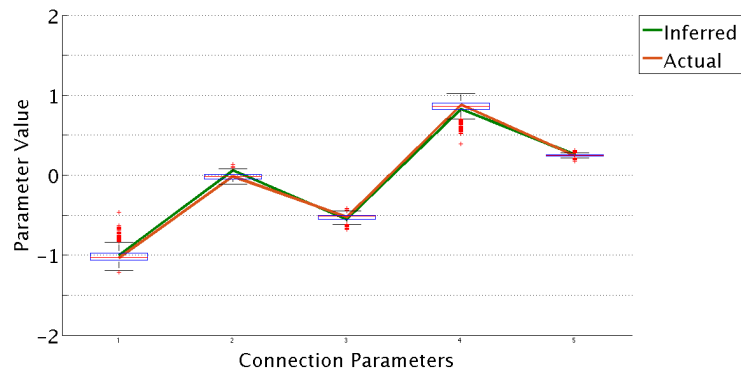


(b) Cluster 2.

Figure 7.13: Box plot of the parameter values for the two obtained clusters without feature selection. The green line is the true values and the red line is the inferred value.



(a) Cluster 1.



(b) Cluster 2.

Figure 7.14: Box plot of the parameter values for the two obtained clusters with feature selection. The green line is the true values and the red line is the inferred value.

The inferred probability of including each feature is presented in table 7.5. All features have a high probability of being included, reflecting the difference between the true parameter values of the two clusters.

Method	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
FGMMFS	0.9616	0.9920	0.9857	0.8200	0.9551

Table 7.5: Probability of including each feature in the clustering of connectivity parameters.

## Chapter 8

# Conclusions

In this thesis discriminant models for unsupervised learning is introduced. Both a parametric and non parametric model is described and evaluated. Both obtaining satisfying results on real and synthetical data sets. Feature selection is introduced to improve the clustering in the instance of noisy features. This addition proves to greatly improve the clustering performance and aide the interpretability of the inferred results. The feature selection proves useful even in cases without noisy features, with the inferred probability of including a feature acting as a measure of relative importance. This measure of relative importance helps the interpretation of the results by pointing out the most important features for the discrimination between clusters.

The occurrence of label switching in the context of MCMC and mixture models is introduced. Relabelling algorithms, with the aim of removing the affects of the label switching, is presented and compared. An algorithm based on K-means is found to give satisfying results, leading to accurately inferred cluster labels and unimodal sample distributions without having any major concerns when applied.

The feature selection is added to an unified framework for generative embedding in the context of DCMs for multi-subject analysis. The obtained results from the unified framework concurred with the findings of the standalone feature selected clustering models when noisy features are not present. The relative importance of features can come of use for practitioners when inferring on the underlying differences between, for instance, diagnosed and healthy subjects.

### 8.1 Further research

Further extensions to the unified framework could be made to incorporate multi-view clustering. Multi-view clustering on the connectivity parameters could possibly lead to various cluster structures depending on the feature space for each view. The extension of the feature selection to multi-view could yield new insights into the differences in the dynamics of the brain between groups.



To improve the efficiency of the sampler a method enhancing the label switching could be introduced, improving the exploration of the state space.

Further experiments should be carried out with the unified model to examine its robustness and precision. A comparison should be done to the method used by Brodersen et al. [2013] on the mentioned schizophrenia data set.

# Bibliography

- Aldous, D. (1985). Exchangeability and related topics. In Hennequin, P., editor, *École d'Été de Probabilités de Saint-Flour XIII — 1983*, volume 1117 of *Lecture Notes in Mathematics*, pages 1–198. Springer Berlin Heidelberg.
- Almuallim, H. and Dietterich, T. G. (1991). Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 547–552. AAAI Press.
- Bache, K. and Lichman, M. UCI machine learning repository.
- Bennett, J. E., Racine-Poon, A., and Wakefield, J. C. (1996). *Markov chain Monte Carlo in practice*, chapter MCMC for nonlinear hierarchical models, pages 339–357. Springer.
- Besag, J., Green, P., Higdon, D., and Mengersen, K. (1995). Bayesian computation and stochastic systems. *Statistical Science*, 10(1):pp. 3–41.
- Bishop, C. M. (October 1, 2007). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer.
- Brodersen, K., Schofield, T., Leff, A., Ong, C., Lomakina, E., Buhmann, J., and Stephan, K. (2011). Generative embedding for model-based classification of fMRI data. *PLoS computational biology*, 7(6).
- Brodersen, K. H., Deserno, L., Schlagenhaut, F., Lin, Z., Penny, W. D., Buhmann, J. M., and Stephan, K. E. (2013). Dissecting psychiatric spectrum disorders by generative embedding. *NeuroImage, Clinical*(4):98–111.
- Buxton, R. B., Wong, E. C., and Frank, L. R. (1998). Dynamics of blood flow and oxygenation changes during brain activation: The Balloon model. *Magnetic Resonance in Medicine*, 39(6):855–864.
- Cardie, C. (1993). Using decision trees to improve case-based learning. In *In Proceedings of the Tenth International Conference on Machine Learning*, pages 25–32. Morgan Kaufmann.
- Castren, E. (2005). Is mood chemistry? *Nat Rev Neurosci*, 6(3):241–246.

- Chang, J. and Fisher III, J. W. (2013). Parallel sampling of DP mixture models using sub-clusters splits. In *Proceedings of the Neural Information Processing Systems (NIPS)*.
- Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335.
- Cover, T. M. and Thomas, J. A. (2006). *Elements of information theory*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, 2nd edition.
- Diebolt, J. and Robert, C. P. (1994). Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(2):pp. 363–375.
- Fisher, S. R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.
- Friston, K., Harrison, L., and Penny, W. (2003). Dynamic causal modelling. *NeuroImage*, 19:1273–1302.
- Friston, K. J., Mechelli, A., Turner, R., and Price, C. J. (2000). Nonlinear responses in fMRI: the Balloon model, Volterra kernels, and other hemodynamics. *NeuroImage*, 12:466–477.
- Fruhwirth-Schnatter, S. (2001). Markov chain Monte Carlo estimation of classical and dynamic switching and mixture models. *Journal of the American Statistical Association*, 96(453):194–209.
- Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press Professional, Inc., San Diego, CA, USA, 2 edition.
- Gamerman, D. and Lopes, H. F. (2006). *Markov chain Monte Carlo: stochastic simulations for Bayesian inference*. Chapman and Hall/CRC, 2 edition.
- Grubb, R., Raichle, M., Eichling, J., and Ter-Pergossian, M. (1974). The effects of changes in paco<sub>2</sub> on cerebral blood volume, blood flow and vascular mean transit time. *Stroke*, 5:630–639.
- Guyon, I. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.
- Jain, S. and Neal, R. M. (2004). A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1):158–182.
- Jasra, A., Holmes, C. C., and Stephens, D. A. (2005). Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, 20(1):50–67.

- Kim, S., Tadesse, M. G., and Vannucci, M. (2006). Variable selection in clustering via Dirichlet process mixture models. *Biometrika*, 93(4):877–893.
- Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324.
- Liang, F. and Wong, W. H. (2001). Real-parameter evolutionary Monte Carlo with applications to Bayesian mixture models. *Journal of the American Statistical Association*, 96(454):653–666.
- Lu, H. and van Zijl, P. (2005). Experimental separation of intra and extravascular BOLD effects using multi-echo VASO and BOLD fMRI at 1.5T and 3.0T. *Magnetic Resonance in Medicine*, 53(808–816).
- Neal, R. M. (1996). Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6(4):353–366.
- Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265.
- Niu, D., Dy, J. G., and Ghahramani, Z. (2012). A nonparametric Bayesian model for multiple clustering with overlapping feature views. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, April 21-23, 2012*, pages 814–822.
- Obata, T., Liu, T., Miller, K., Luh, W., Wong, E., Frank, L., and Buxton, R. (2004). Discrepancies between BOLD and flow dynamics in primary and supplementary motor areas: application of the Balloon model to the interpretation of BOLD transients. *NeuroImage*, 21:144–153.
- Papastamoulis, P. and Iliopoulos, G. (2010). An artificial allocations based solution to the label switching problem in Bayesian analysis of mixtures of distributions. *Journal of Computational and Graphical Statistics*, 19(2):313–331.
- Pascual-Leone, A., Freitas, C., Oberman, L., Horvath, J. C., Halko, M., Eldaief, M., Bashir, S., Vernet, M., Shafi, M., Westover, B., Vahabzadeh-Hagh, A. M., and Rothenberg, A. (2011). Characterizing brain cortical plasticity and network dynamics across the age-span in health and disease with TMS-EEG and TMS-fMRI. *Brain Topography*, 24(3-4):302–315.
- Raman, S. S. and Stephan, K. E. A unified framework for clustered connectivity in dynamic causal models.
- Rasmussen, C. E. (2000). The infinite Gaussian mixture model. *Advances in neural information processing systems*, 12:554–560.
- Silvennoinen, M., Clingman, C., Golay, X., Kauppinen, R., and van Zijl, P. (2006). Comparison of the dependence of blood R2 and R2\* on oxygen saturation at 1.5 and 4.7 Tesla. *Magnetic Resonance in Medicine*, 49(47–60).

- Stephan, K. E., Friston, K. J., and Frith, C. D. (2009). Dysconnection in schizophrenia: From abnormal synaptic plasticity to failures of self-monitoring. *Schizophrenia bulletin*, page sbn176.
- Stephan, K. E., Weiskopf, N., Drysdale, P. M., Robinson, P. A., and A, K. J. F. (2007). Comparing hemodynamic models with DCM. *NeuroImage*, 38:387–401.
- Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809.
- Strehl, A. and Ghosh, J. (2002). Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617.
- Yao, W. (2012). Bayesian mixture labeling and clustering. *Communications in Statistics - Theory and Methods*, 41(3):403–421.

# Appendix A

## Data Sets

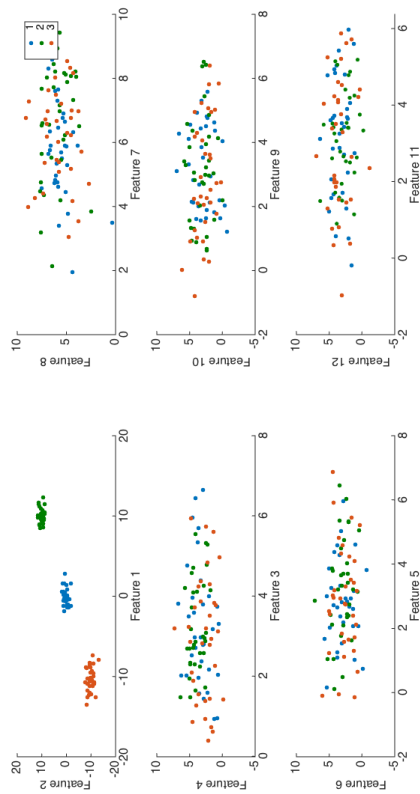


Figure A.1: Scatter of synthetic dataset with two relevant features. The relevant features are (1, 2), which are shown in the figure.

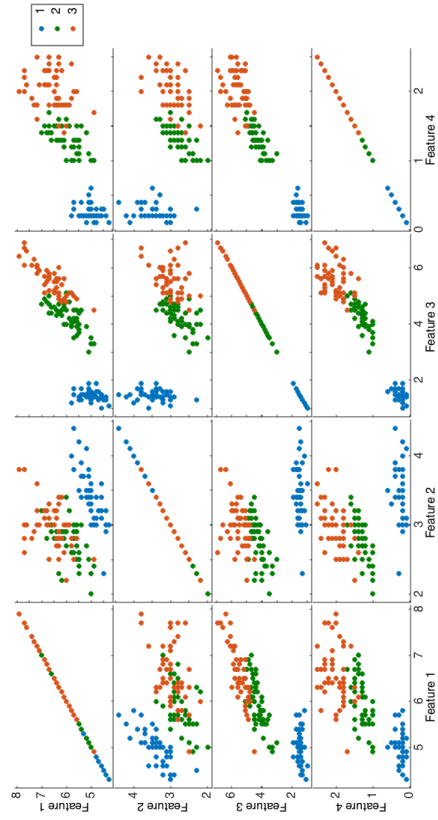


Figure A.2: Scatter of Iris dataset.