Variational Methods for Bayesian Independent Component Analysis

Rizwan A. Choudrey
Pattern Analysis and Machine Learning
Robotics Research Group

A THESIS SUBMITTED TO THE
UNIVERSITY OF OXFORD
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

SOMERVILLE COLLEGE MICHAELMAS TERM 2002

Variational Methods for Bayesian Independent Component Analysis

Rizwan A. Choudrey

A thesis submitted to the University of Oxford
for the degree of Doctor of Philosophy

Pattern Analysis and Machine Learning
Department of Engineering Science

Somerville College
Michaelmas 2002

Abstract

The fundamental area of research in this thesis is Independent Component Analysis (ICA). ICA is a tool for discovering structure and patterns in data by factoring a multidimensional data distribution into a product of one-dimensional, statistically independent component distributions. Statistical independence is equivalent to information-theoretic independence. Therefore, if the original M-dimensional data distribution is factored into $L \leq M$ independent components, then the M streams of observed numbers, which may have structure and information encoded *across* them, are transformed into L independent streams of numbers which will have structure and information encoded only *within* each stream. This has the effect of making patterns within the original data more cogent.

Traditional ICA methods, however, can be limited in the flexibility of their decompositions, particularly in the modelling of component distributions and ascertaining the most appropriate number of components to adequetly represent the observed data distribution. This thesis aims to develop a more flexible formulation of ICA which can overcome these limitations, allowing ICA to be applied to a wider range of data analysis problems. The thesis is presented in eight chapters and can be considered an exposition in two parts. Chapters 1-4 constitute the first part, detailing the theory underpining the work in the second part, Chapters 5-8. After a general introduction in Chapter 1, Chapter 2 introduces ICA theory and highlights areas where traditional ICA methods are inadequate. Chapter 3 explains the benefits of Bayesian learning and how it can be used to extend ICA. Chapter 4 derives the variational approximation to Bayesian learning in order to make Bayesian ICA practical. Chapter 5 constructs a flexible ICA model learnt using variational Bayesian methods, which will be shown to outperform traditional ICA methods. Chapter 6 extends the single ICA formalism to a mixture of ICAs model, allowing different areas in the data space to be described by different locally-adapted ICA models and thus increasing the flexibility of the ICA process still further. Chapter 7 develops dynamic ICA models in order to uncover any temporal information that may be present within the data, information that is traditionally ignored. Chapter 8 concludes the thesis with a summary, discussion of outstanding problems and possible directions for future research.

Acknowledgements

First and foremost, I must thank my supervisor Stephen Roberts for his help, advice and support; I would not be in the position I am now without his incredible faith and confidence in me, especially in times when mine own was found wanting. I am deeply grateful to Iead Rezek for his insight, understanding and unbridled help whenever I have needed it, and in dispensing computing expertise in situations where my cornucopian talents in such areas have evaded me. Similar gratitude must extend to Mike Gibbs for relentlessly mending computers after situations where such talents have returned. Thanks also to Peter Sykacek for enlightening discussions that have helped me gain a deeper understanding of my research than I would have otherwise, and for continuously subjecting me to his own brand of Teutonic wit. I would like to thank my former collegue Will Penny for all the invaluable mathematical and programming help he has given, but not for likening me to Ali G. My warmest and sincerest gratitude, though, must go to Evangelos Roussos for buying me cakes whenever I have needed them most, and to Lyndsey Pickup for baking them (even though she tries to keep them from me).

Most importantly, I must thank my parents Niaz and Tasneem Choudrey for making me who I am, for always trusting me in making the right decision - even when I make the wrong one - and for supporting me in whatever I do, especially in situations where they have tried to explain to friends and family why I'm not married yet. Finally, through gritted teeth, I have to thank my brother's Zeeshan and Sumran for constantly asking 'When will you finish?', 'Have you finished yet?', 'Why haven't you finished yet?' and the inevitable 'When are you going to get a bloody job?'. At least now they can shut up about the first three.

Contents

1 Introduction **1**

1.1 Statistical Pattern Recognition 2

1.1.1 Probability theory 5

1.1.2 Information theory 11

1.1.3 Common functions and distributions 12

1.2 Scope 14

1.3 Contributions 15

1.4 Overview 17

2 Independent Component Analysis **20**

2.1 Why Independence? 21

2.1.1 Intrinsic coordinate system 21

2.1.2 Neurological processing 28

2.2 Applications 33

2.3 The ICA Problem 34

2.3.1 ICA as a mapping problem 35

2.3.2 ICA as a modelling problem 36

2.3.3 Measuring independence 37

2.4 History 42

2.4.1 Mapping approach 42

2.4.2 Modelling approach 44

2.4.3 Unification 45

2.5 Generative Model 46

2.5.1 Identifiability of ICA solution 47
2.5.2 Likelihood 49
2.6 Simple Example - Square, Noiseless ICA 50
2.6.1 How does ICA work? 51
2.6.2 Using the likelihood to learn 53
2.6.3 Results 55
2.7 Problems and Limitations 57
2.8 Summary 58

3 Bayesian Modelling **61**
3.1 Bayes Theorem 62
3.2 Graphical Models 63
3.3 Bayesian Inference 65
3.3.1 Learning the model 66
3.3.2 Using the model 68
3.4 Model Comparison 69
3.5 Priors - Good or Bad? 70
3.6 Approximations to Bayesian Inference 71
3.6.1 Maximum likelihood 71
3.6.2 Maximum a posteriori 72
3.6.3 Expectation-Maximisation algorithm 73
3.6.4 Evidence approximation 74
3.6.5 Monte Carlo methods 75
3.7 Approximations to Model Comparison 76
3.8 Summary 77

4 Variational Approximation **78**
4.1 Derivation 79
4.2 Maximising the Objective Function 81
4.3 Variational Method for Generative Models 83
4.4 An Example - Mixture of Gaussians 83

4.4.1 The model 84
4.4.2 Variational Bayesian learning for a MoG 86
4.4.3 Optimising the posteriors 88
4.4.4 Evaluating the negative free energy 91
4.4.5 Results 95
4.4.6 The effect of priors 97
4.5 Summary 101

5 Variational Bayesian Independent Component Analysis 103
5.1 ICA - The State of the Art 104
5.2 The Proposed Model 105
5.2.1 Source Model 106
5.3 Variational Bayes for ICA 107
5.3.1 The Priors 107
5.3.2 Variational Methodology 108
5.3.3 The Posteriors 111
5.3.4 Hierarchical Interpretation 115
5.3.5 Implementing vbICA 115
5.4 Results 119
5.4.1 Toy Data 119
5.4.2 Comparison with other methods 126
5.4.3 Comparison of vbICA algorithms 131
5.5 Using Prior Constraints 136
5.5.1 Automatic Relevance Determination 136
5.5.2 Positivity 140
5.6 Real Data - Removing Signal Artifacts 144
5.7 Discussion 146
5.7.1 The Problem of Correlated Data 147
5.7.2 Further Applications and Extensions 148

6 Mixtures of Independent Component Analysers 150

6.1 Why? . . . 151

6.2 The Generative Mixture Model . . . 153

6.3 Variational Bayesian Mixture of ICAs . . . 154

6.3.1 ICA Source Model . . . 156

6.3.2 Variational Learning for vbMoICA . . . 157

6.3.3 The Posteriors . . . 158

6.3.4 Implementing vbMoICA . . . 162

6.3.5 Results . . . 164

6.4 Real data - Image Decomposition . . . 168

6.5 Discussion . . . 173

6.5.1 Problems . . . 173

6.5.2 Further Applications . . . 174

6.5.3 Extensions . . . 175

7 Dynamic Independent Component Analysis 177

7.1 Hidden Markov Models . . . 179

7.2 Learning and Inference . . . 182

7.2.1 Evaluating the likelihood . . . 183

7.2.2 Learning the parameters . . . 183

7.2.3 The Forward-Backward algorithm . . . 184

7.2.4 Inferring the state path . . . 188

7.3 Bayesian Formalism . . . 189

7.4 Dynamic ICA Models . . . 191

7.4.1 ICA with HMM Sources . . . 192

7.4.2 HMM with ICA Generators . . . 198

7.5 Real Data - Hierarchical Dynamics . . . 200

7.6 Discussion . . . 204

7.6.1 Problems . . . 205

7.6.2 Further Applications and Extensions . . . 205

8 Close **207**

8.1 Summary . . . 207

8.2 Problems to Overcome . . . 210

8.2.1 Initialisation . . . 210

8.2.2 Speed . . . 212

8.2.3 Active model selection . . . 212

8.3 Future Directions . . . 213

8.3.1 Further source models . . . 213

8.3.2 Fully dynamic models . . . 213

8.3.3 Nonlinear ICA . . . 213

8.3.4 Blind Deconvolution . . . 214

8.3.5 Overcomplete sources . . . 214

8.4 Conclusion . . . 214

A Probability Distributions **230**

A.1 Gaussian . . . 230

A.2 Gamma . . . 231

A.3 Dirichlet . . . 232

A.4 Exponential . . . 232

A.5 Truncated Gaussian . . . 233

B Derivations for vbICA1 **235**

B.1 Source Model . . . 235

B.2 Network Model . . . 238

B.3 Free Energy . . . 240

C Derivations for vbICA2 **242**

C.1 Network Model . . . 242

C.2 Source Model . . . 243

C.3 Energy . . . 245

D Derivations for Prior Constraints **246**

D.1 ARD 246

D.2 Positivity 247

E Derivations for vbMoICA **248**

E.1 Mixture Variables 248

E.2 Energy 249

F Derivations for vbHMM **250**

F.1 HMM Variables 250

F.2 Energy 252

Chapter 1

Introduction

The human mind is a *tour de force* classifier and seeks to pigeon-hole everything it comes across, including this thesis. In the academic world, this often counter-productive endeavour is exacerbated by the *disjecta membra* that is scientific discipline. Is this a maths thesis? Certainly, there is a liberal sprinkling of mathematics, but no more so than a physics tome. Is it, then, an original work in physics? True, the central mathematical tool used is borrowed from statistical physics, but it is applied to problems in computation. Ah, so it must be a computer science doctorate. But there appear to be no lines of code, mention of flops and cycles, or, indeed, references to rotund and hirsute middle-aged men in penguin-emblazoned T-shirts? In fact, there are elements of all three plus some guest appearances. If one must distill 250+ pages into one label, then this thesis can be considered an exercise in finding patterns in data.

Data and information are, respectively, the resource and commodity of the modern world. Data is extracted from the world like crude oil from a reservoir - in its rawest form, data is of little use. It is only when information is extracted via processing - like petroleum from oil - that data becomes 'meaningful'. The discovery, analysis and recognition of patterns is of paramount importance if information is to be extracted from the raw observations on the world. This is something that comes naturally to humans. The human brain is a pattern recogniser *par excellance* and, as such, mathematical pattern recognition forms the foundation of 'intelligent' data processing by computers. The mathematical

study of patterns is rapidly becoming an area of huge significance, particularly as the world becomes increasingly data-driven and information-hungry. New novel and anthropogenic methods will be needed to interrogate, process, organise, store, access and understand this data quickly and intuitively. This thesis focuses on one such method - Independent Component Analysis.

Data is presented to computers as numbers. Calculating and analysing the distribution of these numbers - how often certain numbers or groups of numbers appear - is of fundamental importance in finding patterns in those numbers. If there is only one stream of numbers, this process is well established. If the data consists of multiple, simultaneous observations on the world, the distribution of numbers will be a multidimensional entity and the task of finding patterns becomes much more difficult. Independent Component Analysis (ICA) is the practice of breaking down such distributions into a product of simpler, one-dimensional ones, called 'components'. To avoid problems usually involved in skinning cats, each of these components must contain - as far as possible - information different from each other component. This makes each component 'independent' from every other component. The foundations and mechanics of ICA are discussed in more detail in Chapter 2.

This work is aimed at anyone with a basic working-knowledge in the Calculus. A previous exposure to probability and information theory is advantageous for a deeper understanding, but by no means necessary as the rest of Chapter 1 acts as a gentle introduction to probabilities, why they are important to this work and how they can quantify something as nebulous as 'information'. The Chapter will conclude with an overview of the rest of the thesis.

1.1 Statistical Pattern Recognition

The tenth edition of the Concise Oxford English Dictionary defines a pattern as 'an arrangement or sequence regularly found in comparable objects or events'. Implicit in this definition is that said objects or events (or, similarly, observations) are not *identical*, but share fundamental characteristics that make them

comparable. For example, all trees look like trees, even though each tree is unique. Each tree exhibits the same underlying structure or pattern of a tree - trunk, branches, leaves etc. - while being different in detail to other trees. It is this recognition of the pattern of a tree underlying the 'noise' that is detail which allows one to classify a branchy, leafy thing with a trunk as a tree, rather than, say, as a cup of coffee.

How can this distinction be made using numbers? Imagine that the defining characteristics of a tree are 'branchyness', 'leafyness' and 'trunkyness' . Now, different trees have different amounts of each, so let each feature take a value between 0 and 1, signifying the relative prominence of that feature. For example, a tree measured as $[0.1, 0.1, 0.9]$ would be almost all trunk, with very little in the way of branches or leaves (perhaps struck by lightning?). All trees would have different ratios of each, with most concentrated around a 'typical' ratio. These statistics will form a distribution of points on a 3-dimensional graph, where each axis represents the value of each feature, and where each point represents a tree. Figure 1.1 shows an example distribution. To make subsequent manipulation more efficient, this distribution may be represented mathematically by a simpler model that captures the overall shape using a set of parameters governing macroscopic features such as mean, spread, symmetry etc., negating the need to (expensively) store all data points.

Now imagine making a further observation on the world by measuring the feature values of an unknown object, and plotting these values on the aforementioned graph. If this new point is well within the distribution representing trees, then this object will have a high probability of being a tree; if this new point is deemed to lie further out (using some measure based on, say, the mean and spread of the distribution), then it is less likely to be a tree. If it is, in fact, a cup of coffee, it will have values very close to $[0, 0, 0]$ so will have a high probability of *not* being a tree. The distribution of 'shrubs', on the other hand, is not so distinct as they exhibit appreciable levels of branchyness and leafyness, although little trunkyness. Extreme versions of trees and shrubs ('outliers') may

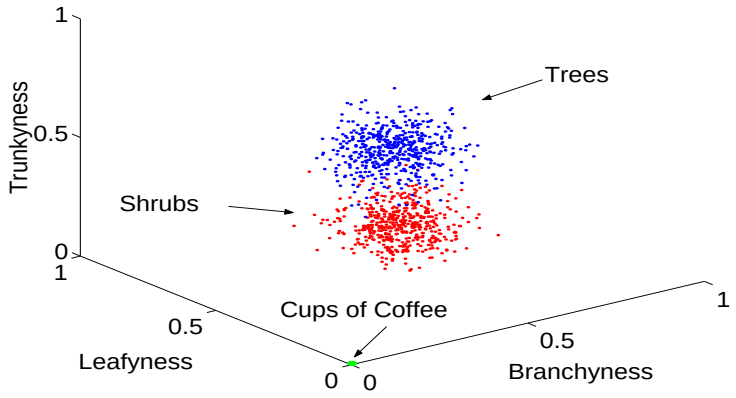


Figure 1.1: How to categorise objects using numbers.

look similar, so are harder to distinguish. New data points can be incorporated into the overall model, adjusting the model parameters to a greater or lesser degree depending on the probability score of each being a tree. As more and more observations come in, the model adapts becoming a better overall description of 'treeness' as data is gathered, increasing its power to discriminate between, say, trees and shrubs.

Note that the defined pattern is sensitive to the features used to describe the class of objects. How one chooses relevant features is a topic in itself and will not be discussed here, although Bayesian theory (see Chapter 3) is often helpful. For more information, see [1]. Generally, one would utilise universally applicable features, for example pixel values in an image, electrical signals from a sensor etc. These are the kind of observations dealt with in this thesis.

Although the example used is one of object recognition using 'clustering' in data, the modelling of data distributions - and of probabilities to incorporate further data and elicit information from these distributions - is applicable to all problems in statistical pattern recognition. The next section will cover the basics of probability and will introduce the notion of *conditional probability* which will be fundamental in understanding how to incorporate new information into a model, and how to elicit information from it. For an introductory text on

statistical pattern recognition, see [1].

1.1.1 Probability theory

Probability theory is concerned with describing the average behaviour of phenomena, so its importance in recognising patterns is clear. The axiomatic basis of probability theory is connected with the results of experiments and events arising from these results, and relies on set theory [2]. A complete exposition of set theory is beyond the scope of this thesis; what follows is a flavour of the fundmentals of probability theory and the important notion of *random variables*.

Axiomatic basis

An experiment has a number of possible elementary results, for example throwing a die yields the results that 1, 2, 3, 4, 5 or 6 is rolled. The collection of all possible elementary results forms a *set*, whose *elements* are 1, 2, 3, 4, 5 and 6. Associated with the possible results of the experiment are *events* which are statements that go beyond the results themselves. For example 'the number rolled is even' is not an elementary result of rolling a die, but is an event due to the result of the experiment. These events may be connected with one or more of the possible results, e.g. 'the number rolled is even' can be attributed to the results 2, 4 and 6. Events, therefore, form *subsets* of the overall set of possible elementary results. If $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$ denotes the possible results of rolling a die, then the event 'the number rolled is even' is the subset $\mathcal{E} = \{2, 4, 6\}$. If the set $\mathcal{X}$ has N elements, then there are 2^N possible subsets, $\mathcal{E}_i$, where $i = 1, \ldots, N$. This collection of subsets is called the *power set* of $\mathcal{X}$ and are all the possible events associated with the result of the experiment, including the *certain event* $\mathcal{E} = \mathcal{X}$ - i.e. the event 'the number rolled is either $1, 2, 3, 4, 5, 6$' is certain to occur - and the *null event* $\mathcal{E} = \emptyset \doteq \{\}$ which has no elements and is the event that there is not result.

This formulation is a way of mathematically coding, in a rigorous way, the statements one can make about the result of an experiment. In addition to simple statements about the occurances of single events, there are statements

one could make about event $\mathcal{E}_1$ occurring *or* event $\mathcal{E}_2$ occurring, or event $\mathcal{E}_1$ occurring *and* event $\mathcal{E}_2$ occurring, or even event $\mathcal{E}_1$ *not* occurring[1]. Mathematically, these are the set operations of *union*, *intersection* and *complement* respectively, performed on the relevant subsets denoted by $\mathcal{E}_i$. The union $\mathcal{E}_1 \cup \mathcal{E}_2$ is all the elements of $\mathcal{E}_1$ plus all the elements of $\mathcal{E}_2$. For example, the event ' "the number rolled is less than 3" or "the number rolled is even" ' is $\{1,2\} \cup \{2,4,6\} = \{1,2,4,6\}$. The intersection $\mathcal{E}_1 \cap \mathcal{E}_2$ are all the shared elements and is the event ' "the number rolled is less than 3" and "the number rolled is even" ', which gives $\{1,2\} \cap \{2,4,6\} = \{2\}$. The complement $\bar{\mathcal{E}_1}$ is the event ' "the number rolled is less than 3" does not happen' and is $\{3,4,5,6\}$. Please note that the results of an experiment need not be discrete outcomes in a finite range. However, the manipulations of sets with infinite elements - and the eliciting of probabilities thereafter - requires further theory which is beyond the scope of this introduction, although the basic concepts hold true. For further details, see [2].

The *probability* of an event $\mathcal{E}$ is a number $p(\mathcal{E})$ assigned to the event $\mathcal{E}$ that satisfies the following axioms

$$\text{I} \qquad p(\mathcal{E}) \geq 0 \tag{1.1}$$

$$\text{II} \qquad p(\mathcal{X}) = 1 \tag{1.2}$$

$$\text{III} \qquad \text{if} \quad \mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset \quad \text{then} \quad p(\mathcal{E}_1 \cap \mathcal{E}_2) = p(\mathcal{E}_1) + p(\mathcal{E}_2) \tag{1.3}$$

What this number represents in 'real-life' is a thorny, philosophical issue. The 'frequentist' school of thought equate probabilities to the long-term fractional outcomes of experiments, i.e. if $p(\mathcal{E}) = 0.7$, then that event will occur 70% of the time in an infinite sequence of experiments. The 'Bayesian' school equates probabilities to 'degrees of belief', where $p(\mathcal{E}) = 0.7$ means one believes *ahead of time* that 70% of future events will be $\mathcal{E}$. The interpretation of probabilities is discussed in more detail in section 1.1.1; suffice to say, it is these interpretations that make the connection between mathematical definition and 'real-life'.

[1] of course, each of these statements is an event in itself.

Although this axiomatic layer is the foundation for probability theory, it is the concept of *random variables* that is crucial for statistical pattern recognition.

Random variables

A random variable (rv), X, is a variable that can take a range of values depending on the particular result of an experiment. For example, in gambling one may assign a monetary value to each of the faces of a die. This relationship maybe tabular or functional. The possible winnings is an rv, X; the actual amount won on a particular throw is the instantiated value $X = x$ of the rv.

The probability axioms above imply that a probability is a number between 0 and 1 and thus is a mapping from a unique event to the interval $[0, 1]$. A random variable is a reversable mapping from a unique elementary result to the real line. Therefore, a probability can also measure the likelihood of a random variable X taking a particular value x. It is this relationship between probabilities and random variables that is exploited in pattern recognition, allowing observation data to be viewed and analysed in a probabilistic manner.

The set of possible values an rv can take is called its sample space, Ω_X. These values may be discrete, $\{x^1, ..x^k.., x^n\}$, or continuous within an (possibly infinite) interval $[a, b]$. Although there is a formal distinction between the variable X on the one hand, and its instantaneous value x, the probability of taking this value is usually denoted $p(x)$. The collection of probabilities $\{p(x^1), ..p(x^k).., p(x^n)\}$ is called the probability distribution of x; in the continuous case, $p(x)$ for $a \leq x \leq b$ is also known as the probability density function (pdf) of x. The variable cannot take more than one value at a time (states are mutually exclusive) and *must* take one of the values in the sample space (states are exhaustive - the certain event). Axiom II in (1.2) means that the sum of all probabilities over a sample space, Ω_X, equals unity

$$\sum_{\Omega_X} p(x) = 1 \qquad \text{Discrete} \tag{1.4}$$

$$\int_{\Omega_X} p\,(x)\,dx = 1 \qquad \text{Continuous} \tag{1.5}$$

If there are two variables, X and Y, the probability of $X = x$ and $Y = y$ is described by the joint probability distribution $p(x, y)$. The sample space is given by the Cartesian product of the two individual sample spaces $\Omega_{XY} = \Omega_X \Omega_Y$, i.e. the conjunction of all possible values. To calculate the probability of $X = x$ regardless of the value Y takes, one can use (1.4)/(1.5) to *marginalise* over y

$$p(x) = \sum_{\Omega_Y} p(x, y) \qquad \text{Discrete} \tag{1.6}$$

$$p(x) = \int_{\Omega_Y} p(x, y)\, dy \qquad \text{Continuous} \tag{1.7}$$

This is known as the *marginal* probability distribution of x.

The notion of a joint distribution may be similarly extended to N variables, where the distribution is N-dimensional. In the discrete case, this is an N-dimensional table, while the volume covered by an N-dimensional pdf is exponential in the number of variables. Each entry in the table, or each point in the probability space, corresponds to a particular combination of values that the N variables can take. Marginalisation over $M < N$ variables is similar to (1.7), where the summation/integration is carried out over the M state spaces.

Conditional probability

In the case of pattern recognition, one is interested in the probability of a value given some additional information or evidence.

The *conditional distribution* $p(x|y)$ is read as 'probability of event X=x given event/information Y=y' and is defined

$$p(x|y) = \frac{p(x, y)}{p(y)} \tag{1.8}$$

provided $p(y) \neq 0$ (otherwise $p(x|y)$ is not defined). A variable X is *independent* of variable Y if and only if (iff)

$$p(x, y) = p(x)\, p(y) \tag{1.9}$$

Permuting the variables in (1.8) and noting that $p(y, x) = p(x, y)$, an important expression can be derived

$$p(y|x) = \frac{p(x|y)\, p(y)}{p(x)} \tag{1.10}$$

This is known as Bayes' theorem. The quantity $p\left(y|x\right)$ is known as the *posterior* probability of y, $p\left(x|y\right)$ is the *likelihood* of x and $p(y)$ is the *prior* probability of y. The denominator $p(x)$ is the marginal likelihood of x and is given by

$$p\left(x\right) = \int_{\Omega_Y} p\left(x|y\right) p(y) dy \tag{1.11}$$

and follows from the constraint of (1.5) (summation if discrete).

Bayes' theorem is fundamental to modern artificial intelligence and machine learning as it allows the update of one's beliefs in the light of new information. Consider a model of the world embodied by p(world). Implicit in this is the behaviour one would expect to see under this prior model. Now make an observation on the real world. The likelihood of this observation under the prior model is given by p(observation|world). The new model of the world after this observation has been made is given by the posterior p(world|observation), computed using Bayes' theorem. If the observation matched expected behaviour, then $\frac{p(\text{observation}|\text{world})}{p(\text{observation})}$ would be close to one, and the posterior view would be similar to the prior view, and vice versa. In the case of discerning trees from cups of coffee, a prior probability would indicate how likely a tree is to be observed regardless of whether an observation is made or not. This could, for example, be based on information that the data about to be analysed was gathered from a park, or conversely, from an office. From past observations known to be from trees (or elsewhere), a likelihood can be constructed coding what kind of data points are expected from observations on trees, and what are not. An observation, x, is then presented and the likelihood of this new data point is calculated. Bayes' theorem can the be used to compute the posterior probability

$$p\left(\text{tree}|x\right) = \frac{p\left(x|\text{tree}\right) p(\text{tree})}{p\left(x|\text{tree}\right) p(\text{tree}) + p\left(x|\text{not tree}\right) p(\text{not tree})} \tag{1.12}$$

that this data point was generated from a tree and thus can be classified.

Bayes' theorem also allows one to infer information that is not readily accessible. The belief that a patient has a disease given a particular set of symptoms is not directly obtainable from statistics. However, the prior probability of

getting a disease, and the likelihood of symptoms given the presence of that disease, can be measured using a frequentist approach on available epidemiological statistics. A posterior diagnosis can now be made by invoking Bayes' theorem. Chapter 3 explains Bayesian inference and modelling in more detail.

Interpretation of probabilities

The philosophical interpretation of a probability is the subject of much debate, and often polarises into two schools of thought. In one corner are the 'frequentists', who view probabilities as 'frequencies' of particular outcomes in an infinite number of events. In the other corner are 'Bayesians', who view probabilities as 'degrees of belief'. For example, a frequentist would argue that if the probability of a coin toss turning up heads is 0.5, this means that a coin would turn up heads half the time in an infinite number of tosses [2]. A Bayesian would argue that if the probability of a coin toss turning up heads is 0.5, this means one *believes* ahead of time that the the coin can turn up heads or tails with equal likelihood. Now, this belief may very well be based on *past* experiments, but the probability assigned is still a measure of belief if the coin is not tossed an infinite number of times. The difference is subtle, but the consequences are far-reaching. A frequentist will not assign a probability to an event that cannot be tested, in principle, an infinite number of times; a Bayesian can. If a student states 'the probability of finishing this thesis on time is 30%', this assertion is clearly nonsensical in a frequentist framework as finishing the thesis cannot be subject to infinite experiments. What the student is asserting is a degree of belief. This may be based on previously similar - but *different* - situations.

Cox showed in [3] that 'beliefs' can be manipulated in a rigorous and principled manner if based on the axioms of probability theory, in particular Bayesian probability theory. This is what allows learning and intelligence to be treated mathematically.

[2] More strictly, the fraction of tosses that turned up heads *approaches* 0.5 as the number of tosses *approaches* infinity (as half of infinity is infinity).

1.1.2 Information theory

The quantitative notion of information was derived by Shannon [4] by equating probability distributions with 'states of knowledge'. Shannon - a telecommunications engineer - was interested in the communication of signals using symbols. As Shannon showed, information in this case was equated with the total number of patterns an information source, with a given dictionary of symbols, was capable of producing. Essentially, the argument goes like this. Certain symbols (e.g. numbers) are produced. If each symbol is as likely as another, then the observer is equally uncertain about which symbol will be produced. The observer is equally 'surprised' by the output of each symbol, so each symbol conveys an equal amount of information. If, however, different symbols have different likelihoods of appearing, then there will be more surprise (and more information conveyed) with less likely symbols than with the production of more likely ones.

It is clear that probabilities have an intimate relationship with information. The uncertainty regarding the production of symbols is encoded in the probability distribution over them. Therefore, it is reasonable to assume that the information content of a signal is a functional of this probability distribution. This, as Shannon showed, turned out to be the case. The uncertainty over the values a source, X, will output is a function of the probability distribution over the allowed symbols and is termed the *entropy* of x

$$\mathcal{H}[p(x)] = -\sum_{\Omega_X} p(x) \log p(x) \tag{1.13}$$

In the continuous case, this is given by the differential entropy

$$\mathcal{H}\left[p(x)\right] = -\int_{\Omega_X} p(x) \log p(x) dx \tag{1.14}$$

Both quantities will be termed entropy. Shannon also showed that the entropy is maximum when the symbols are all equally likely, i.e. each symbol is as informative as the next. In the case of continuous, unbounded quantities with a given variance ('spread'), the entropy is maximum when the distribution is Gaussian (see Appendix A).

In pattern recognition, it is not the information transmitted by each symbol that is of importance, but what patterns in the observations convey about the nature of the information source. As more and more of a sequence is observed, more and more information about the data generation process is (hopefully) gained. The *information gain* is the reduction in uncertainty over which values X can take after one or more instances of X have been observed. This is encoded in a 'model' of the information source, say $\hat{X}$, and its associated distribution over the dictionary of symbols, $p(x|\mathcal{M})$, where $\mathcal{M}$ represents assumptions inherent in the modelling process. Generally speaking, a model prior to any observations will be vague (unstructured) and a posterior model $p(x|\{x_{\text{obs}}\}, \mathcal{M})$ will be more structured, with the information gained a measurement of the difference in their entropies. In this sense, a high entropy distribution is synonymous with low information *content*. For example, a prior model for unbounded and continuous data may assume Gaussian-distributed observations as this can be shown to have the highest entropy for a given variance and is often assumed to be the distribution of unstructured noise. The amount of structure in a sequence of numbers may then be quantified by measuring the non-Gaussianity of their distribution, seeing how much the sequence diverges from random noise. In fact, this is one of the methods that can be used to perform ICA in a bid to find individual components that contain more discernable structure than in the original observation sequence. This will be discussed further in the next Chapter.

1.1.3 Common functions and distributions

Standard functions

If the distribution over random variable X given model parameters θ is denoted $p(x|\theta)$, and is defined over space Ω, then the notation is as follows

$$\langle f(x) \rangle_{p(x|\theta)} \doteq \int_{\Omega} p(x|\theta) f(x) dx \tag{1.15}$$

$$\mathcal{H}\left[p(x)\right] \doteq -\int_{\Omega} p(x|\theta) \log p(x|\theta) dx \tag{1.16}$$

$$KL\left[p(x)\|q(x)\right] \doteq \int_{\Omega} p(x|\theta_p) \log \frac{p(x|\theta_p)}{q(x|\theta_q)} dx \tag{1.17}$$

where (1.15) is the *expectation* of $f(x)$ under $p(x|\theta)$, (1.16) is the entropy of the distribution over x, and (1.17) is the *Kullback-Liebler divergence* (KL-divergence) between distributions $p(x|\theta_p)$ and $q(x|\theta_q)$. The conditioning of the LHS of (1.16) and (1.17) on θ is clear from context and has thus been dropped for brevity. The KL-divergence is also known as the cross-entropy between the constituent distributions and can be rewritten as

$$KL\left[p(x)\|q(x)\right] = -\mathcal{H}\left[p(x|\theta_p)\right] - \left\langle \log q(x|\theta_q) \right\rangle_{p(x|\theta_p)} \tag{1.18}$$

The entropy $\mathcal{H}\left[p(x|\theta_p)\right]$ given here is measured in *natural* bits. The relationship $\log_n x = \frac{\log_e x}{\log_e n}$ can be used to convert to any other units (eg base 2). Where the distribution over x is clear, the entropy of x may be written $\mathcal{H}[x]$; where the variable is clear, the KL-divergence may be written $KL[p\|q]$.

The following standard function definitions are also used

$$\Gamma(a) \doteq \int_0^{\infty} x^{a-1} \exp(-x) dx \tag{1.19}$$

$$\Psi(a) \doteq \frac{\partial}{\partial a} \log \Gamma(a) \tag{1.20}$$

$$\mathrm{erf}(x) \doteq \frac{2}{\sqrt{\pi}} \int_0^{x} \exp(-t^2) dt \tag{1.21}$$

$$\mathrm{erfc}(x) \doteq 1 - \mathrm{erf}(x) \tag{1.22}$$

$$\mathrm{erfcx}(x) \doteq \exp(x^2)\mathrm{erfc}(x); \tag{1.23}$$

where (1.19) is known as the *gamma* function, (1.20) is the *digamma* function, and (1.21), (1.22) and (1.23) are, respectively, the *error* function, the *complementary* error function and the *scaled* complementary error function.

Distributions

The functional forms for the distributions used in this thesis, together with their relevant statistics and measures, can be found in Appendix A. The succinct notation used for each distribution is given below.

A univariate (1D) Gaussian distribution over $-\infty \leq x \leq \infty$ is denoted

$$P(x|m, b) = \mathcal{N}(x; m, b) \tag{1.24}$$

where m is the mean and b is the precision (inverse variance). A multivariate Gaussian is similar, with a vector $\boldsymbol{m}$ representing the mean, and a symmetric matrix $\boldsymbol{b}$ called the precision matrix (inverse covariance matrix). A univariate truncated Gaussian distribution over $0 \leq x \leq \infty$ is denoted

$$P(x|m,b) = \mathcal{N}^{\text{tr}}(x;m,b) \tag{1.25}$$

A univariate Gamma distribution over $0 \leq x \leq \infty$ is denoted

$$P(x|b,c) = \mathcal{G}(x;b,c) \tag{1.26}$$

where b is the width parameter and c is the scale parameter. A multinomial Dirichlet distribution over a vector $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_m\}$ for $0 \leq \pi \leq 1$ is

$$p(\boldsymbol{\pi}|\boldsymbol{\lambda}) = \mathcal{D}(\boldsymbol{\pi};\boldsymbol{\lambda}_{1:m}) \tag{1.27}$$

where $\boldsymbol{\lambda}_{1:m} = \{\lambda_1, \ldots, \lambda_m\}$ and where λ_k is the count for π_k. A univariate Exponential distribution over $0 \leq x \leq \infty$ is denoted

$$p(x|b) = \text{Exp}(x;b) \tag{1.28}$$

where b is the width parameter.

1.2 Scope

The scope of this work is to explore the cutting edge of ICA research. Traditional ICA research focuses on mappings of noiseless observations with the number of independent components equal to the number of dimensions in the data distribution (square decompositions), and where each component is described by the same distribution. ICA has recently been recast as data density modelling, which has increased the flexibility of ICA to find nonsquare decompositions where each component may be described by a different distribution.

This thesis takes up this baton and runs with it by bringing this flexible ICA model into the Bayesian sphere using the variational Bayes method. This Bayesian reformulation will be shown to increase performance and flexibility,

allow the ready incorporation of prior knowledge and imposition of constraints, decompose complex data more effectively by using multiple ICA models, capture temporal information in observations through the use of dynamic extensions, and allow the rigorous comparison of assumptions underlying one's view of the data.

This thesis is not, however, a collection of results from repeated applications of the same ICA algorithm to a range of datasets in the hope of revealing something novel. Such exploration for exploration's sake plays a vital role in science and is, arguably, the driving motivation behind science. In the particular case of pattern recognition, such endeavours widen the appeal and use of intelligent data analyis methods in hitherto virginal territories. The aim of this thesis is to further *develop* ICA by increasing its flexibility and extending its scope such that ICA *becomes* more widely applicable through *principle* rather than through (blind) practice. At every step, the reasoning behind proposed extensions are thoroughly explained to give the reader an understanding of why such extensions are necessary. The advantages they bring are underlined through pertinent demonstrations and illustrative examples. Most results are presented using synthetic data where the underlying patterns and distributions are known, allowing focused and quantifiable evaluation. Some results on real pattern recognition problems are also presented to ensure that blue skies ICA has its feet firmly planted on the *terra firma* of practical application.

1.3 Contributions

There are a number of contributions this thesis makes to ICA research. The fundamental contribution is that of the most complete Bayesian treatment of ICA to date. A Bayesian ICA model has key advantages over non-Bayesian methods, such as a natural tolerance to overlearning, quantifiable model comparison, principled noise modelling, robust parameter estimation, and the ability to incorporate prior knowledge. Real data is noisy, but traditional ICA formulations presume noise-free data and thus perform poorly on noisy data. Inferring the

intrinsic dimensionality of a data distribution is of fundamental interest, but remains difficult in practice. Using ones prior knowledge of a problem domain to 'tune' ICA to particular datasets would allow extra information to be extracted from observations. A Bayesian treatment of ICA allows such outstanding problems to be addressed.

The component distributions sought by ICA are usually assumed to have a particular form. This greatly constrains the type of structure found by ICA in data distributions. The ICA methods developed here utilise a very flexible component or 'source' model using Mixtures of Gaussians (MoG), allowing a wide variety of structure to be captured in detail. This flexibility clears the way for ICA to be applied to a wide variety of previously 'difficult' problems, such as the unmixing of image mixtures.

A further contribution is the construction of a Bayesian Mixture of ICAs model. There has been growing interest in subspace data modelling and representation over the past few years. Methods such as Principal Component Analysis, Factor Analysis and ICA have gained in popularity and have found many applications in image modelling, signal processing and data compression to name just a few. Gaussian-based mixture modelling methods have been proposed using principal and factor analysers to represent different parts of the data space as local Gaussian clusters in order to find more detailed structure. Meaningful representations may be lost, however, if these local manifolds are non-Gaussian or discontinuous. This is remedied by utilising a Mixture of ICAs; localised non-Gaussian manifolds are analysed using different, locally-adapted ICA networks. Discontinuous manifolds can be modelled due to the mixture of Gaussians sources in each ICA network. The Bayesian formulation allows the intrinsic dimensionality of each manifold to be ascertained, and the most likely number of clusters in the data space to be inferred.

Last, but not least, this thesis takes an important step towards fully non-stationary ICA. The vast majority of ICA methods assume each observation vector is independent from the preceding and following observations, thereby

disregarding any temporal information. This assumption is relaxed by incorporating dynamics into ICA using Hidden Markov Models, both within the source models and between ICA networks in a mixture model. This allows temporal information to be captured within the observations and/or within the independent components underlying the observation data. This helps find dynamic changes of state in the observation generating process and increases the robustness of ICA under noise.

1.4 Overview

An exposition must be targeted to a specific audience, otherwise it will lack focus. This is necessarily based on assumptions about the potential audience's prior knowledge and expertise. Theses are no different and generally subscribe to two dicta - assume much, explain little, or assume little, explain much. The inevitable pressures - external and internal - of writing such a work, often coupled with a not-wholly disconnected motivational drought, lead to many following the former philosophy. This thesis follows the latter [3].

The thesis is presented in eight Chapters. Aside from the first and last Chapters, this thesis is essentially split into two parts. Chapters 2-4 cover background theory, explaining the traditionally arcane thesis title in reverse, while Chapters 5-7 derive, describe and demonstrate the eponymous methods.

Independent Component Analysis is the fundamental area of research in this thesis. As such, Chapter 2 is the largest of the theory Chapters and serves as an introduction to ICA. The Chapter explains exactly what ICA is, why it is useful, and - more to the point - why it is difficult in practice. This is followed by a formulation of the problem in a machine learning context. A basic method for ICA is derived and demonstrated. The Chapter concludes with a discussion of the limitations of traditional ICA methods.

The central contribution this thesis makes is in developing the most comprehensive *Bayesian* formulation of ICA to date. What this means and - more

[3] Although the aforementioned hurdles still apply...

importantly - why this is desirable is discussed in Chapter 3 which describes the Bayesian framework for machine learning. Machine learning is pitched as a problem in *modelling* a process that is assumed to have generated the observed data. Bayesian theory is introduced as a consistent way of manipulating beliefs, assumptions and parameters that govern this generative process. The Chapter explains how Bayesian inference can be used to learn models and subsequently select between competing 'explanations' for the observed data. Bayesian machine learning, however, is computationally intractable for all but the simplest of models. The Chapter explores established methods for approximating Bayesian inference. Each method is shown to embody different aspects of the Bayesian framework, but, for varying reasons, all are necessarily rejected for ICA.

Over the last few years, a new formalism has been developed that captures all the advantages of Bayesian modelling while conferring tractability. Chapter 4 describes this *variational* approximation to Bayesian learning. The Chapter starts by deriving the variational approximation from an intuitive basis. A methodology is then constructed for general machine learning. The rest of the Chapter applies the variational Bayesian learning scheme to an otherwise intractable Bayesian mixture of Gaussians model. Key benefits of Bayesian learning are shown to survive this approximation.

The algorithmic methods form the detailed core of the thesis and are covered in Chapters 5-7. Chapter 5 brings ICA into the Bayesian sphere by applying the methodology developed in Chapter 4. The Chapter starts with a short overview of the current cutting-edge of ICA research. This is followed by a detailed description of the proposed model and derivations of two alternative algorithms. These are compared with currently popular ICA methods and are shown to be more flexible and powerful at extracting patterns from noisy, sophisticated data. The two algorithms are then compared, focusing on relative advantages and disadvantages. The Bayesian framework's ability to incorporate prior knowledge is utilised by incorporating automatic structure determination and constraints for non-negative data. The derived model is used to remove unwanted artifacts

from real electrocardiogram signals.

The decomposition and analysis of signal mixtures is the canonical application of ICA. However, as Chapter 6 will show, Bayesian ICA may also be used as the building block of more complex *mixtures* of models that discover self-similar areas in the data space, an important step in discovering clusters and classifying data. In Chapter 6, Gaussian-based mixture models are extended to an Independent Component Analysis mixture model. The model developed in Chapter 5 is used to construct a novel approach for modelling non-Gaussian, discontinuous manifolds. The local dimensionality of each manifold is determined automatically and Bayesian model selection is employed to calculate the optimum number of ICA components needed in the mixture. This sophisticated extension is demonstrated on complex synthetic data and its application to real data is illustrated by decomposing Magnetic Resonance Images into meaningful features.

Chapter 7 develops a dynamic extension of the algorithms developed in Chapters 5 and 6 in order to tap this further source of information. Hidden Markov models (HMMs) are widely used in signal processing for finding dynamic changes of state in the process that is assumed to generate the observations. After a comprehensive introduction to the use of HMMs, Chapter 7 develops a dynamic extension of the ICA model introduced in Chapter 5 by deriving Bayesian ICA with HMM components. This is then turned on its head to produce a Bayesian HMM with ICA observation generators, a dynamic extension of the ICA mixture model of Chapter 6. A combination of these two models is used to find patterns in share indices.

Chapter 8 concludes the thesis with a summary, discussion of outstanding problems, and suggestions for further extensions.

Chapter 2

Independent Component Analysis

Consider a set of M number sequences, for example observations picked up by an array of M sensors, that together constitute a single M-dimensional vector sequence. If these M sequences are generated by a common process or originate from a common source, there may be statistical dependencies between these sequences. Independent Component Analysis is the practice of transforming these M statistically dependent sequences into $L \leq M$ statistically *independent* sequences. The goal of this transformation is to take any information that is spread across these M sensor signals and separate it into L independent streams of information, usually dubbed the latent or *source* signals. The aim is to make subsequent information extraction and analysis much easier. In the parlance of information theory, ICA transforms M mutually informative sensor signals into L exclusively informative source signals.

Another way to look at ICA is the factorisation of a multidimensional distribution into a number of 1-dimensional distributions. Each M-dimensional sensor vector can be plotted as a point in an M-dimensional 'sensor space', so the whole sequence forms a distribution of points in this space. This distribution may contain structure which implies information. Independent Component Analysis is the projecting of this M-dimensional observation sensor space, $\Re^M$, onto an L-dimensional source space, $\Re^L$, described by an independant co-

ordinate frame, i.e. the union of L distinct 1-dimensional spaces[1]. The goal of this projection is to find some coordinate system intrinsic to the data that makes structure within it more cogent. The information encapsulated by the single M-dimensional distribution is repackaged more efficiently into $L \leq M$ independent 1-dimensional *component* distributions and so the resulting distributions may be considered, in effect, a compression of the observations. Although this compression is advantageous at a prosaic level, reducing statistical dependence across informative signals is also motivated by theoretical concerns.

2.1 Why Independence?

What information can be extracted from data depends acutely on how it is represented. Presenting the wanderings of the FTSE 100 over one year as 365 numbers is not very informative; representing the same numbers as points on a line graph is. Humans are very adept at recognising and extracting patterns from raw data, but the way it is represented can greatly influence how efficiently this is done. Therefore, an important step in pattern recognition is to find an efficacious representation.

How this is achieved in practice is, of course, a matter of much research. In the machine learning community, this problem is generally approached from two angles - one mathematical, the other neurological. Mathematically, the problem of representation is one of finding a projection of the data distribution that leads to some sort of 'intrinsic' coordinate system. Neurologically, one wants to represent the data the way the brain does in initial sensory processing. The two are not as incompatible as they sound - both result in a need for independence.

2.1.1 Intrinsic coordinate system

The way data is presented very much influences what patterns can be seen and how much information can be extracted from it. A primary goal in pattern

[1] Although, in principle, complex numbers are permitted, the problem is in its infancy and, as such, only real numbers will be dealt with here.

recognition, then, is to find some intrinsic coordinate system in which the data structure is most apparent. Traditionally, second-order information in the guise of the data covariance structure has been used to construct these coordinate frames, yielding Gaussian-based techniques such as Principal Component Analysis and Factor Analysis. With the increase of computational power over recent years, however, the use of higher-order information has become feasible leading to more sophisticated non-Gaussian methods such as Exploratory Projection Pursuit and Independent Component Analysis.

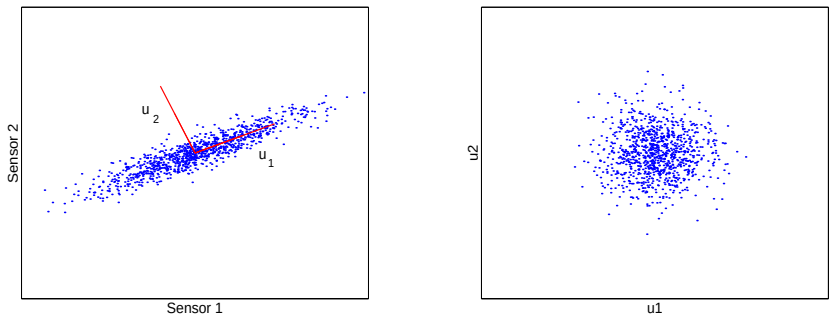
Principal Component Analysis

Principal Component Analysis (PCA) [5, 6] is a widely used statistical tool for representing and compressing data by finding an intrinsic orthogonal coordinate system for the observation data that preserves the maximum amount of variance ('spread' of the distribution) possible. It is used to find a lower-dimensional representation of the data distribution that preserves dominant features within the data distribution while discarding less important ones. PCA is a linear projection that seeks an orthonormal decorrelating source space (or basis) for the data subject to the criterion that the mean-square error between the original data and the data reconstructed using this (lower-dimensional) representation is minimised [6]. Consequently, PCA is a second-order method which uses data covariance information to find orthogonal directions of maximum variance, and therefore (implicitly) assumes Gaussian distributed data. If $\boldsymbol{X}$ denotes an $M \times N$ data matrix and $\boldsymbol{R}_{\boldsymbol{X}} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T$ its sample covariance matrix, then directions of variance can be found by either a simple eigenvector decomposition of $\boldsymbol{R}_{\boldsymbol{X}}$ or by a singular value decomposition of the original (zero-mean) data

$$\boldsymbol{R}_{\boldsymbol{X}} = \frac{1}{N}\boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^T \tag{2.1}$$

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{2.2}$$

where $\boldsymbol{U}$ is an $M \times M$ orthonormal matrix, $\boldsymbol{V}$ is an $M \times N$ orthonormal matrix and $\boldsymbol{\Sigma}$ is a positive diagonal $M \times M$ matrix.



(a) PCA axes

(b) PCA representation

Figure 2.1: Principal Component Analysis.

The columns of $\boldsymbol{U}$ are the eigenvectors of the covariance matrix $\boldsymbol{R_X}$ and are known as the *principal components* of the data $\boldsymbol{X}$. The columns of $\boldsymbol{V}$ are the projections of the columns of $\boldsymbol{X}$ in the coordinate frame described by $\boldsymbol{U}$. The diagonal entries of $\boldsymbol{\Sigma}$ are the singular values of $\boldsymbol{X}$ while the diagonal entries of $\frac{1}{N}\boldsymbol{\Sigma}^2$ are the eigenvalues of the covariance matrix and quantify how much variance there is along the corresponding eigenvector. These eigenvalues are arranged in descending order and the columns of $\boldsymbol{U}$ are permuted accordingly such that the first principal component lies in the direction of maximum variance, the second in an orthogonal direction with the second most variance etc (Figure 2.1(a)).

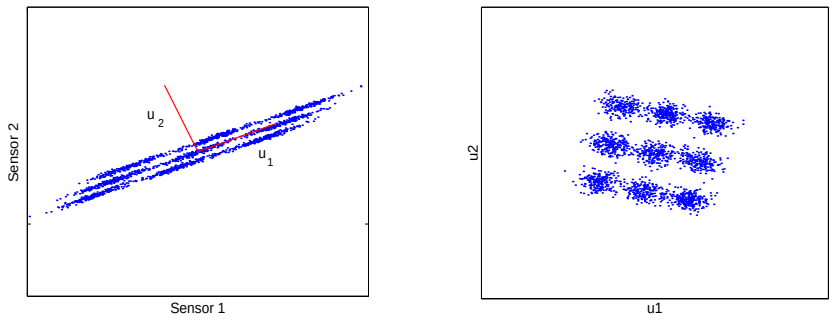
The matrix $\boldsymbol{W} = \boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T$ is the PCA transformation that projects the observation data on to a source space that decorrelates them and scales the projections to have equal variance (Figure 2.1(b)). For Gaussian distributed data decorrelation is equivalent to independence, so PCA can be seen as a special case of finding the independent components of (noise-free) Gaussian data. These components, however, cannot be uniquely identified. PCA projection leads to a source distribution that has equal variance in all directions and is therefore spherical. Consequently, the source distribution can be rotated by any orthogonal matrix leaving the representation unchanged. PCA defines a particular orientation by ordering the principal components in descending order

of variance explained. This ordering, though, may not be appropriate for certain applications. For example, if the observations in Figure 2.1(a) were originally generated by mixing two independent Gaussian-distributed signals, and the aim is to identify these signals, then these signals cannot be recovered by PCA (and - by extension - ICA) although the mixing can.

As well as decorrelating signals, PCA can also be used as a tool for compression. If most of the variance is explained by the first L entries in $\mathbf{\Sigma}$ then the last $M - L$ entries can be regarded as noise and discarded with little information loss. By choosing the first L columns of $\boldsymbol{U}$, PCA projection can be used in finding an efficient lower-dimensional representation of the data, $\boldsymbol{S} = \boldsymbol{U}^T\boldsymbol{X}$. It can be shown [1] that such an L-dimensional representation minimises the mean-square error in reconstructing the data, $\hat{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{S}$. For example, projecting the distribution in Figure 2.1(a) onto the first principal component u_1 gives the most faithful 1-dimensional representation possible in a Euclidean sense.

As the straight forward decompositions in (2.2) do not discriminate between noise and genuine data, alternative methods must be used when the observation data are noisy. Probabilistic PCA [7] extends the PCA formalism by modelling isotropic Gaussian noise while Factor Analysis [8] drops the isotropic constraint. These methods essentially extract an estimate of the noise covariance from the sample covariance then compute the principal components of the 'clean' data. The inclusion of noise prevents a closed form solution, so noisy PCA is usually achieved via iterative algorithms rather than matrix decompositions.

The central problem with Principal Component Analysis is that it views the world through a Gaussian lens - structure not described by the sample covariance, such as clustering or decay of the data distribution's 'tails', may not be preserved correctly in the projection. PCA is a method for factorising Gaussian distributions by finding an orthogonal coordinate frame to diagonalise the covariance matrix. The motivating principle is to represent the data as accurately as possible in a Euclidean sense, not the preservation of structure. Interesting signals are non-Gaussian almost by nature, but for many non-Gaussian distri-



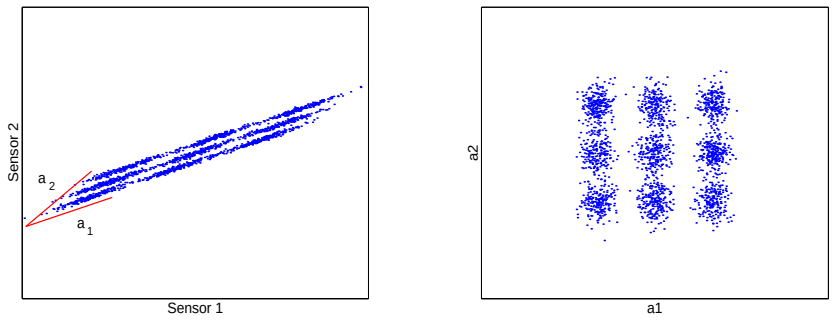
(a) PCA axes

(b) PCA representation

Figure 2.2: Principal Component Analysis of non-Gaussian data.

butions a natural coordinate frame is not necessarily an orthogonal one built on Euclidean distances. Figure 2.2(a) shows a non-Gaussian density with a similar covariance to the Gaussian in Figure 2.1(a). As far as PCA is concerned, it 'sees' a Gaussian and treats it as such, only factorising up to the second moment. Simply decorrelating the non-Gaussian density leads to the representation in Figure 2.2(b). The data density has been 'unwarped' with equal variance in both directions. Although this representation is more favourable, projection onto one or other of the axes loses much of the clustering structure. A more natural coordinate frame would be one in which the new representation also 'unrotated' the source distribution in Figure 2.2(b) by factoring all the statistical moments (mean, covariance, skewness, kurtosis etc.). Factoring second-order moments can only 'unwarp' the density, so factoring non-Gaussian densities is not possible with second-order information alone.

The separation of all statistical moments leads to a representation based on independence rather than just decorrelation. Figure 2.3 shows the axes and resulting source density using independent axes. By dropping the orthogonality constraint, a more natural representation is achieved. Projection onto the independent axes now preserves the clustering structure.



(a) ICA axes

(b) ICA representation

Figure 2.3: Independent Component Analysis.

Exploratory Projection Pursuit

Another method that uses higher-order information for data representation - and one intimately linked with ICA - is *Exploratory Projection Pursuit* (EPP) [9, 10, 11]. The aim of EPP is to find a low-dimensional (typically of order 2 or 3) projection of the observation data that aids the visualisation of structure not encapsulated in the data covariance, such as clustering and skewness. A 'pursuit index' is assigned to every projection that quantifies the 'interestingness' of the structure found. The projections are then optimised numerically to maximise this index.

The 'interestingness' of a particular direction in the distribution is very much a task-dependent property, but it can be argued [9, 11] that the Gaussian is the least interesting of all distributions. As discussed above, Gaussians are completely described by second-order statistics therefore display no structure beyond covariance. Furthermore, for fixed variance, the Gaussian has the highest entropy of all unbounded distributions so can be said to encapsulate the least amount of information. Therefore, the notion of 'non-Gaussianity' can form the basis of constructing an index. Any quantity that measures the non-Gaussianity of a projection can serve as measurement of 'interestingness'. Such measures include functions of statistical moments, cumulants [12] and negentropy (see

section 2.3.3). Which measure one uses depends on the structure sought for. Statistical moment based indices measure non-Gaussianity in the main body of the distribution so are useful for finding clusters [10]. Cumulants and negentropy measure non-Gaussianity based on the tails of the distribution and are better at highlighting skewness [9].

Different projections of the data may find different structures, so the EPP procedure is usually iterated a number of times. After each index maximisation, the direction found is either extracted or 'Gaussianated' [10] and the EPP process repeated to find further directions. This continues until no more directions remain. The set of projections found form a collection of directions or components each containing separate structural information. It can be shown [13] that, in the low observation-noise limit, these components are in fact the independent components of the distribution. Indeed, ICA algorithms can be derived by using non-Gaussiantity as a measure of independence [14, 15, 16] (see section 2.3.3).

Although a far more sophisticated tool than second-order methods such as PCA, EPP can be limited in application. Measuring non-Gaussianity utilising the above quantities directly is computationally expensive so approximations are usually made. These approximations have the effect of smoothing the data distribution, so subtle structure or structure close to Gaussianity is often lost. There is no explicit way of modelling observation noise, so noisy data can cause spurious directions to be found. EPP algorithms based on cumulants are also sensitive to distribution outliers so can be less efficient with noisy data. The effectiveness of EPP decreases with increasing projection dimensionality due to the computational cost of computing the pursuit indices and, in particular, performing the structure removal step [10]. As discussed above, the set of 'interesting $\equiv$ non-Gaussian' components is equivalent to the set of independent components. Independence can be quantified in a variety of ways (see section 2.3.3), not just by measuring non-Gaussianity. Therefore, finding projections using *independence* as the criterion of interest can overcome many of these prob-

lems. Furthermore, maximising an objective functions measuring independence obviates the need for a removal step as all the independent directions can be found simultaneously.

2.1.2 Neurological processing

An alternative to a purely mathematical view is a physiological one. The brain is incredibly powerful at analysing and extracting structure from raw sensory data. This data is processed in many different ways and on many different levels by many different areas. The arcana of the brain are manifold, but exciting work on the neocortex has led to two candidate paradigms in the initial stages of sensory processing - redundancy reduction and sparse coding.

Redundancy reduction

It has been argued by Barlow [17, 18, 19] and Attick and Redlich [20] amongst others [21, 22, 23] that a primary goal of early sensory processing in the neocortex is to reduce redundancy across input signals. The world is a highly regular place that obeys the strict laws of physics, so the signals and stimuli that nature generates have high statistical structure. This means the space of all *probable* input patterns is a very small subset of the space of all *possible* patterns. Redundancy reduction seeks to find a representation that concentrates its descriptive power on the region of probable inputs only, rather than the whole input space.

For example, natural scenes are very similar across sections of the image, with groups of pixels displaying similar colours and textures, gradual changes of contrast except at edges etc. This means that the values of many pixels can be predicted from the values of surrounding pixels, thereby making much of the image signal redundant. In fact, this is one of the principles used in MPEG compression to compress digital television signals by more than 90 percent [24]. Sensory systems that take advantage of this structure to reduce processing overheads clearly have an ecological and evolutionary advantage. The limited dynamic range of neuronal responses coupled with the relatively small amount of information that can be analysed at higher sensory levels [25] means that sen-

sory signals have to be highly compressed if they are to be processed quickly. This implies that the primary goal of the neocortex is to find *compact* codes that best represent the sensory inputs.

The amount of redundancy in an L-dimensional signal, $\boldsymbol{S} = \{s_1, ..., s_L\}$, is quantified by the *Shannon redundancy* [4], R

$$R = 1 - \frac{\mathcal{H}[\boldsymbol{S}]}{C} \tag{2.3}$$

where $\mathcal{H}[\boldsymbol{S}]$ is the entropy of $\boldsymbol{S}$ and C is the *channel capacity* [4]. If the signals in $\boldsymbol{S}$ are represented using some alphabet of symbols, $\mathcal{A}$, then C is the logarithm of the total number of L-dimensional messages possible using the symbols of $\mathcal{A}$. If there are N symbols in $\mathcal{A}$ then there are N^L possible messages and $C = L \log N$. The entropy $\mathcal{H}[\boldsymbol{S}] \leq C$ is a measure of how many messages are actually ever produced so $0 \leq R \leq 1$ measures how efficiently $\boldsymbol{S}$ uses $\mathcal{A}$.

The sources of redundancy are clear if (2.3) is re-written as

$$R = \frac{1}{C}\left[\left(C - \sum_{i=1}^{L} \mathcal{H}[s_i]\right) + \left(\sum_{i=1}^{L} \mathcal{H}[s_i] - \mathcal{H}[\boldsymbol{S}]\right)\right] \tag{2.4}$$

The first term in the brackets is the redundancy due to unequal use of the alphabet $\mathcal{A}$. If the N symbols in $\mathcal{A}$ are used equally by each of the signals, then $\mathcal{H}[s_i] = \sum_{|\mathcal{A}|} \frac{1}{N} \log N = \log N$ and the first term disappears. The second term is the mutual information of $\boldsymbol{S}$, $I[\boldsymbol{S}]$, where $I[\boldsymbol{S}] \geq 0$ with equality if and only if the signals are statistically independent i.e. $p(\boldsymbol{S}) = \prod_i p(s_i)$. This is the redundancy due to dependencies amongst the L individual signals. The mutual information measures the amount of information 'repeated' across the signals and results from intersignal and intersymbol dependencies. These dependencies may be between pairs of of symbols, between multiple symbols, consecutive or non-consecutive symbols, from complicated interactions between signal streams etc. Whereas unequal symbol usage is a 'single' source of redundancy, mutual information measures redundancy from a multitude of dependencies and thus often makes up the highest portion of the total redundancy [25], particularly in natural stimuli due to the regularities introduced by the laws of physics.

Minimising R involves finding a new code or representatation, $\mathcal{B}$, that minimises the two redundancy contributions in (2.4). Codes that minimise R are called minimum-redundancy codes while those that minimise the mutual information are dubbed factorial or minimum-entropy codes [26] and have the effect of making the L signals as independent as possible. Minimising the mutual information while preserving the total information communicated by $\boldsymbol{S}$ (i.e. keeping $\mathcal{H}[\boldsymbol{S}]$ fixed) is equivalent to minimising the individual entropies, $\mathcal{H}[s_i]$. This may not reduce R itself as this minimisation increases the first term in (2.4) - the use of alphabet symbols becomes more unequal and so the individual signals become more structured.

It has been suggested by Barlow [18] and Attick [25] that finding codes that minimise the mutual information are useful on three fronts. Firstly, these codes are a useful first-step in minimising redundancy as a whole. Reducing unequal symbol usage is relatively trivial and as mutual information is often the most significant contribution to redundancy in natural signals, transferring this redundancy to symbol usage makes redundancy reduction much easier. Secondly, Barlow argues that associative memory needs access to prior probabilities of events if it is to recognise whether two events, s_1 and s_2, are unrelated occurances or that one can predict the other. Multiplying the prior probabilities of s_1 and s_2 gives the probability of the two events occuring by chance. If one event is followed by the other with a higher frequency, then a non-random association (it is argued) is learnt. The set of probabilities required grows exponentially with the number of events and all possible combinations of events and would quickly exceed the storage capacity of the brain if stored in raw from. Barlow suggests that the most efficient way of calculating these probabilities is if each individual event is considered *independent* from each other event *a priori*. The probability of any conjuction of N events is then straight forward to calculate needing only the N independent probabilities. This implies that factorial (independent) coding may be necessary for such cognitive tasks. Finally, increasing the structure within the individual signals makes them more informative and

any intrinsic patterns more cogent. This increases the speed and efficiency of higher-level cognitive tasks such as pattern recognition, increasing the chances of survival in a big, bad world.

Sparse representation

A related topic to redundancy reduction is *sparsity* in representation. According to Field [27], the main goal of the neocortex is not necessarily to reduce redundancy overall, but to just reduce the statistical dependence between input signals. This is equivalent to finding factorial codes that reduce the mutual information term in (2.4). Field argues that the aim of early sensory processing is not to represent the data with the minimum number of sensory units but with the minimum number of *active* units. Rather than constructing compact codes, the primary goal of the neocortex is to find *sparse* codes that best represent the sensory inputs. A compact code represents sensory input with the minimum number of units. This means that every M-dimensional input pattern has an $L \ll M$ compact pattern associated with it that uses all the L units. In contrast, a sparse code has $L \simeq M$ units but of which only a small subset are ever activated simultaneously. Although the difference between the two schemes may seem minor, the consequences run deeper.

Each sensory unit has a responsive field and response pattern associated with it. The responsive field is the section of the data distribution that a particular sensory unit is responsible for. The response pattern is the distribution of responses each unit gives for the collection of input patterns. A unit that is part of a compact coding scheme has responsibility for the whole region covered by the input data, as do its siblings in the coding. Each unit contributes to representing every input pattern, although how much it contributes depends on its response distribution. As each unit must contribute to every pattern, its response distribution must cover the whole data distribution, so the difference between high and low responses is relatively small.

A unit that is part of a sparse coding scheme, however, has responsibility

for mainly a small region covered by the input data. Each unit is responsible for a slightly different portion of the input distribution, with enough overlap to cover the region of probable inputs comprehensively. This means only a subset of units contribute to representing each input pattern. As each unit now only fires when presented with a pattern under its jurisdiction, the response distribution of each unit is more highly peaked, so the difference between high and low responses is much larger. This has a number of benefits over a compact coding scheme. When a unit does fire in response to a pattern, its output is much higher than under a compact coding scheme. This greatly improves the signal-to-noise ratio of the unit's output, leading to cleaner signals for the next level of processing. More importantly, high response signals the presence of a feature with greater probability. Less equivocal responses aid feature detection and make pattern recognition more robust. Furthermore, there is evidence [27] that networks that model associative memory work more efficiently if presented with sparse inputs. This implies that sparsity is necessary for efficient analyses further up the sensory pathway.

Most intriguingly, Field showed that natural images cannot be properly represented by PCA, a compact code. PCA codifies correlation information only, and this can be shown to only capture the amplitude spectra of scenes [27]. Field argued that the distribution of natural images cannot be Gaussian as localised features, edges, lines etc. can not be captured in the amplitude spectrum only - phase information is also needed. This means the structure typical of natural scenes can only be captured in higher-order moments. One moment of interest is the kurtosis which governs the decay of the tails of a distribution. Field presented empirical evidence that suggested the distribution of natural scenes could be factorised into distributions with high kurtosis (i.e. highly peaked) and, therefore, could be represented more effectively by sparse rather than compact codes.

Sparse codes convert redundancy between signals into redundancy within signals. If the redundancy between signals is removed completely, then the

sparse code is a factorial code. In other words, sparse coding makes the input signals as independent as possible.

2.2 Applications

The ability to recast multidimensional data as a sum of independent one-dimensional data lends itself to many problems. The canonical application is *Blind Source Separation* (BSS), the problem that originally led to the development of ICA [28]. A set of L signals, $\boldsymbol{s} = \{s_1, ..., s_L\}$, are mixed to produce a set of M signals, $\boldsymbol{x} = \{x_1, ..., x_M\}$. BSS is the recovery of the L underlying signals given only the mixtures, $\boldsymbol{x}$, and with minimum assumptions about the mixing process or the source signal statistics. The classic example is the 'cocktail party' problem where an array of M microphones picks up mixtures of L people talking. BSS attempts to separate the cacophony into the individual voices - something the human brain finds relatively easy. ICA is now the *de jure* basis of modern BSS research and has been applied to a bewildering array of BSS problems including the processing of biomedical signals (EEG, MEG etc.) [29, 30] and the analyses of financial time series [31, 32] as well as the obvious auditory applications [33].

There is much interest in the information-theoretic coding of images as a basis for sophisticated compression, an efficacious representation in image recognition and, intriguingly, as a model for neurological representation. This has led to interesting work in face recognition [34, 35] and the analysis of natural scenes [36, 37]. The latter in particular has led to debate on parallels between ICA and neocortical coding [38, 39]. Other diverse applications include the representation of text corpora [40, 41], removing intereference in wireless telecommunications [42, 43] and the analysis of ground reflectances in remote sensing [44].

2.3 The ICA Problem

Independent Component Analysis is really a two-stage process. The first involves finding a relationship, Υ, between the sensor space, $\Re^M$, and an[2] independent source space, $\Re^L$; this is the *learning* process. This relationship is usually considered linear for identifiability and computational reasons, and will be considered linear in this thesis. Some measure or 'contrast function', ψ, is formulated that can quantify the independence of a space given some representative observation data and candidate relation $\Upsilon = \Upsilon'$. Ideally, this measure would be convex with respect to independence such that it would be at an extremum if Υ' gave an independent source space. However, due to inherent ambiguities in the ICA procedure (see section 2.5.1), the contrast function can only be locally convex. The search for an optimal Υ is then linked to ψ such that - when extremised - an independent source space is found. This involves adjusting Υ, but may also involve adjusting the characteristics of $\Re^L$. The training data used during this process is usually a subset of the data to be projected. The second stage uses the learnt relationship to project the observations onto the independent coordinate frame - the *mapping* process. At first sight, the mapping may seem trivial given the results of the learning process. However, the way in which the ICA problem is formulated has direct bearing on the relationship between the first and second stages of the ICA process.

The whole ICA objective reduces to two central issues:

- What form Υ should take.
- How to formulate an appropriate measure, ψ.

The choice of one generally influences the choice of the other. The problem has been approached from a number of different angles, from constructing neuromimetic architectures [28], through utilising information theoretic principles to maximise information transmission [45], to constructing probabilistic models

[2]The word 'an' implies that this space is not necessarily unique - see section 2.5.1 for more details.

[46]. However the problem has been tackled, ICA research has generally fallen into two camps - ICA formulated as either a *mapping* problem or a *modelling* problem.

2.3.1 ICA as a mapping problem

The traditional approach considers ICA as a mapping problem, usually based on neurological concerns. The aim is to find some mapping $\Upsilon \doteq \boldsymbol{g} : \Re^M \mapsto \Re^L$ that makes the L signals as statistically independent as possible. During the learning process, the parameters that govern $\boldsymbol{g}$ are adjusted such that the output L signals meet the criterion set by ψ. The mapping process is then simply a case of applying this learnt function directly to the observation dataset. This approach is often taken when considering information-theoretic measures such as mutual information. These quantities are, conceptually, the most appealing as they fundamentally measure information content in a sequence of numbers. The mapping approach is the simplest way of extremising these quantities.

Information-theoretic measures can define independence precisely, but they are notoriously difficult to measure [47], so approximations usually have to be made. These approximations can restrict the types of sequences that can be analysed - for example, those with multi-modal distributions - and may lead to sub-optimal projections (see, for example, [45] for a discussion of pathological cases). Also, although formulating ICA in this way makes the mapping stage trivial, there is no simple way of taking (potential) noise in the observations into account. As such, noisy training data will result in learning a mapping that is sub-optimal and so poor at transforming the observation dataset. Furthermore, if the observations were originally generated by mixing independent source signals, this mapping may be identifiable, but its inverse may not, particularly if $M \neq L$. Consequently, the mapping approach has usually been applied to the noise-less, square-mixing ($L = M$) case.

2.3.2 ICA as a modelling problem

More recently, the *generative modelling* approach has come to the fore, mostly from the desire for an intrinsic coordinate frame. In this formalism, the M observation sequences are considered generated by mixing L independent source signals. ICA is now recast as a modelling problem. A parametric model is constructed that mimics the forward mixing process of L source signals into M observation signals: $\Upsilon \doteq \boldsymbol{f} : \Re^L \mapsto \Re^M$. During the learning process, the hypothesised model's parameters are adjusted so as to maximise the likelihood of the model generating the observed data (the measure ψ). During the mapping process, the model parameters are frozen and observation vectors are 'clamped' to the model's output, one by one. The model is then effectively run 'in reverse' to find the most likely (hidden or *latent*) source vector that generated each data vector clamped to the output. Model-based likelihood measures - in the guise of probabilities - are easier to measure and manipulate than (model-free) information-theoretic measures. Although both are intimately linked, probabilities are straight forward to calculate once the model has been specified while information-theoretic quantities have to be *derived* from the probability distributions. Probabilities can also be manipulated in a rigorous and consistent way under the axioms of probability theory.

The learning process is usually slower than the traditional mapping approach as there are generally many more parameters to estimate. The mapping stage is also more cumbersome as source vectors have to be inferred. This approach does, however, allow non-square and noisy mixing to be naturally accomodated as part of the model's parameters. Furthermore, any prior information one has about the processes that generated the data (be it physical or simply theoretically appealing) can be incorporated into the model, something not easily possible under a pure mapping approach. This can allow, for example, multi-modal data densities to be factored, constraints on the mixing process to be enforced, etc.

2.3.3 Measuring independence

Whether one chooses to follow the direct-mapping or modelling approach, some kind of measure, ψ, is needed to quantify the independence between signals. Before candidate measures are explored, recall the definition of independence. A set of L random variables $\boldsymbol{s} = \{s_1, ..., s_L\}$ are deemed statistically independent if and only if (iff) their joint density factorises such that

$$p(\boldsymbol{s}) = \prod_{i=1}^{L} p_i(s_i) \tag{2.5}$$

Consequently, any measure of independence will be a measure on a distribution.

Mutual information

A measure of the inequality between the LHS and RHS of (2.5) is a measure of the dependence amongst the components of $\boldsymbol{s}$. Consider the following quantity

$$KL\left[p_1(\boldsymbol{s}) \| p_2(\boldsymbol{s})\right] \doteq \int p_1(\boldsymbol{s}) \log \frac{p_1(\boldsymbol{s})}{p_2(\boldsymbol{s})} d\boldsymbol{s} \tag{2.6}$$

Equation (2.6) defines the Kullback-Leibler divergence [48] between distributions $p_1(\boldsymbol{s})$ and $p_2(\boldsymbol{s})$. Due to the convexity of logarithms, the KL-divergence satisfies

$$KL\left[p_1(\boldsymbol{s}) \| p_2(\boldsymbol{s})\right] \geq 0 \tag{2.7}$$

with equality iff $p_1(\boldsymbol{s}) = p_2(\boldsymbol{s})$. Substituting (2.5) into (2.6) gives

$$\begin{aligned} KL\left[p(\boldsymbol{s}) \| \prod_{i=1}^{L} p_i(s_i)\right] &= \int p(\boldsymbol{s}) \log \frac{p(\boldsymbol{s})}{\prod_{i=1}^{L} p(s_i)} d\boldsymbol{s} \\ &= \int p(\boldsymbol{s}) \log p(\boldsymbol{s}) d\boldsymbol{s} - \sum_{i=1}^{L} \int p(s_i) \log p(s_i) ds_i \\ &= \sum_{i=1}^{L} \mathcal{H}[s_i] - \mathcal{H}[\boldsymbol{s}] \\ &= I[\boldsymbol{s}] \end{aligned} \tag{2.8}$$

where $d\boldsymbol{s} = \prod_i ds_i$. The term $I[\boldsymbol{s}]$ is the mutual information of $\boldsymbol{s}$ and measures the amount of information shared across the sources (i.e. components).

Equating (2.7) with (2.8) reveals the intuitive notion that the lower the mutual information of $\boldsymbol{s}$, the greater the independence between the sources, with equality only when (2.5) holds.

Mutual information is generally considered the most intuitive measure of the independence between signals. Its use in ICA was first proposed by Comon [14] and simply states that, if signals are independent, information extracted from one tells you nothing about information contained in another. The problem with this quantity is that it is difficult to calculate [47], especially as $p(\boldsymbol{s})$ is not known. Estimating the joint density $p(\boldsymbol{s})$ is difficult without large amounts of data, particularly if the number of components is large. In practice, approximations are made [14], which may break the convexity of the measure w.r.t. independence, or other quantities are used that can measure independence.

Entropy

It was noted by Nadal and Parga [49] that mappings that maximise the information transmitted between two spaces, $\Re^M \mapsto \Re^L$, were capable of producing factored densities. This leads to a popular measure - the entropy of the source space. The amount of information shared by two random variables is given by

$$\begin{aligned} I[\boldsymbol{x};\boldsymbol{s}] &= \mathcal{H}[\boldsymbol{s}] - \mathcal{H}[\boldsymbol{s}|\boldsymbol{x}] \\ &= \sum_{i=1}^{L} \mathcal{H}[\boldsymbol{s_i}] - I[\boldsymbol{s}] - \mathcal{H}[\boldsymbol{s}|\boldsymbol{x}] \end{aligned} \tag{2.9}$$

using (2.8). The last term on the RHS can be ignored if the variable $\boldsymbol{x}$ is noise-free and the mapping from $\boldsymbol{x} \mapsto \boldsymbol{s}$ is deterministic as there are no other conduits for information. The mutual information term on the RHS is still difficult to calculate, but by maximising the component entropies $\mathcal{H}[s_i]$ one hopes to indirectly minimise the mutual information $I[\boldsymbol{s}]$.

As the entropies are functions of single variables, this maximisation is relatively straight forward. Although, in general, $I[\boldsymbol{s}]$ is simultaneously minimised, the entropy measure is not strictly convex w.r.t. independence. Under some circumstances [45], the mutual information on the RHS can actually *increase*, leading to more - rather than less - dependence across components.

Statistical moments

If the distribution over the source space factors according to (2.5), then so do the moments of $\boldsymbol{s}$. Independence is a strong statement so the first and second moments have little factoring power; the first measures position while the second can discriminate only up to correlation. The third moment - skewness - is similarly impoverished as its descriptive prowess is limited to which side of the expectation has the greater mass. It's not until the fourth moment is reached that there is enough information to test independence. More precisely, it is the fourth *cumulant*, $C_{ijkl}[\boldsymbol{s}]$, that is used [12]. This is a statistical measure based on the log fourier transform of a probability distribution. For a single variable, this is known as the *kurtosis*, $\text{kurt}[s_i]$, and is defined

$$\text{kurt}[s_i] \doteq \langle (s_i^4) \rangle - 3\langle (s_i^2) \rangle^2 \tag{2.10}$$

where $\langle . \rangle$ is the expectation operator under the pdf of s_i. The second term is the fourth moment of a Gaussian, thus the kurtosis is also a measure of non-Gaussianity - kurtosis is zero for Gaussians.

When random variables are independent, the fourth cumulant simplifies to a sum of their individual kurtoses. An independence measure based on pdf moments is then given by

$$\psi \doteq \sum_{ijkl} \left(C_{ijkl}[\boldsymbol{s}] - \text{kurt}[s_i]\delta_{ijkl} \right)^2 \tag{2.11}$$

This measure is zero iff the component variables, $\{s_i\}$, are independent.

When higher-order moments are evaluated using samples from a pdf (as in EPP), they are highly sensitive to outliers. In other words, non-representative samples, for example noisy data or samples from the tails, corrupt estimates of the fourth-order statistics. Consequently, such measures - although fast - are not very robust with small datasets or in the presence of noise. Furthermore, empirical studies in [50] indicate that fourth-order information is not enough to separate more than a handful of sources.

Negentropy

Something that kurtosis highlights is the relationship between ICA factorisation and non-Gaussianity. This relationship can be better understood by invoking the Central Limit Theorem (CLT) [51]. According to the CLT, mixing a large number of independent random variables makes the resultant distribution more Gaussian than either of the original distributions. Hence, the density over observations can be considered more Gaussian than the desired factored distributions. Driving these densities away from Gaussianity, therefore, provides a method of finding a factored form.

An information-theoretic quantity that measures the deviation of a distribution from a Gaussian with the same covariance is the *negentropy*, $J[\boldsymbol{s}]$.

$$J[\boldsymbol{s}] \doteq \mathcal{H}_{\mathcal{N}}[\boldsymbol{s}] - \mathcal{H}[\boldsymbol{s}] \tag{2.12}$$

where $\mathcal{H}_{\mathcal{N}}[.]$ is the entropy of a Gaussian with the same covariance as $\boldsymbol{s}$. The negentropy and mutual information are related by

$$I[\boldsymbol{s}] = J[\boldsymbol{s}] - \sum_{i=1}^{L} J[s_i] + \frac{1}{2} \log \frac{\det[\mathrm{diag}(\boldsymbol{R})]}{\det(\boldsymbol{R})} \tag{2.13}$$

where $\boldsymbol{R}$ is the covariance of $\boldsymbol{s}$. Comon [14] showed that if the observation data are preprocessed by PCA to decorrelate the M sequences (known as pre-whitening), maximising the non-Gaussianity of the source signals is equivalent to minimising the mutual information between them.

The most fundamental problem in using non-Gaussianity as an independence measure is that the CLT is only valid for a large number of variables. For a few variables (less than ten), the pdf of the mixture may still be sufficiently non-Gaussian to limit the effectiveness of negentropy. Also, if the original source pdfs were close to Gaussian they may not be correctly estimated. Furthermore, as with mutual information, negentropy is difficult to evaluate in practice. Measures based on negentropy usually resort to approximations, for example based on cumulants. A straight forward cumulant expansion leads to a kurtosis-based approach similar to above, with the associated lack of robustness. Rather more

sophisticated expansions replace the expectations of moments with expectations of general nonquadratic functions [52]. Although more robust than direct moment based measures, they rely on some arbitrary choice of functions which can greatly effect the accuracy and desired convexity of the negentropy approximation.

Likelihood

In the generative modelling approach, one wishes to maximise the likelihood of the data under a constructed model (or, equivalently, maximise the log-likelihood). This likelihood is the probability that some model, $\mathcal{M}$, generates the observed vector sequence, $\boldsymbol{X}$, given the model parameters, $\boldsymbol{\Theta}_{\mathcal{M}}$

$$\psi \doteq \log p(\boldsymbol{X}|\boldsymbol{\Theta}_{\mathcal{M}}, \mathcal{M}) \tag{2.14}$$

The measuring problem is now turned on its head as the measure is now on the data space, $\Re^M$. A (difficult) direct measure of the independence of space $\Re^L$ is no longer necessary as independence is hard-wired into the model. The measure is maximised by adjusting the forward mapping - and perhaps the source space - under the constraint that *the source space is independent by definition of the model*. By including explicit models for a noise process, the likelihood is also very robust to missing or spurious data and can be shown to be asymptotically consistent (parameter/source estimates approach the 'correct' values as the number of data samples increases) and asymptotically efficient (the variance or 'error' in the parameter/source estimates approaches a theoretical minimum - the Cramer-Rao bound [53] - as the number of data samples increases). Furthermore, any other information - details of the forward mapping process, known or desired characteristics of the source space - can also be coded into the model allowing the ICA projection to be tailored to specific problems.

To construct a true likelihood from generative models, assumptions must be made about the densities in the source space and coded into the model. Also, although more straight forward to calculate than the measures presented above, it can still be a computationally demanding process if the model is detailed and

sophisticated, particularly if using a Bayesian framework for model construction. In such cases, approximations can be formulated to expedite the calculations without jeapordising the convexity of the measure.

2.4 History

Independent Component Analysis is (at the time of writing) 16 years young. Although its history is short, ICA research has been very active and has produced a body of work that belies its youth.

2.4.1 Mapping approach

Independent Component Analysis was first formulated in 1986 by Herault and Jutten [28] in an attempt to solve the BSS problem in signal processing. In line with established BSS research, Herault and Jutten assumed square (number sources = number sensors), instantaneous, linear mixing and used a recursive artificial neural network (ANN) to estimate an inverse mapping. Unlike previous methods, the crucial step in evaluating this mapping was to assume that the underlying signals were *independent* of each other. In other words, they suggested the BSS problem could be addressed by forcing the data towards independence and thus, Independent Component Analysis (in all but name) was born. The mapping were subsequently learnt using a Hebbian-like learning rule [54]. This rule employed non-linear functions of the outputs to evaluate the independence amongst pairs of outputs such that a product of these functions disappeared when independence was achieved.

This original, but rather sketchy work was put on a sound theoretical basis by subsequent papers in 1991 by Jutten and Herault [55] and Comon *et al* [56]. The former first coined the term 'Independent Component Analysis' - in analogue with Principal Component Analysis - while the latter analysed the mathematical foundations underpinning this new procedure. ICA was formally defined in 1994 by Comon [14], in which he proposed mutual information as the most natural measure (ψ) of independence, albeit a difficult one in practice. In

the same paper, he derived an approximation to the mutual information based on Edgeworth expansions [57] in terms of cumulants. Similar expansions have been proposed by Amari *et al* [58], while algebraic methods using cumulants have been explored by Cardoso and Comon [59] and by Cardoso [60].

It was also shown by Comon [14] that the negentropy could be used as a proxy for the mutual information. Comon showed that maximising the non-Gaussianity of the source signals was equivalent to minimising the mutual information between them. An approximation to this measure was used in a nonlinear PCA implementation of ICA by Karhunen and Joutensalo [61], Oja [62], and by Hyvärinen and Oja [15] in their FastICA algoritm. Girolami also used negentropy approximations in projection pursuit formulations of ICA [13].

In [22], Linsker showed that linear mappings of Gaussian densities that maximise information transmittance - the 'INFOMAX' principle [21] - perform PCA. It was further shown by Nadal and Parga [49], that non-linear-mappings that follow this principle are capable of producing factored distributions in the source space. In an effort to model information transfer in neurons, Bell and Sejnowski [45] extended the INFOMAX principle to non-linear mappings of non-Gaussian densities. The input data was first mapped to intermediate variables by a linear transform, then these were mapped by sigmoidal non-linearities to an output. By maximising the entropy of the output through adjusting the linear transform parameters, they showed that the intermediate variables were a linear ICA projection of the input data. Similar algorithms were independently suggested by Roth and Baraum [63] and Cardoso and Laheld [59].

The original INFOMAX algorithm adjusted the linear mapping, but left the non-linear mapping static during the learning phase (equivalent to adjusting Υ for a given $\Re^L$). Consequently, it could only factor the input distribution into products of unimodal, super-Gaussian distributions (highly peaked distributions with more mass in the tails than a Gaussian - i.e. positive kurtosis). This was fine if the algorithm was used to separate mixtures of voices or music as these tend to be super-Gaussian distributed. For sub-Gaussian sources (i.e. negative

kurtosis), however, separation was not possible. A solution was proposed by Girolami [64] whereby the sources switched between sub- and super-Gaussian distributions according to kurtoses measurements and was incorporated into Bell and Sejnowski's formulation by Lee *et al.* [65].

2.4.2 Modelling approach

The maximum-likelihood (ML) approach was first used by Gaeta and Lacoume [66] and later by Pham, Garrat and Jutten in 1992 [67]. Pham *et al* treated the source densities as fixed as they considered these to be nuisance parameters and evaluated only the mixing matrix. Important contributions by MacKay [68], Pearlmutter and Parra [69] and Cardoso [70] showed that the INFOMAX algorithm could be derived from ML estimation of a square-mixing, noiseless model. These models were then extended to the non-square case.

Everson and Roberts [35] extended these methods by incorporating a flexible generalised-exponential model for the source densities that could learn both super- and sub-Gaussian distributions. This is equivalent to learning the non-linearity in the INFOMAX case. They also noted that an unmixing matrix that has independent columns must also be decorrelated. This information was used to constrain the learning to the manifold of decorrelating matrices, thereby greatly speeding up the process.

A particularly sophisticated model was developed by Attias [46] that incorporated full-covariance Gaussian noise and modelled each source distribution as a Mixture of Gaussians (MoG). These mixture models are capable of learning almost any density given enough constituent Gaussians. The model - dubbed by Attias as 'Independent Factor Analysis' - was learnt through the Expectation-Maximisation (EM) [71] algorithm. Due to the model's sophistication, the learning and inference process were relatively slow, so the 'variational' [72] approximation was also incorporated when dealing with a large number of sources.

The formulation of ICA in the Bayesian framework is a relatively recent

development due to its computational complexity. Laplace approximations [73] were used by Roberts [74] to make the Bayesian integrations over the mixing matrix and (fixed) source densities tractable, while the isotropic Gaussian noise model was learnt using standard ML. Roberts also showed how the Bayesian approach could be used to infer the most likely dimensionality of the underlying source space. Knuth [75] gave an impressively clear explanation of how the INFOMAX ICA algorithm could be derived using Bayesian considerations. The Bayesian methodology was then used to incorporate physical considerations in real signal mixing into the model. The positive effect of these was demonstrated by separating mixtures of 'Star Trek'^{TM} sound effects.

Lappalainen introduced the variational approximation to Bayesian ICA in [76]. The prior densities over the model parameters were all Gaussian and the approximating distribution to the posterior was also a Gaussian. Furthermore, the source densities were unimodal MoGs. This simplified the learning process but limited the flexibility of the model. A similar formalism was used by Lawrence and Bishop [77], but where a richer variety of functional forms for the priors are used. No functional forms for the approximating distribution to the posterior were assumed - the forms for the posteriors expressed themselves in the optimisation procedure. Crucially, however, the source model was kept fixed and only the mixing matrix and noise statistics were learnt. This was rectified by Miskin and MacKay in [78], although the source densities were left unimodal.

2.4.3 Unification

In fact, both the mapping and modelling strands of research have now been brought under the unifying framework of probabilistic modelling. Lee *et al* detailed in [79] how all the approaches detailed above could be shown to be minimising mutual information (as one would expect!). Using similar arguments, it can be shown [80] that minimising mutual information is equal to maximising a generative log-likelihood to within an additive constant. In other words,

there exist generative models for all the above procedures. As such, it is the generative modelling formalism that is presented here.

2.5 Generative Model

The ICA projection can be realised by constructing a model for the observation sequence generating process. The information is considered generated in some source space and arrives at the sensor array through a process that mixes L independent source sequences into M dependent sensor ones, where $M \geq L$. Noise may be added by the sensors, before outputting the M sequences. Let $\boldsymbol{x}^t = [x_1^t, ..., x_M^t]^T$ be a random column-vector describing an observation at time t and let $\boldsymbol{s}^t = [s_1^t, ..., s_L^t]^T$ denote the associated random vector in the source space. The generative model can then be described as a mapping from L sources to M sensors

$$\boldsymbol{x}^t = \boldsymbol{f}\left(\boldsymbol{s}^t\right) + \boldsymbol{n}^t \tag{2.15}$$

where $\boldsymbol{f}$ is a deterministic mapping function and $\boldsymbol{n}$ is an M-dimensional additive noise vector. The goal of ICA is to estimate the mapping, noise statistics and source vectors, $\boldsymbol{S} = \{\boldsymbol{s}^1, ..., \boldsymbol{s}^t, ..., \boldsymbol{s}^T\}$, given observations $\boldsymbol{X} = \{\boldsymbol{x}^1, ..., \boldsymbol{x}^t, ..., \boldsymbol{x}^T\}$. Although, in principle, the mapping $\boldsymbol{f}$ may be any smooth, deterministic function, in practice non-linear mappings suffer identifiability problems [81], so for mathematical and computational simplicity the mapping will be considered linear and instantaneous

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{N} \tag{2.16}$$

where $\boldsymbol{X}$ is the $M \times T$ data matrix, $\boldsymbol{A}$ is the $M \times L$ *basis* or *mixing* matrix, $\boldsymbol{S}$ is the $L \times T$ source matrix and $\boldsymbol{N}$ is an $M \times T$ Gaussian noise matrix.

How one interprets $\boldsymbol{A}$ or $\boldsymbol{S}$ depends on one's view of the data. If the data are M time-varying signals, the data can be seen as a linear mixing of L underlying independent signals represented by the L rows of $\boldsymbol{S}$. For example, the data may be signals received at M microphones at a cocktail party. These signals are then simply mixings of L independent voices. ICA is used to 'blindly' (i.e.

without access to $\boldsymbol{S}$ and $\boldsymbol{A}$) separate the sensor signals into the underlying voice signals $\{\boldsymbol{s}, ..., \boldsymbol{s}_L\}$. This is the Blind Source Separation problem in signal processing literature [56]. If the observations are considered to be constructed from varying amounts of L static features then the columns of $\boldsymbol{A}$ represent the L static basis vectors and the components of $\boldsymbol{s}^t$ represent the amount of each basis used for a given data vector, $\boldsymbol{x}^t$. For example, if the data represent an image, that image may be considered a mixture of underlying (maybe fewer) independent edges, textures etc. The aim of ICA in this case is to 'unmix' the dataset and recover these representative features, a process known as feature extraction.

In either case, ICA can be recast as data density modelling. The data density $p(\boldsymbol{x})$ may be considered a linear transform (i.e. scaling, rotation, shearing and/or a possibly higher-dimensional projection) of an unknown manifold - the source density $p(\boldsymbol{s})$. In BSS, the unknown manifold is the distribution of the independent source signals before mixing. ICA is primarily used to recover the source density, estimating the transform matrix and noise statistics in the process. In feature extraction, ICA is primarily used to recover the transform matrix, whose columns represent the independent features. The latent source manifold is the distribution of amplitudes - the proportion of each feature used for each observation. In both cases, ICA models the data density as an underlying, factored source density projected through a transform matrix (see Figure 2.3).

2.5.1 Identifiability of ICA solution

There are some restrictions that must be made if a solution is to be identifiable. First of all, ICA cannot separate mixings of Gaussians. A mixing of Gaussian signals is itself a Gaussian so decorrelation alone is enough to factor a Gaussian distribution. Although the mixing matrix may be recovered, the factors are not identifiable. One explanation for this is discussed above in the section on Principal Component Analysis. An alternative is that the Central Limit

Theorem implies that mixing independent distributions result in distributions that tend towards Gaussianity. This means that the observed data density is more Gaussian than the hypothetical sources. When ICA attempts to factor this M-dimensional density into L 1-dimensional densities, these source densities are, in a sense, driven *away* from Gaussianity. Given these restrictions, it can be shown [14] that ICA is possible only if no more than one source is Gaussian distributed, usually reserved for the noise process describing N in (2.16).

A second restriction is that $M \geq L$, otherwise - as in simultaneous equations - there are more unknowns than equations, although some inroads have been made into the 'over-complete' case ($M < L$) [38, 82, 83]. A third restriction related to the second is that the mixing matrix must be full column rank. In other words, dimensions cannot collapse in the hypothetical mixing process of (2.16) as this process cannot be reversed.

Whether a mapping or a modelling approach is used, there are inherent ambiguities in the ICA learning process. These can be illustrated in the following way. Let $\hat{\boldsymbol{A}}$ represent an estimate of the mixing matrix and $\hat{\boldsymbol{S}}$ denote an estimate of the source matrix. Equation (2.16) implies two ambiguities in the estimation of $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{S}}$

- If $\boldsymbol{P}$ is an $L \times L$ permutation matrix, then the mappings $\boldsymbol{A} \mapsto \boldsymbol{AP}$ and $\boldsymbol{S} \mapsto \boldsymbol{PS}$ leave the data matrix, $\boldsymbol{X}$, unaffected.
- If $\boldsymbol{D}$ is an $L \times L$ diagonal scaling matrix, then the mappings $\boldsymbol{A} \mapsto \boldsymbol{AD}$ and $\boldsymbol{S} \mapsto \boldsymbol{D}^{-1}\boldsymbol{S}$ also leave the data matrix unaffected.

It then follows that - in the noiseless case - the best that can be achieved is

$$\hat{\boldsymbol{A}} = \boldsymbol{APD} \tag{2.17}$$

$$\hat{\boldsymbol{S}} = \boldsymbol{PD}^{-1}\boldsymbol{S} \tag{2.18}$$

with equivalent ambiguities under the mapping approach. The permutation and scaling of $\boldsymbol{S}$ in (2.18) leaves the mutual information of the sources unchanged (i.e. $I[\boldsymbol{PD}^{-1}\boldsymbol{S}] = I[\boldsymbol{S}]$), so all values of $\hat{\boldsymbol{S}}$ that satisfy (2.18) are perfectly good source estimates.

The permutation ambiguity is broken if there is anisotropic sensor noise while the scaling ambiguity may be removed by either setting the source variances to unity or by column-normalising $\boldsymbol{A}$. In practice, however, these indeterminancies have no effect on ICA performance.

With the problem now correctly posed - and the caveats exposed - how is the likelihood constructed?

2.5.2 Likelihood

Let $\mathcal{M}$ represent a particular generative model. The observations $\boldsymbol{X}$ are generated by the model in the following fashion. A source vector $\boldsymbol{s}^t$ is generated at time t by model $\mathcal{M}$ according to a process governed by parameters[3] $\boldsymbol{\theta}$. The probability of this occurance is denoted $p(\boldsymbol{s}^t|\boldsymbol{\theta}, \mathcal{M})$. This source vector is then mixed by the linear mapping $\boldsymbol{A}$ and noise is added to give observation vector $\boldsymbol{x}^t$. The probability of generating observation vector $\boldsymbol{x}^t$ at time t given the source vector $\boldsymbol{s}^t$, mixing matrix $\boldsymbol{A}$ and noise parameter $\boldsymbol{\Lambda}$ of model $\mathcal{M}$ is $p(\boldsymbol{x}^t|\boldsymbol{s}^t, \boldsymbol{A}, \boldsymbol{\Lambda}, \mathcal{M})$. There is a distinction to be made here. The parameters $\{\boldsymbol{A}, \boldsymbol{\Lambda}, \boldsymbol{\theta}\}$ are considered part of the model structure while the variable $\boldsymbol{s}$ is separate to the model as it is generated by it. Because its values are hidden and never observed, it is labelled a *hidden* or *latent* variable. The *likelihood* of observing datum $\boldsymbol{x}^t$ is the probability of generating it irrespective of the configuration of latent variables so these have to be integrated out

$$p(\boldsymbol{x}^t|\boldsymbol{\Theta}, \mathcal{M}) = \int p(\boldsymbol{x}^t|\boldsymbol{s}^t, \boldsymbol{A}, \boldsymbol{\Lambda}, \mathcal{M}) p(\boldsymbol{s}^t|\boldsymbol{\theta}, \mathcal{M}) d\boldsymbol{s} \tag{2.19}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{A}, \boldsymbol{\Lambda}, \boldsymbol{\theta}\}$. The conditioning on $\mathcal{M}$ indicates that all implicit assumptions in the construction of $\mathcal{M}$, such as model structure, dimensionality of the source space etc., are acknowledged and their effect on relevant quantities made explicit.

By definition of the ICA projection, the source density factorises

$$p(\boldsymbol{s}^t|\boldsymbol{\theta}, \mathcal{M}) = \prod_{i=1}^{L} p(s_i^t|\theta_i, \mathcal{M}) \tag{2.20}$$

[3]Strictly speaking, the parameters should be subscripted $\mathcal{M}$ to signify model membership, but this has been omitted for brevity and is clear from context.

where $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_L\}$. Substituting (2.20) into (2.19) gives the likelihood of an observation being generated by the ICA process described above

$$p(\boldsymbol{x}^t|\boldsymbol{\Theta}, \mathcal{M}) = \int p(\boldsymbol{x}^t|\boldsymbol{s}^t, \boldsymbol{A}, \boldsymbol{\Lambda}, \mathcal{M}) \prod_{i=1}^{L} p(s_i^t|\theta_i, \mathcal{M}) ds_i \tag{2.21}$$

The likelihood for all T observations (assuming identically and independently distributed (iid) data) is then

$$p(\boldsymbol{X}|\boldsymbol{\Theta}, \mathcal{M}) = \prod_{t=1}^{T} p(\boldsymbol{x}^t|\boldsymbol{\Theta}, \mathcal{M}) \tag{2.22}$$

Finally, the measure ψ required is the log-likelihood and is simply

$$\psi \doteq \mathcal{L}_{\mathcal{M}}(\boldsymbol{X}|\boldsymbol{\Theta}) = \sum_{t=1}^{T} \log p(\boldsymbol{x}^t|\boldsymbol{\Theta}, \mathcal{M}) \tag{2.23}$$

The log-likelihood quantifies everything about an ICA model. Specific models for the noise process can be coded into $p(\boldsymbol{x}|\boldsymbol{s}, \boldsymbol{A}, \boldsymbol{\Lambda}, \mathcal{M})$ while the same can be done for $p(\boldsymbol{s}|\boldsymbol{\theta}, \mathcal{M})$.

Now that the measure $\psi \doteq \mathcal{L}_{\mathcal{M}}(\boldsymbol{X}|\boldsymbol{\Theta})$ has been constructed, it can be extremised to find the relationship $\Upsilon \doteq \boldsymbol{f} : \Re^L \mapsto \Re^M$. This mapping is defined in (2.16) by the mixing matrix, $\boldsymbol{A}$, and noise statistics, $\boldsymbol{\Lambda}$. The next section illustrates how this mapping can be found in the simple noiseless, square-mixing case.

2.6 Simple Example - Square, Noiseless ICA

In the noiseless, square-mixing ($L = M$) case, $\Upsilon = \boldsymbol{A}$. If the mapping is noiseless, then for a single observation

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{s}, \boldsymbol{A}, \boldsymbol{\Lambda}, \mathcal{M}) &= \delta(\boldsymbol{x} - \boldsymbol{A}\boldsymbol{s}) \\ &= \prod_{j=1}^{M} \delta(x_j - \sum_{j=1}^{M} A_{ji} s_i) \end{aligned} \tag{2.24}$$

This is not strictly a probability as the delta function assigns unity to everywhere in the range of $\boldsymbol{A}$ that coincides with data and zero elsewhere, but is adequate for this example. Substituting (2.24) into (2.21) gives

$$p(\boldsymbol{x}|\boldsymbol{\Theta}, \mathcal{M}) = \int \prod_{j=1}^{M} \delta(x_j - \sum_{j=1}^{M} A_{ji} s_i) \prod_{i=1}^{L} p(s_i|\theta_i, \mathcal{M}) ds_i$$

$$= \frac{1}{|\det \boldsymbol{A}|} \prod_{j=1}^{M} p(\hat{s}_i|\theta_i, \mathcal{M}) \tag{2.25}$$

where $\hat{s}_i = \sum_j A_{ij}^{-1} x_j$. As $L = M$, $\boldsymbol{A}$ is square so using the substitution $\boldsymbol{W} = \boldsymbol{A}^{-1}$, the log-likelihood is

$$\mathcal{L}_{\mathcal{M}}(\boldsymbol{x}|\boldsymbol{\Theta}) = \log|\det \boldsymbol{W}| + \sum_{i=1}^{L} \log p(\hat{s}_i|\theta_i, \mathcal{M}) \tag{2.26}$$

The vector $\hat{\boldsymbol{s}}$ is the projection of the observation vector onto the source space to give a hypothesised source vector. This is a pleasing and intuitive result. The log-probability of the observed vector $\boldsymbol{x}$ being generated by $\mathcal{M}$ is simply the log-probability of $\mathcal{M}$ generating the hypothetical source vector $\hat{\boldsymbol{s}}$ to within an additive constant, $\log|\det \boldsymbol{W}|$. This measures the change in volume in going from the observation space to the source space. Maximising the log-likelihood with respect to the mapping $\boldsymbol{W}$ maximises the sum of the volume change and the probability assigned to each source vector. This sum is fundamental to the ICA projection.

2.6.1 How does ICA work?

Before ICA can project the observation density onto the source space, the mapping $\boldsymbol{W}$ must be learned. This is achieved by maximising (2.26) w.r.t. $\boldsymbol{W}$ (assuming for the time being that $\boldsymbol{\theta}$ is known). For clarity, rewrite (2.25) as

$$p(\boldsymbol{X}) = |\det \boldsymbol{W}|\, p(\boldsymbol{W}\boldsymbol{X}) \tag{2.27}$$

for T data points. The distribution on the right is a linear projection of the distribution on the left. For a non-zero probability on the left, the determinant of $\boldsymbol{W}$ is non-zero (i.e. $\boldsymbol{A}$ is full column rank) so no dimensions collapse in the projection. Any linear projection can only scale, rotate, shear and permute, therefore any structure - and thus information - in the data distribution is preserved in the source distribution. The properties that this projection has are determined by the ICA learning process.

The first term on the RHS is the volume change of the distribution from observation space to source space. The second term is the probability distribution

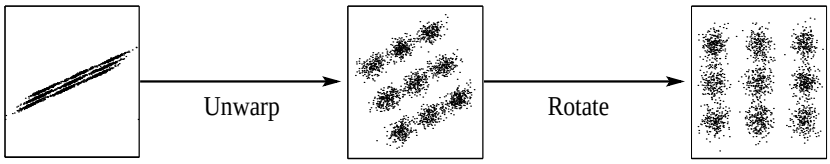


Figure 2.4: The ICA process.

of the projected data. Learning consists of maximising the product of these two quantities. Both of the quantities cannot simultaneously increase during learning as they both act in opposing ways upon the projected data. If the volume of the projection increases, the source pdf spreads out and the probability assigned to each point decreases (as the pdf must integrate to unity), so $p(\boldsymbol{W}\boldsymbol{X})$ decreases. Increasing $p(\boldsymbol{W}\boldsymbol{X})$ makes the source pdf want to assign maximum probability mass to the source vectors, reducing its spread and therefore the volume change $|\det \boldsymbol{W}|$. This, however, is opposed by the maximising pressure on $|\det \boldsymbol{W}|$. If the observation density is asymmetric, then the volume-increasing force will be larger than the pdf-decreasing force in directions with high data density and vice versa in directions with low density. Therefore, maximising the product in (2.27) results in a source density that is the most symmetric possible projection of the observation density while maintaining the intrinsic structure. This is how ICA 'unwarps' densities (see Figure 2.4) and is equivalent to PCA.

Now rewrite (2.27) as

$$p(\boldsymbol{X}) = |\det \boldsymbol{W}| \prod_{t=1}^{T} \prod_{i=1}^{L} p(\sum_{j=1}^{M} W_{ij} x_j^t) \tag{2.28}$$

The learning process tries to maximise $p(\boldsymbol{W}\boldsymbol{X})$ by maximising the product of its marginal densities. Each marginal is a projection of the source density onto one of L axes. If these marginals are not independent, then increasing one decreases the product of the others by the same amount as there is less density to project onto the remaining axes. This leaves the value of the product unchanged. The only way of increasing the product of the marginals is if the axes are rotated to make each marginal independent of the others. Therefore, a projection is learnt that rotates the observation density to 'line it up' with the axes. This is how

ICA 'un-rotates' the density.

Although the likelihood above is from a special case (noiseless, square mixing), the core ICA procedure is the same for any formulation.

2.6.2 Using the likelihood to learn

To learn the ICA model, the likelihood in (2.26) must be maximised with respect to the mapping, $\boldsymbol{W}$, and source model parameters, $\boldsymbol{\theta}$. To simplify this illustrative example, assume the source model is known and that only $\boldsymbol{W}$ need be learned. Differentiating (2.26) w.r.t. $\boldsymbol{W}$ and using the appropriate matrix identities gives

$$\frac{\partial}{\partial W_{ij}} \mathcal{L}_{\mathcal{M}}(\boldsymbol{x}|\boldsymbol{\Theta}) = A_{ji} + x_j \phi_i(\hat{s}_i) \tag{2.29}$$

where

$$\phi_i(\hat{s}_i) = \frac{\partial \log p(\hat{s}_i|\theta_i, \mathcal{M})}{\partial \hat{s}_i} \tag{2.30}$$

and where the definition of $\hat{s}_i^k = \sum_j W_{ij}^k x_j$ has been reintroduced. A simple, gradient-ascent learning rule for the elements of $\boldsymbol{W}$ is then

$$W_{ij}^{k+1} = W_{ij}^k + \Delta W_{ij}^k \tag{2.31}$$

where k indicates the k^{th} estimate of $\boldsymbol{W}$ and where

$$\Delta W_{ij}^k = \eta \left[A_{ji}^k + x_j \phi_i(\hat{s}_i^k) \right] \tag{2.32}$$

in which η is a user-defined learning rate. In order to proceed, a form for the source density $p(\hat{s}_i|\theta_i, \mathcal{M})$, or alternatively the non-linear function $\phi_i(\hat{s}_i)$, must be specified. If a Gaussian source model is chosen, then $\phi_i(\hat{s}_i)$ is a linear function of the datum x_j. The right hand term in the brackets then reduces to a covariance measure which means $\boldsymbol{W} = \boldsymbol{A}^{-1}$ is a decorrelating matrix. In other words, ICA with a Gaussian source model is (unnormalised) PCA without the variance ordering.

If a non-linear function of the form

$$\phi_i(\hat{s}_i) = \frac{1}{1 + e^{-\hat{s}_i}} \tag{2.33}$$

is chosen then (2.32) gives precisely the learning rule presented in [45]. Therefore, the INFOMAX algorithm is a special case of the maximum-likelihood ICA model.

A mathematically convenient fixed source model is the reciprocal cosh density

$$p(\hat{s}_i|\mathcal{M}) = \frac{1}{\pi \cosh(\hat{s}_i)} \tag{2.34}$$

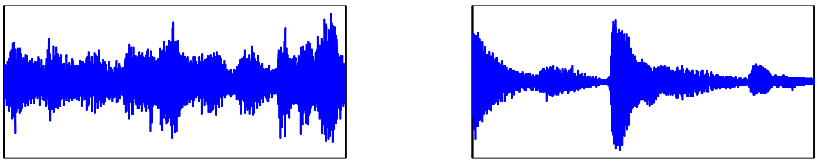
This density gives a simple form for $\phi_i(\hat{s}_i)$ and has positive kurtosis so also conforms to Field's notion of sparseness. The matrix update is now

$$\Delta W_{ij}^k = \eta \left[A_{ji}^k + x_j \tanh\left(\sum_j W_{ij}^k x_j \right) \right] \tag{2.35}$$

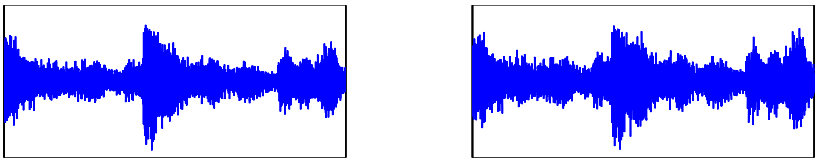
The right hand term in the brackets is now a nonlinear function of x_j. This implies that ICA is in fact a non-linear form of PCA. This is indeed the case as Roweis and Gharamani showed [84] that ICA could be interpreted as finding a non-linear mapping of Gaussian sources. They considered an invertible and differentiable non-linear mapping in the generative model. This was shown to be equivalent to a linear mapping of sources with non-Gaussian pdfs, which in turn resulted in a (different) nonlinearity in the learning rule. The $\tanh(\hat{s})$ nonlinearity in (2.35) is equivalent [84] to a non-linear mapping $g(a)$ of a zero-mean, Gaussian source a where

$$g(a) = \ln \left\{ \tan \left[\frac{\pi}{4} \left(1 + \mathrm{erf}(\frac{a}{\sqrt{2}}) \right) \right] \right\} \tag{2.36}$$

Learning $\boldsymbol{W}$ is an iterative process. First, some initial value of $\boldsymbol{W}$ is set, perhaps at random. Then (2.35) is iterated until the elements of $\boldsymbol{W}$ converge. Although relatively straightforward, this process can be slow if the gradient is shallow. There are also convergence and stability issues, which were examined by [85]. The learning can be greatly sped up by making (2.35) covariant [68, 58] which has the effect of making the 'units' on either side of the equality in (2.35) agree. Stability issues can be ovecome by appropriate choice of $\phi_i(s_i)$ [85].



(a) Music signals



(b) Music mixtures

Figure 2.5: Source and Sensor signals.

2.6.3 Results

To illustrate ICA in action, the algorithm presented above was used to separate two mixtures of two music sources. Music and voice signals are well represented by a reciprical-cosh density. The music was sampled at 11.3KHz and the two signals were mixed by a randomly generated matrix. Figure 2.5 plots both the original source (music) signals and the mixture (sensor) signals. The ICA algorithm was intialised randomly and learning took place on 500 samples randomly drawn from the 20000 sample dataset.

As a comparison, PCA was also used to decompose the mixtures. Figure 2.6 shows the source reconstructions of PCA and ICA. The sources recovered by PCA are clearly incorrect. The ICA reconstructions, on the other hand, are very close to the original music signals, albeit inverted and permuted.

The quality of the source reconstructions can be appreciated when plotted against the original source signals. Figure 2.7 is a scatter plot of true and reconstructed sources. The PCA plot on the left shows how poorly the sources have been recovered. The ICA reconstructions, however, have a very narrow scatter range and are, therefore, more accurate reconstructions. The quality

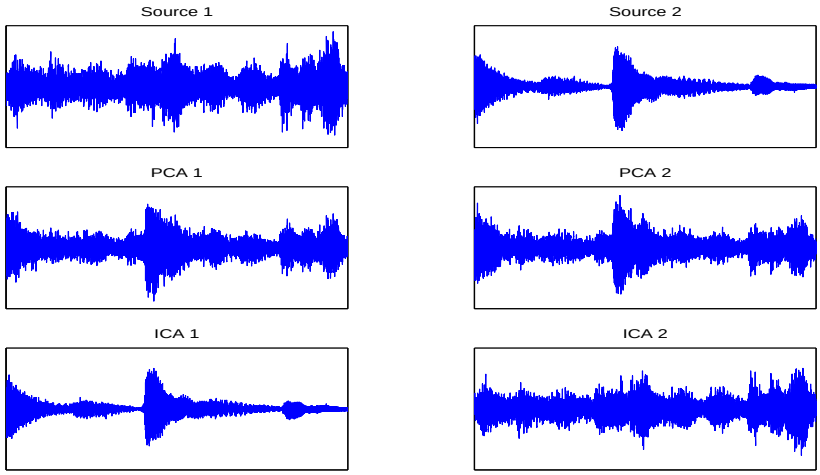
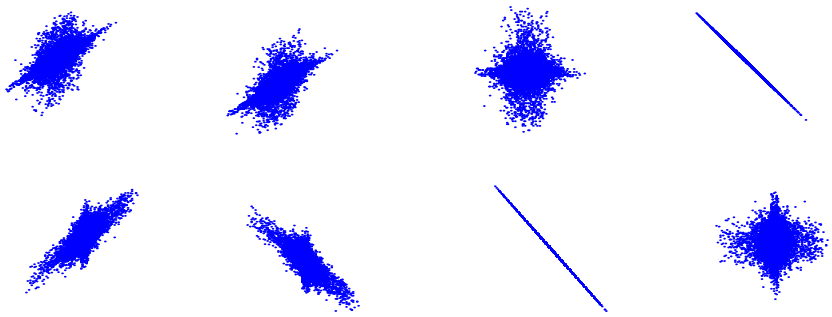


Figure 2.6: Music reconstructions.



(a) PCA scatter plot

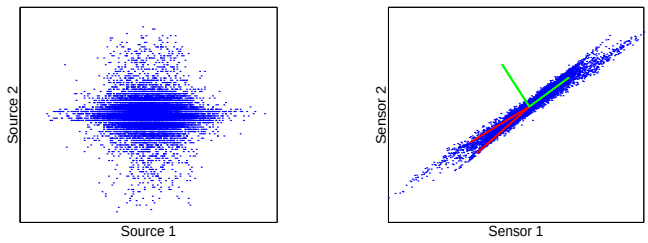
(b) ICA scatter plot

Figure 2.7: Scatter plots.

of the reconstruction can be quantified by computing the mean-square error (MSE) between the true and estimated sources

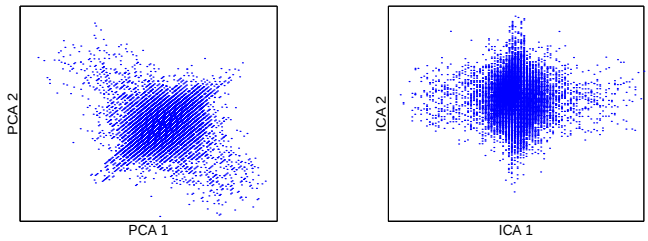
$$\text{MSE} = \frac{1}{LT}\sum_{t=1}^{T}\sum_{i=1}^{L}(s_i^t - \hat{s}_i^t)^2 \tag{2.37}$$

where $\hat{s}_i^t = \sum_j W_{ij}x_j^t$, the source reconstruction. The source reconstructions were permuted and normalised with respect to the true sources s to remove the scaling and permutation ambiguities. Over the whole 20000 sample dataset, the PCA sources were found to have MSE = 4.087. The ICA error was three orders of magnitude lower with MSE = 0.004.



(a) Source density

(b) Data density



(c) PCA density

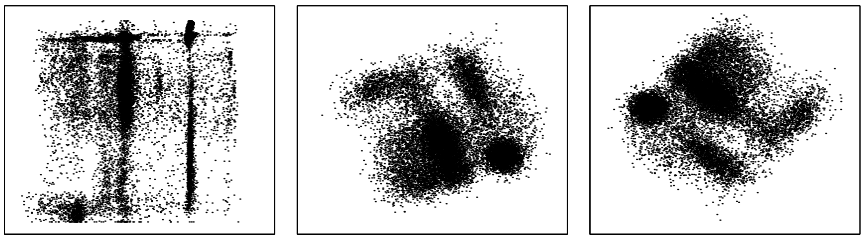
(d) ICA density

Figure 2.8: Music, sensor, PCA and ICA densities.

The difference in performance stems from the orthogonality and Gaussian constraints of PCA. Figure 2.8(a) shows the original source density. The sensor density is shown in Figure 2.8(b) together with the representative axes found by PCA (green) and ICA (red). The orthogonality and Gaussian constraints of PCA leads to a recovered source density that is a rotation of the true source density, shown in Figure 2.8(c). ICA has no such constraints and consequently recovers the true density (with permuted sources) shown in Figure 2.8(d).

2.7 Problems and Limitations

Although this simple ICA algorithm performs well on noiseless mixtures of sound, it does suffer from three main problems. Firstly, there is no noise model so its effectiveness diminishes greatly with noisy data. More importantly, the source model is fixed and unimodal. A well matched source model is crucial in finding independent directions. If the source density is not modelled accurately,



(a) Source density (b) Data density (c) ICA density

Figure 2.9: Image data.

source signals may not be separable. Consider the density illustrated in Figure 2.9(a). This is the density of the images in Figure 2.10(a) and is clearly not well described by a product of reciprical cosh pdfs; in fact, it is multi-modal. The data density in Figure 2.9(b) represents the mixtures in Figure 2.10(b).

The mixing is a small, orthogonal rotation of the original source density with 5% added Gaussian noise. The ICA algorithm is unable to recover the source density as its source model is woefully inadequate. Figure 2.9(c) shows the density recovered and Figure 2.10(c) are the poorly reconstructed source images. If these images are to be recovered, a source model capable of capturing multi-modal densities is needed. Furthermore, the real world is a noisy place, so an explicit noise model is necessary if ICA is to be applied to real signals.

Finally, there is no way of inferring how many sources there are i.e. the intrinsic dimensionality of the data manifold. If this is not known in advance, then spurious sources may be found which have little or no relevance in analysing the data.

2.8 Summary

The desire for representation based on independent components is motivated by mathematical and neurological concerns. Mathematically, an efficacious representation of raw data makes subsequent analyses much easier. Some coordinate system intrinsic to the data is sought based on some criterion to be optimised.

(a) Source Images

(b) Image mixtures

(c) ICA Reconstructions

Figure 2.10: Image Reconstructions.

Principal Component Analysis assumes the data distribution is Gaussian and tries to diagonalise the covariance structure. Unfortunately, most 'interesting' datasets are highly non-Gaussian, displaying structure beyond covariance. In these cases, PCA has been shown to be inadequate in finding an appropriate representation. Exploratory Projection Pursuit fares much better as its criterion is based on the notion of non-Gaussianity. If the data are noise-free, then EPP is equivalent to Independent Component Analysis. The computational complexity of EPP, however, limits its use to low-dimensional representations of noise-free data.

Research into early sensory processing has hinted at two possible strategies the neocortex employs to represent sensory data - redundancy reduction and sparse coding. If the primary goal of the neocortex is to reduce redundancy across the sensory inputs, then an important stage in this reduction is the construction of a compact factorial code. Such a code represents the statistically dependent input signals by fewer statistically independent ones. These signals can then be easily manipulated to reduce overall redundancy. A related strategy is sparse coding. In this case, the neocortex represents the inputs using the

minimum number of active sensory units, although the total may equal the number of input signals. This has also been shown to produce a factored code. In both these cases, then, independence between representative components is a fundamental property of the sensory coding.

Independent Component Analysis is the practice of minimising the statistical dependence between non-Gaussian signals. ICA generalises PCA while being more robust than EPP. It is a factorial coding scheme that can produce both compact and sparse representations. In its most general form, ICA is performed by learning a model of the hypothosised (real or mathematically conveniant) data generation process. The most basic model assumes noise-less data, square mixing and simple non-Gaussian distributions. This model can be extended to include Gaussian noise and non-square mixing, reaching its logical conclusion with Attias' 'Independent Factor Analysis' [46]. Although very general, the formalism in [46] can suffer from over-fitting, lacks a method for incorporating prior knowledge one has of the problem domain and cannot compare models with different underlying assumptions. For example, what is the most appropriate number of source dimensions? What is the most appropriate mixing model? How can over-fitting be avoided if there is little data? A *Bayesian* framework must be used to move beyond [46] and answer these questions. This framework and how it can be used is introduced in the next Chapter.

Chapter 3

Bayesian Modelling

The Western tradition explores and understands phenomena by breaking them down into smaller pieces that are easier to understand. These pieces are individually analysed to see what role they play and how each piece relates to each other. This collection of descriptions constitutes a *model*, a simplified description of the phenomena. A model (implicitly or explicitly) encodes beliefs one has about the nature of the phenomena, not least of which is the assumption that it *can* be understood as a 'sum of its parts' in the first place.

In data analysis, generative models encode the assumptions and beliefs one has about the system and processes that generated the observed data and, therefore, constitute an 'explanation' for the data observed. If a model is to be imposed, these beliefs must be subject to revision as and when the data demand it, and must be manipulated in a rational and consistent manner.

The beliefs embodied by a generative model, $\mathcal{M}$, are encoded in the parameters and structure of the model. The model parameters, $\boldsymbol{\Theta}$, govern the behaviour of individual sub-sections of the model while the model structure, $\mathcal{S}$, defines the relationship between these sub-sections. The sub-sections may be variables or quantities of interest, or even small models in their own right. The values of these variables (for example) may be given by the data under analysis, or some of them may not be directly accessible and therefore 'hidden'. The values of these hidden variables will have to be inferred using their relationships with the visible variables. When data is to be analysed using generative modelling,

one must first construct a model by choosing a parameteric form i.e. *how* the parameters govern the behaviour of the variables/quantities/sub-models, and a structure for the model that signifies the relationships between these variables etc. The model is then 'learnt' by adjusting the parameters (and, in principle, structure) and inferring the values of the hidden variables to best explain the observed data. This model can then be used to make further inferences and predictions about the data and may be compared with other, competing explanations.

3.1 Bayes Theorem

The learning of a model, and its use thereafter, rests on the rational and consistent manipulation of beliefs. If degrees of belief are quantified with probabilities, Cox showed [3] that, under reasonable assumptions, Bayes' theorem is the only rational and consistent way to manipulate these beliefs.

Bayes' theorem is a prescription for updating one's beliefs in the light of new information, and was discovered independently by Reverend Thomas Bayes [86] and Laplace [87]. The theorem was derived in section 1.1.1 and is repeated here

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \tag{3.1}$$

This innocuous relationship has far reaching consequences for machine learning: it is a prescription on how to systematically update ones knowledge of a problem domain given the data observed. In words, (3.1) simply states that one's understanding of y after seeing data x (the posterior $p(y|x)$) is the previous knowledge of y (encapsulated in the prior $p(y)$) modified by how likely the observation x is under that previous model (the likelihood $p(x|y)$). The denominator, $p(x)$, is a normalising term called the marginal likelihood, or *evidence*, and ensures that the posterior behaves as a probability. As discussed in section 1.1.1, Bayes' rule allows one to infer information that would otherwise be difficult to obtain, for example the probability of a disease given the symptoms or the most likely underlying (hidden) causes of the given observations. Using the

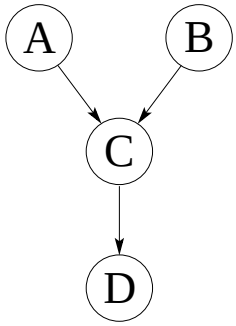


Figure 3.1: A simple Directed Acyclic Graph (DAG).

beliefist notion of probabilities, Bayes' theorem can also be used to quantifiably assess assumptions underlying different models.

3.2 Graphical Models

Probabilistic models of problem domains can be presented using a graphical representation. The variables being observed ($\boldsymbol{X}$), the hidden variables ($\boldsymbol{H}$) and the model parameters ($\boldsymbol{\Theta}$) are represented as nodes in a graph. Influences and relationships between these are represented as edges between nodes.

Basic concepts

A *Graph* $\mathcal{G}$ has a finite set of *nodes*, V, and a finite set of *edges*, E, between pairs of nodes: $\mathcal{G} = (V, E)$. Nodes may be circular, representing random variables, or square, representing deterministic ones. The edges can be undirected (*undirected graphs*), directed (*directed graphs*) or a mixture of the two (*mixed graphs*). In the context of generative models, only directed graphs are considered. Undirected graphs are important in statistical modelling such as image and spatial processing [88, 89, 90], and stochastic modelling in neural networks [91].

A graph, $\mathcal{G}$, can be associated with a set of variables $\boldsymbol{Y} = \{\boldsymbol{X}, \boldsymbol{H}, \boldsymbol{\Theta}\}$ by allocating a one-to-one relationship between the nodes, V, and the variables, $\boldsymbol{Y}$. A graph is represented with nodes as circles and squares, and with directed edges as arrows (lines if undirected).

Figure 3.1 shows a simple graph with $V = \{A, B, C, D\}$. The set $\{A, B\}$ are the *parents* of C

$$\text{pa}\,(C) \quad = \quad \{A, B\} \tag{3.2}$$

C is the *child* of $\{A, B\}$ and the parent of D

$$\text{ch}\,(A) \quad = \quad \{C\} \tag{3.3}$$

$$\text{ch}\,(B) \quad = \quad \{C\} \tag{3.4}$$

$$\text{pa}\,(D) \quad = \quad \{C\} \tag{3.5}$$

The parents and children of a node plus the children's *other* parents constitute a *Markov Blanket*. The Markov Blanket of A is $\{B, C\}$

$$\text{mk}\,(A) \quad = \quad \{\text{pa}(A), \text{ch}(A), \text{pa}(\text{ch}(A))\} \tag{3.6}$$

$$\text{mk}\,(A) \quad = \quad \{B, C\} \tag{3.7}$$

A graph makes explicit what is assumed about a problem domain. The lack of an edge between two nodes implies some sort of independence between the two equivalent variables. Directed edges imply some sort of relationship or conditional dependence. These characteristics of directed graphs make them ideal tools in representing conditional probability distributions over many variables. As edges explicitly code dependencies, the probabilistic relationship between a node and the rest of the model reduces to

$$p(v_i|\mathcal{G}) = p(v_i|\text{pa}(v_i)) \tag{3.8}$$

Therefore, once the graph is specified the complete equation of state of the model (the joint probability distribution over all the nodes) can be 'read-off' as a combination of simpler conditional densities

$$p(V|\mathcal{G}) = \prod_{i=1}^{|V|} p(v_i|\text{pa}(v_i)) \tag{3.9}$$

Figure 3.2 shows a graph of the simple noiseless ICA model derived in the previous Chapter (with 3 sources and 5 sensors for illustrative purposes). The

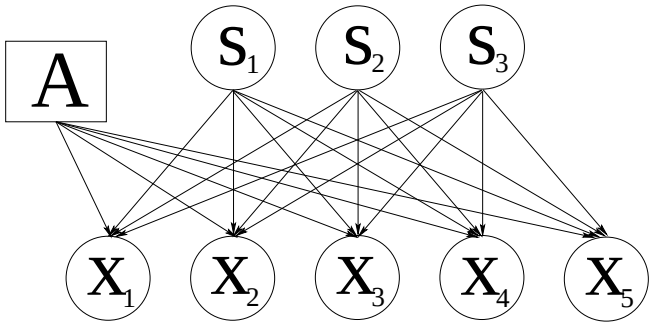


Figure 3.2: Graphical model of simple ICA.

lack of edges between the unobserved (hidden) source nodes codes the assumed independence between the sources. The parents of the observed nodes, $\boldsymbol{x}$, are $\{\boldsymbol{s}, \boldsymbol{A}\}$ and makes clear the conditional dependence of $\boldsymbol{x}$ on the mixing of source signals[1]. The square node for $\boldsymbol{A}$ (representing all the mixing matrix elements for brevity) signifies a deterministic rather than a random variable. The lack of parents for $\boldsymbol{s}$ means there is no parameteric source model (e.g. fixed reciprical cosh function). The equation of state for the model can be read-off as

$$p(\boldsymbol{x}, \boldsymbol{s}|\boldsymbol{A}, \mathcal{M}) = \prod_{j=1}^{M} p(x_j|\boldsymbol{s}, \boldsymbol{A}, \mathcal{M}) \prod_{i=1}^{L} p(s_i|\mathcal{M}) \tag{3.10}$$

Substituting in (2.24) and (2.34) gives a graphical model that represents the beliefs used in the example in section 2.6. Learning consists of manipulating these beliefs to maximise some objective, for example the likelihood the model generated the observed data. Further information on graphical models and their uses can be found in [92, 93, 94].

3.3 Bayesian Inference

In a Bayesian sense, inference is calculating the posterior probability density over the possible hidden variable values, while learning a generative model involves calculating the posterior probability density over the possible model parameter values. Therefore, inference and learning are effectively the same under a Bayesian framework. The distinction lies in the utility of the two - observations

[1]Note that $\boldsymbol{x} = \{x_j\}$ are *conditionally* independent i.e. iff $\boldsymbol{s}$ and $\boldsymbol{A}$ are known.

and hidden variables are quantites identified with the 'real world' whose distributions are being modelled using the parameters. Parameters exist to make the model 'work' and their distributions (if any) are modelled using further parameters (called *hyper-parameters*). Bayesian inference can be used to calculate both.

As the formalism is true for both hidden nodes and parameter values, the framework is presented using the weights $\boldsymbol{W}$ as a concatenation of both model parameters and hidden variables.

3.3.1 Learning the model

The posterior over weights $\boldsymbol{W}$ is given by

$$p(\boldsymbol{W}|\boldsymbol{X}, \mathcal{M}) = \frac{p(\boldsymbol{X}|\boldsymbol{W}, \mathcal{M})p(\boldsymbol{W}|\mathcal{M})}{p(\boldsymbol{X}|\mathcal{M})} \tag{3.11}$$

where $\mathcal{M}$ embodies all the other assumptions and beliefs about the model, for example the structure and the values for hyper-parameters governing the prior density $p(\boldsymbol{W}|\mathcal{M})$. The denominator in (3.11) is the evidence for $\mathcal{M}$ and ensures the posterior is normalised

$$p(\boldsymbol{X}|\mathcal{M}) = \int p(\boldsymbol{X}|\boldsymbol{W}, \mathcal{M})p(\boldsymbol{W}|\mathcal{M})d\boldsymbol{W} \tag{3.12}$$

The prior density captures all the information known about the possible weight values and constraints before data is seen and acts as a regulariser to limit overfitting. Possible weight values must simultaneously give a high data likelihood *and* be probable under the constraint of the prior to give an appreciable posterior probability. The posterior is a measure of what is known after the data is seen and quantifies any *new* knowledge gained. The data likelihood $p(\boldsymbol{X}|\boldsymbol{W}, \mathcal{M})$ is a measure of how well the model predicted the data and essentially decides whether the data under investigation contains any new information.

So, in theory, learning the model weights is simply computing the posterior over the weights. In practice, though, the integration in (3.12) is intractable for all but the most trivial models as it needs to be performed over the whole

weight space. Help is at hand, however, from the independence relationships between the weights.

In practice, $\boldsymbol{W}$ will be a collection of weight-sets over the different nodes of the model. If $\boldsymbol{W} = \{w_1, w_2, ..., w_L\}$, then the posterior density over a particular weight-set, w_i, is conditioned on the values of parent and children weights

$$p(w_i|\boldsymbol{X}, \text{mk}(w_i)) = \frac{1}{z} p(\boldsymbol{X}|\boldsymbol{W}) p(\text{ch}(w_i)|\text{pa}(\text{ch}(w_i))) p(w_i|\text{pa}(w_i)) \tag{3.13}$$

where the evidence in the denominator has been replaced by z for brevity, and where the explicit dependence on $\mathcal{M}$ is ignored for the same reason. The term $\text{mk}(w_i)$ is the Markov blanket of w_i (i.e. the parents, children and children's other parents). Note that the posterior is conditioned on the children as well as the data. The children of a node are considered the 'data' for that node regardless of whether they are real-world observations, hidden variables or other parameters. Note also that while the prior over w_i only depends only upon its parents, the posterior introduces dependencies between it and the other parents of its children. This is not surprising as w_i is, in a sense, 'competing' with the other parents to explain the observed behaviour of its children. To take generative responsibility for its children's behaviour, it must be aware of competing explanations to know what can be 'explained-away' by other parents.

The problem with a posterior of the form (3.13) is that it depends on point estimates of other, uncertain weights. This has the effect of compounding the uncertainty in w_i and leads to biased estimates [73]. Rather than choose one particular setting for the markov blanket of w_i, and then calculate the posterior, the uncertainty in the values of the blanket can be integrated out to obtain more robust estimates

$$p(w_i|\boldsymbol{X}, \mathcal{M}) = \int p(w_i|\boldsymbol{X}, \text{mk}(w_i), \mathcal{M}) p(\text{mk}(w_i)|\boldsymbol{X}, \mathcal{M}) d\text{mk}(w_i) \tag{3.14}$$

All the information concerning the other weights is used in computing this posterior so the uncertainty in w_i is reduced and the estimate is unbiased. As implied by (3.14), the multidimensional computation of (3.11) is reduced to a set

of lower dimensional computations that are cycled until the marginal posteriors in (3.14) are consistent with one another (i.e. converge).

The weights can normally be separated into model parameters, $\boldsymbol{\Theta}$, and hidden quantities of interest, $\boldsymbol{H}$, such that $\boldsymbol{W} = \{\boldsymbol{H}, \boldsymbol{\Theta}\}$. For example, in the ICA problem the source vectors are the hidden variables of interest while the parameters that govern their distributions and the mixing process are considered static and are there to define the model. The purpose of learning a model is to work out what values the parameters should take, or in a Bayesian sense, what the posterior distribution over $\boldsymbol{\Theta}$ should be. This will depend on what values of the hidden variables are likely. The possible hidden variable values are given by the marginal posterior distribution over $\boldsymbol{H}$

$$p(\boldsymbol{H}|\boldsymbol{X}, \mathcal{M}) = \int p(\boldsymbol{H}|\boldsymbol{X}, \boldsymbol{\Theta}, \mathcal{M}) p(\boldsymbol{\Theta}|\boldsymbol{X}, \mathcal{M}) d\boldsymbol{\Theta} \tag{3.15}$$

where

$$p(\boldsymbol{H}|\boldsymbol{X}, \boldsymbol{\Theta}, \mathcal{M}) = \frac{p(\boldsymbol{X}|\boldsymbol{H}, \boldsymbol{\Theta}, \mathcal{M}) p(\boldsymbol{H}|\boldsymbol{\Theta}, \mathcal{M})}{\int p(\boldsymbol{X}|\boldsymbol{H}, \boldsymbol{\Theta}, \mathcal{M}) p(\boldsymbol{H}|\boldsymbol{\Theta}, \mathcal{M}) d\boldsymbol{H}} \tag{3.16}$$

Similarly, the posterior over $\boldsymbol{\Theta}$ is

$$p(\boldsymbol{\Theta}|\boldsymbol{X}, \mathcal{M}) = \int p(\boldsymbol{\Theta}|\boldsymbol{X}, \boldsymbol{H}, \mathcal{M}) p(\boldsymbol{H}|\boldsymbol{X}, \mathcal{M}) d\boldsymbol{H} \tag{3.17}$$

where

$$p(\boldsymbol{\Theta}|\boldsymbol{X}, \boldsymbol{H}, \mathcal{M}) = \frac{p(\boldsymbol{X}|\boldsymbol{H}, \boldsymbol{\Theta}, \mathcal{M}) p(\boldsymbol{H}|\boldsymbol{\Theta}, \mathcal{M}) p(\boldsymbol{\Theta}|\mathcal{M})}{\int p(\boldsymbol{X}|\boldsymbol{H}, \boldsymbol{\Theta}, \mathcal{M}) p(\boldsymbol{H}|\boldsymbol{\Theta}, \mathcal{M}) p(\boldsymbol{\Theta}|\mathcal{M}) d\boldsymbol{\Theta}} \tag{3.18}$$

Learning the model, therefore, consists of working out what the posterior over $\boldsymbol{H}$ given the current estimate over the posterior over $\boldsymbol{\Theta}$, then working out the posterior over $\boldsymbol{\Theta}$ given the newly computed posterior over $\boldsymbol{H}$, and so on until convergence. When learning has converged, the posterior over the parameters is frozen and the model is now considered ready for use.

3.3.2 Using the model

Once the learning process is over, the model is considered an accurate representation of the process that generated the observations under investigation.

The purpose of using the model is to infer the hidden quantities that produced the observations. Some of the observations would have been used as training data, so the posterior over the hidden variables given that data would have been computed during the learning process. For new data, $\boldsymbol{x}_{\text{new}}$, the likely values of the hidden variables are given by their posterior

$$p(\boldsymbol{H}|\boldsymbol{x}_{\text{new}}, \boldsymbol{X}, \mathcal{M}) = \int p(\boldsymbol{H}|\boldsymbol{x}_{\text{new}}, \boldsymbol{\Theta}, \mathcal{M}) p(\boldsymbol{\Theta}|\boldsymbol{X}, \mathcal{M}) d\boldsymbol{\Theta} \tag{3.19}$$

where

$$p(\boldsymbol{H}|\boldsymbol{x}_{\text{new}}, \boldsymbol{\Theta}, \mathcal{M}) = \frac{p(\boldsymbol{x}_{\text{new}}|\boldsymbol{W}, \mathcal{M}) p(\boldsymbol{H}|\boldsymbol{\Theta}, \mathcal{M})}{\int p(\boldsymbol{x}_{\text{new}}|\boldsymbol{W}, \mathcal{M}) p(\boldsymbol{H}|\boldsymbol{\Theta}, \mathcal{M}) d\boldsymbol{H}} \tag{3.20}$$

The model can also be used to predict new data by computing the posterior density over new observations, called a predictive density

$$p(\boldsymbol{x}_{\text{new}}|\boldsymbol{X}, \mathcal{M}) = \int p(\boldsymbol{x}_{\text{new}}|\boldsymbol{W}, \mathcal{M}) p(\boldsymbol{H}|\boldsymbol{x}_{\text{new}}, \boldsymbol{\Theta}, \mathcal{M}) p(\boldsymbol{\Theta}|\boldsymbol{X}, \mathcal{M}) d\boldsymbol{H} d\boldsymbol{\Theta} \tag{3.21}$$

Although the parameter posteriors are fixed in the learnt model, they may be re-computed in the light of new information if one wants the model to continuously adapt as new information arrives. Such a learning regime is said to be 'on-line', as opposed to the 'batch' learning used in the examples above and for the rest of this thesis.

3.4 Model Comparison

In fact, Bayesian inference leads to a natural method for comparing assumptions and models. The ability to make assumptions and hypotheses explicit leads to a very useful application for Bayes' theorem: the most appropriate assumptions for the data model can be infered. If $\mathcal{M}$ is split into assumptions under scrutiny, $\mathcal{M}'$ and unquestioned assumptions, $\mathcal{I}$, then one can compare alternative assumptions, $\mathcal{M}'$ by calculating the posterior

$$p(\mathcal{M}'|\boldsymbol{X}, \mathcal{I}) = \frac{p(\boldsymbol{X}|\mathcal{M}', \mathcal{I}) p(\mathcal{M}'|\mathcal{I})}{p(\boldsymbol{X}|\mathcal{I})} \tag{3.22}$$

where $p(\boldsymbol{X}|\mathcal{M}', \mathcal{I})$ is the evidence for $\mathcal{M}'$. If all assumptions have equal probablity prior to comparison, then the maximum of (3.22) is given by the maximum

of $p(\boldsymbol{X}|\mathcal{M}', \mathcal{I})$. The evidence automatically penalises unnecessarily complex models and overly simple ones. Complex models have more degrees of freedom so can model a wide range of datasets, therefore the probability given to any one dataset (for example $\boldsymbol{X}$) is relatively low. Models that are too simple will not be able to fit the whole dataset $\boldsymbol{X}$ adequately so will also be accorded a relatively small probability for $\boldsymbol{X}$. Only models complex enough to explain the data sufficiently, but not complex enough to spread themselves too thinly, will be scored highly. This leads to a natural 'Occam's Razor' for model selection, as deftly explained in [73].

A fundamental problem in ICA is choosing the most appropriate number of component densities. It is clear from the discussion above that this can be inferred using Bayesian inference. Choosing a particular dimensionality of the source space is one of the main assumptions in constructing an ICA model. By training a number of models over a range of dimensionalities, the evidence for each model can be calculated and the optimal source dimensionality can be found.

3.5 Priors - Good or Bad?

The use of priors is seen as the greatest 'problem' with the Bayesian framework. The priors are defined by the user, so are necessarily subjective. Also, how are these priors to be chosen? Although much research has gone into priors and the optimal choice of priors (see [95], for example), this apparent problem is more red herring than weakness. All analyses involve preconceptions and assumptions of one sort or another: the beauty of the Bayesian approach is that these subjective preconceptions and assumptions are made explicit. Once made explicit, they are subject to scrutiny and falsifiability, the very essence of the scientific method. If the assumptions are appropriate, this will show itself in a compact posterior, and vice versa. This allows different models built on different assumptions to be compared objectively, probabilistically ranking more plausible models over less plausible ones.

Priors also provide natural regularisation. Specifying no prior is equivalent to setting a flat prior so all weight values are considered equally as likely. This can lead to numerical problems and implausible solutions such as densities collapsing onto a small subset of points in an effort to maximise the data likelihood. This gives models that 'over-fit' the training data which are poor at generalising beyond the training data. Priors can provide sensible constraints on the weight space allowing smoother, more general solutions.

3.6 Approximations to Bayesian Inference

The fundamental stumbling block in utilising Bayesian methods in learning and model comparison is the evaluation of the integral in (3.14). The correlations induced by Bayes' theorem mean the integration must be carried out over the whole Markov blanket. Calculating such integrals is generally computationally intractable in all but the simplest of models - in fact, it can be shown to be NP-hard [96]. Consequently, approximations have to be made to proceed. These approximation bring their own pathologies into the learning process.

3.6.1 Maximum likelihood

This is the most brutal approximation to Bayesian inference. Maximum likelihood (ML) does not define a prior over the weights $\boldsymbol{W}$ and finds point-estimates that maximise the data likelihood (or log-likelihood)

$$\boldsymbol{W}_{ML} = \arg\max_{\boldsymbol{W}} [p(\boldsymbol{X}|\boldsymbol{W}, \mathcal{M})] \tag{3.23}$$

This was the method used in section 2.6. Although computationally the simplest method of finding the weights, it cuts out all the benefits of Bayesian learning. The lack of constraint on the weight space can allow numerically improbable solutions as discussed above. This is like giving one's doctorate supervisor a precise date for the completion of a chapter and them actually believing you. Bayesian supervisors know better. Although the likelihood of finishing the chapter given the date and no other information may be high, the prior probability

of a reasonable date given that it's set by a student under pressure is sufficiently low to make the date highly improbable. Priors are there to ensure reasonable estimates. The lack of a density over $\boldsymbol{W}$ precludes the opportunity for model regularisation via the imposition of 'smoothness' and other constraints. There is no principled way of combating over-fitting leading to poor generalisation.

Furthermore, ML does not marginalise over all possible weights, so comparison of underlying model assumptions is not possible. More complex models have more free parameters so can necessarily model datasets more precisely than simpler ones. Consequently, the data likelihood itself will generally increase with complexity and inappropriate models may be chosen. Methods exist which introduce penalty terms for complex models, such as the Bayesian Information Criterion and Minimum Description Length [97] (see section 3.7), but these are far from optimal approximations to Bayesian model comparison [98]. Without the 'Occam's Razor' of (3.22), appropriate model order cannot be inferred with confidence.

This learning strategy is by far the most widespread and is used in the vast majority of ICA algorithms, whether explicitly [66, 67, 68] or implicitly [45]. For extending ICA, however, ML is of little use as it cannot infer the posterior over model weights nor find the optimal latent dimensionality of an ICA model.

3.6.2 Maximum a posteriori

This is the next level of approximation to Bayesian inference. Maximum *a posteriori* (MAP) methods define a prior over the whole weights $\boldsymbol{W}$ but do not compute the difficult integral in (3.14). The MAP method finds point-estimates that maximise the numerator of (3.11)

$$\boldsymbol{W}_{MAP} = \arg\max_{\boldsymbol{W}} \left[p(\boldsymbol{X}|\boldsymbol{W}, \mathcal{M}) p(\boldsymbol{W}|\mathcal{M}) \right] \tag{3.24}$$

This is equivalent to finding (unnormalised) posteriors of the form (3.13). The use of priors can be used to preclude unrealistic parameter values. This does not mean, however, that the MAP estimate is the optimal one. Posterior densities will generally be products of two, assimilar densities. This leads to skewed

distributions where the peak is not aligned with the area of most posterior probability mass. This gives a biased estimate of the weights and will result in a sub-optimal model. Furthermore, the lack of a full posterior density again precludes model comparison and complexity control.

This is the method used by Knuth in his Bayesian approach to ICA [75]. With similar shortcomings to ML, this strategy must also be rejected.

3.6.3 Expectation-Maximisation algorithm

The Expectation-Maximisation (EM) algorithm [71] is an efficient optimisation method for maximising the likelihoods in the ML and MAP approximations above and is particularly useful if defining priors over the hidden variables in $\boldsymbol{W}$. If the hidden variables are denoted $\boldsymbol{H}$, then EM can find the full posterior densities over the hidden variables and point-estimates of $\boldsymbol{\Theta}$ that maximise the data log-likelihood conditioned on the parameters

$$\boldsymbol{\Theta}_{EM} = \arg\max_{\boldsymbol{\Theta}} \left[\log p(\boldsymbol{X}|\boldsymbol{\Theta}, \mathcal{M})\right] \tag{3.25}$$

This likelihood is not maximised directly. A more efficient method is used which finds the parameter values iteratively. A strict lower bound on the data log-likelihood is defined in terms of the posterior over the hidden variables

$$\mathcal{L}_{\mathcal{M}}(\boldsymbol{X}|\boldsymbol{\Theta}) \geq \left\langle \log p(\boldsymbol{X}, \boldsymbol{H}|\boldsymbol{\Theta}) \right\rangle_{p(\boldsymbol{H}|\boldsymbol{X}, \boldsymbol{\Theta}')} + \mathcal{H}[\boldsymbol{H}] \tag{3.26}$$

where $\boldsymbol{\Theta}'$ refers to the parameter values found in the previous iteration. The EM algorithm is a two-step procedure. The 'Expectation' step calculates the posterior over hidden variables given the parameter settings from the previous step and computes (3.26) using this posterior. The 'maximisation' step finds the optimal parameter values that maximise this expectation. This procedure can be shown to increase the log-likelihood in (3.25) [71].

The EM algorithm is just a method for performing ML or MAP efficiently so suffers from the same problems. Even if priors are defined over $\boldsymbol{H}$, no priors are defined over the parameters, $\boldsymbol{\Theta}$, so unlikely or unwanted parameter values are not discounted, and there is no integrating out of Markov blankets.

This algorithm was used by Attias for ICA in [46]. EM is an ML method, so cannot be used for model comparison or the incorporation of prior knowledge.

3.6.4 Evidence approximation

The 'evidence approximation' assumes the evidence is sharply peaked around the most probable values of the weights and that the evidence is well approximated by a Gaussian. The evidence is then approximately equal to the height of the peak times the volume of the (implicitly) approximating Gaussian

$$p(\boldsymbol{X}|\mathcal{M}) \approx p(\boldsymbol{X}|\hat{\boldsymbol{W}},\mathcal{M})p(\hat{\boldsymbol{W}}|\mathcal{M})(2\pi)^{\frac{k}{2}}|\det \boldsymbol{C}|^{-\frac{1}{2}} \tag{3.27}$$

where $\hat{\boldsymbol{W}}$ are the most probable weight values found from ML or MAP, and $\boldsymbol{C}^{-1}$ is the covariance matrix of the Gaussian approximation (the 'volume' term). The matrix $\boldsymbol{C}$ is the Hessian of the log-numerator in (3.11)

$$\boldsymbol{C} = -\nabla\nabla \log\left[p(\boldsymbol{X}|\boldsymbol{W},\mathcal{M})p(\boldsymbol{W}|\mathcal{M})\right] \tag{3.28}$$

By assuming an analytically tractable Gaussian approximation to the evidence model comparison and complexity control is now possible, although calculating the Hessian is expensive so this approximation scales poorly with dimensionality. More fundamentally, the evidence approximation assumes the posterior is well represented by a Gaussian. If the true posterior is skewed or highly non-Gaussian, then the expectation w.r.t. the approximation is not equivalent to the expectation w.r.t. the true posterior. Furthermore, the Gaussian approximation assumption stems from the CLT and is therefore only accurate when the number of data points is much greater than the number of parameters [99]. Consequently, measures that require maginalisation may not be estimated correctly.

This approximation was used by Roberts [74] in one of the first Bayesian learning formulations of ICA. One of the primary aims of this thesis is to construct an ICA model that can capture a wide variety of source densities. The inclusion of a large variety of non-Gaussian densities will inevitably lead to non-Gaussian posteriors which makes the evidence approximation irrelevant.

3.6.5 Monte Carlo methods

Although the integral in (3.12) is intractable for ICA, sampling methods can be used to numerically evaluate it. There are a wide range of practical sampling techniques under the banner of 'Monte Carlo methods' (see [100, 1] for an introduction) and they enjoy wide spread attention. The idea is to draw N samples, $\boldsymbol{W}' = \{\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_N\}$, from the weight posterior $p(\boldsymbol{W}|\boldsymbol{X}, \mathcal{M})$ and to approximate the evidence using

$$p(\boldsymbol{X}|\mathcal{M}) \approx \frac{1}{N} \sum_{n=1}^{N} p(\boldsymbol{X}|\boldsymbol{W}_n, \mathcal{M}) \tag{3.29}$$

The weight posterior need not be normalised. If $p(\boldsymbol{W}|\boldsymbol{X}, \mathcal{M}) = \frac{1}{Z} p^*(\boldsymbol{W}|\boldsymbol{X}, \mathcal{M})$, where $p^*(\boldsymbol{W}|\boldsymbol{X}, \mathcal{M})$ is an unnormalised distribution that can be evaluated and Z is the normalising constant, then $Z = \sum_{n=1}^{N} p^*(\boldsymbol{W}_n|\boldsymbol{X}, \mathcal{M})$.

However, this is easier said than done. Even if $p^*(\boldsymbol{W}|\boldsymbol{X}, \mathcal{M})$ can be evaluated at any given point, the amount of points needed to adequately cover the density is exponential in the dimensionality of the weight space. Also, high-dimensional probability distributions often have most of their mass concentrated in a small region known as the *typical set*, so uniformly drawing from the weight space at random points does not help as the probability of drawing from the typical set is small. Therefore, when the dimensionality of the weight space is large (as in the case of ICA), generating samples is difficult, even if using more sophisticated strategies such as importance and rejection sampling [100].

Sampling strategies do exist for high-dimensional distributions which sample preferentially from the typical set and are collectively called *Markov chain Monte Carlo* (MCMC) methods. An MCMC strategy draws a chain of samples from the distribution such that the part of the sample space a sample is drawn from depends on the locality of the previous sample. The idea is that as the number of samples in the chain increases, the collection of samples become more representitive of the probability distribution as a whole.

The simplest strategy is to draw succesive samples according to a random walk, i.e. $\boldsymbol{W}_{n+1} = \boldsymbol{W}_n + \epsilon$ where ϵ is a small random vector. The new samples

are then accepted or rejected according to some measure of whether they are in the typical set or not. The *Metropolis-Hastings* algorithm [101] employs the following strategy [1] to get representative samples

$$\text{if } p^*(\boldsymbol{W}_{n+1}|\boldsymbol{X}) > p^*(\boldsymbol{W}_n|\boldsymbol{X}) \text{ accept}$$
$$\text{if } p^*(\boldsymbol{W}_{n+1}|\boldsymbol{X}) < p^*(\boldsymbol{W}_n|\boldsymbol{X}) \text{ accept with probability } \frac{p^*(\boldsymbol{W}_{n+1}|\boldsymbol{X})}{p^*(\boldsymbol{W}_n|\boldsymbol{X})}$$

where the explicit dependence on $\mathcal{M}$ has been dropped for brevity.

Sampling, however, suffers from a number of problems. It does not yield a closed form solution. It is also difficult to evaluate when the Markov chain has converged and enough representitive samples have been drawn. Sampling is also inefficient at representing densities as all the points have to be stored. Finally, the sampling procedure is computationally intensive for models with many nodes, even using MCMC. For these reasons, such a framework must be rejected for practical Bayesian ICA.

3.7 Approximations to Model Comparison

The ability to compare models is one of the key benefits of Bayesian inference. However, Bayesian inference is NP-hard so the approximations discussed above are widely utilised. Where these approximations retain the essence of Bayesian inference - marginalisation over unknown weights - model comparison is still feasible, as in the case of the evidence approximation and Bayesian sampling regimes. The ML and MAP approximations, however, are too severe for direct model comparison, so likelihood increases with complexity. To combat this, methods can be used to add penalty terms linked to model complexity, such as the Maximum Description Length (MDL) [97] or Bayesian Information Criterion (BIC) [102] penalties. Both these penalties are equivalent with BIC = -MDL. These introduce a penalty to the data log-likelihood linked to the number of model parameters and the size of the dataset. In BIC form, this is

$$BIC(\boldsymbol{X}|\mathcal{M}) = \sum_{t=1}^{T} \log p(\boldsymbol{X}|\hat{\boldsymbol{W}}, \mathcal{M}) - \frac{|\mathcal{M}|}{2} \log T \tag{3.30}$$

where $\hat{\boldsymbol{W}}$ are the most probable weight values found from ML or MAP, $|\mathcal{M}|$ is the total number of parameters in the model and T is the number of data point. This relationship can be derived from the evidence approximation in the large sample limit. As such, its comparitive power is limited to datasets of large sizes (see section 4.4.5).

Although limited in use, MDL does give an insight to what the evidence inherently measures. As implied by its name, the Minimum Description Length estimates the least number of bits the given model needs for encoding the data. In fact, the negative log evidence of a model, $-\log_2 p(\boldsymbol{X}|\mathcal{M})$, is *precisely* the number of bits need to encode $\boldsymbol{X}$ using $\mathcal{M}$ [103].

3.8 Summary

Many levels of inference are possible, once assumptions, beliefs and hypothesises are quantified as prior probabilities under a Bayesian framework. Once made explicit, these beliefs can be incorporated into statistical models allowing a great deal of information to be dynamically processed. The incorporation of prior knowledge and constraints about weight values leads to a natural regularisation. The effect of quantified assumptions on inferences and predictions can be observed, and this allows quantative model comparison.

Using Bayesian learning directly, however, is computationally intractable for all but the simplest models. Consequently, approximations have to be introduced which can greatly reduce the inherent power of the Bayesian formalism. The most popular approximations have been examined and rejected for use in a Bayesian ICA model. A new approximation method - the *variational* framework - has recently been successfully applied without introducing the pathologies of the methods investigated above. Originally used by statistical physicists to model gases and systems of particles, this closed-form approximation is born from the calculus of variations [104]. Its flexibility and relative efficiency make it an ideal method for allowing practical Bayesian ICA. This framework will be explored in the next Chapter.

Chapter 4

Variational Approximation

Although Bayesian inference and learning is highly desirable, it is impractical for most applications, including ICA. There are many avenues one can take, most of which were explored and rejected in Chapter 3. A more attractive approach is the variational approximation which has been shown to be a very flexible and efficient closed-form approximation to intractable Bayesian learning [72].

The variational method to approximating intractable computations has its roots in the 'calculus of variations' [104] and encompasses a whole gamut of tools for evaluating integrals and functionals. The method employed here uses the 'mean-field' variational approximation popular in statistical physics [105]. In the context of Bayesian learning, this learning framework is known as 'variational Bayes', 'free-energy minimisation' or 'ensemble learning'. The central idea is to introduce a set of approximating densities to the posteriors over the weight-sets, and to introduce them in such a way as to make their evaluation tractable. These approximations are then optimised so as to minimise the discrepency between them and the true posteriors using some measure of the difference. The optimisation is carried out by varying the functional parameters of these approximations, thus giving the approximation its name.

The variational Bayesian approximation derives from mean-field theories in statistical mechanics [105] used in calculating the free-energy of a system. The mean-field approximation as applied to neural networks was first shown in [106], but really came to the fore in probabilistic models in [107] for approximat-

ing intractable maximum likelihood functions. This method was subsequently applied to Bayesian parameter estimation and model comparison by MacKay [108], where it was termed ensemble learning. The variational approximation to Bayesian learning usually comes under the moniker of *variational Bayes* and has quickly become a popular way to learn otherwise intractable models [109, 110, 111].

This Chapter starts with a derivation of the variational Bayes approximation based on the Kullback-Leibler divergence between true and approximating distributions. The Chapter explores the practical issues of using the variational framework and concludes with an example of variational Bayesian learning and inferring structure from data in section 4.5.

4.1 Derivation

Although the 'variation' in variational originates from the calculus of variations, the following derivation is more intuitive in a probabilistic sense. Consider the log marginal likelihood for data $\boldsymbol{X}$

$$\log p(\boldsymbol{X}) = \log \frac{p(\boldsymbol{X}, \boldsymbol{W})}{p(\boldsymbol{W}|\boldsymbol{X})} \tag{4.1}$$

which simply follows from the definition of conditional probability and where the explicit dependence on model assumptions has been dropped for brevity. The term $\boldsymbol{W}$ is the vector of all hidden variables and unknown parameters. As the log evidence does not depend on $\boldsymbol{W}$, this can be re-written as

$$\begin{aligned} \log p(\boldsymbol{X}) &= \int p'(\boldsymbol{W}) \log \frac{p'(\boldsymbol{W})}{p'(\boldsymbol{W})} \frac{p(\boldsymbol{X}, \boldsymbol{W})}{p(\boldsymbol{W}|\boldsymbol{X})} d\boldsymbol{W} \\ &= \int p'(\boldsymbol{W}) \log \frac{p(\boldsymbol{X}, \boldsymbol{W})}{p'(\boldsymbol{W})} d\boldsymbol{W} + \int p'(\boldsymbol{W}) \log \frac{p'(\boldsymbol{W})}{p(\boldsymbol{W}|\boldsymbol{X})} d\boldsymbol{W} \\ &= F[\boldsymbol{X}] + KL[p'(\boldsymbol{W}) \| p(\boldsymbol{W}|\boldsymbol{X})] \end{aligned} \tag{4.2}$$

where $p'(\boldsymbol{W})$ is some approximation to the posterior $p(\boldsymbol{W}|\boldsymbol{X})$ and where

$$F[\boldsymbol{X}] = \langle \log p(\boldsymbol{X}, \boldsymbol{W}) \rangle_{p'(\boldsymbol{W})} + \mathcal{H}[\boldsymbol{W}] \tag{4.3}$$

$$KL[p'(\boldsymbol{W}) \| p(\boldsymbol{W}|\boldsymbol{X})] = \int p'(\boldsymbol{W}) \log \frac{p'(\boldsymbol{W})}{p(\boldsymbol{W}|\boldsymbol{X})} d\boldsymbol{W} \tag{4.4}$$

$\mathcal{H}[\boldsymbol{W}]$ is the entropy of $p'(\boldsymbol{W})$. The first term in (4.2), F, is known as the negative variational free energy (a term derived from statistical physics). The second term is the Kullback-Leibler divergence defined in Chapter 1 and introduced in section 2.3.3, a pseudo-distance that measures the difference between two densities. This term is strictly non-negative which means that F is a *strict lower bound* on the log evidence

$$\log p(\boldsymbol{X}) \geq F[\boldsymbol{X}] \tag{4.5}$$

with equality iff the approximating density $p'(\boldsymbol{W})$ equals the true posterior $p(\boldsymbol{W}|\boldsymbol{X})$.

Ideally, one would like to minimise the KL-divergence between the approximating and true posteriors directly, but this is not possible as the true posterior is not known. However, note the log evidence on the LHS of (4.2) is not dependent on the weights, $\boldsymbol{W}$, and is therefore a constant w.r.t. $\boldsymbol{W}$. Therefore, maximising F w.r.t. the approximating posterior $p'(\boldsymbol{W})$ will *necessarily* minimise the KL-divergence between the approximating and true posteriors. By choosing an appropriate factorised form for the approximation $p'(\boldsymbol{W})$, F can be maximised separately w.r.t. each weight group w_i, implicitly integrating out all other weight groups and forcing the approximation towards the true posterior. In this way, closed-form tractable Bayesian learning is performed by both calculating the weight posteriors and computing the (log) evidence. Furthermore, this approximation does not have to be limited to Gaussian form, overcoming this limitation in the evidence approximation.

Note the similarity of (4.3) to (3.26). The variational approximation can be seen as a generalisation of the EM algorithm, where the posterior of the model parameters is also calculated. Unlike the EM approach, however, F is a strict lower bound to the model log-evidence, so a wide variety of models and assumptions can be compared and contrasted by calculating the free energy of each model. The higher the (negative variational) free energy, the higher the likelihood of the data under that model, and, therefore, the better that model

is at explaining the data.

Of course, using the negative free energy assumes the bound on the evidence is close enough to follow the shape of the evidence curve across various assumptions. Miskin has showed [112] that - for simple models - the bound is indistinguishable from the evidence calculated via importance sampling. Miskin also evaluated the difference for a Bayesian ICA model (a unimodal version of the vbICA1 model im Chapter 5). For small amounts of data (100 data vectors), the negative free energy was approximately 10-15% below the evidence calculated by importance sampling. This is because the factorised approximation to the posterior breaks more correlations in complicated models and is therefore more severe. However, the bound still followed the shape of the evidence very closely allowing model comparison to still be quantifiably and succesfully carried out . For larger datasets (1000 data vectors) the bound was again indistinguishable from the evidence calculated from sampling.

4.2 Maximising the Objective Function

The objective function to be maximised in variational Bayesian learning is the negative variational free energy (NFE), $F[\boldsymbol{X}]$

$$F\left[\boldsymbol{X}\right] = \left\langle \log p(\boldsymbol{X}, \boldsymbol{W}) \right\rangle_{p'(\boldsymbol{W})} + \mathcal{H}\left[\boldsymbol{W}\right] \tag{4.6}$$

The approximation $p'(\boldsymbol{W})$ is introduced to allow a closed form solution to the posterior. This in itself does not ensure the solution is tractable. The main barrier to tractability is the correlations induced within the Markov blanket of a variable when Bayes' theorem is invoked. To overcome this, a factorisation of the posterior $p'(\boldsymbol{W})$ is enforced, for example

$$p'_{\phi}(\boldsymbol{W}) = \prod_i p'_{\phi_i}(\boldsymbol{w}_i) \tag{4.7}$$

where ϕ_i are the parameters for density i. This factorisation states that the joint posterior over all the weights is equivalent to a product of their marginal posteriors. This allows each marginal to be approximated individually without

having to go through the intermediate step of (3.13) and necessarily breaks some of the induced correlations. The trick is choosing a factorisation that ignores weak correlations but preserves stronger ones. How one chooses an appropriate factorisation, however, is more art than science. The factorisation is usually taken to be the same as that of the priors as generally this is the easiest to manipulate, both mathematically and computationally. The effect of two alternative factorisations for ICA will be explored in the next Chapter.

One may now proceed by specifying functional forms of each of the approximating posteriors and using these in (4.6). As shown in [108], however, there is no need to specify functional forms for the posteriors if conjugate forms for the densities are chosen - they 'fall-out' of the maximisation process. Families of conjugate functions are such that, when member functions are multiplied together, they give a function in the same family. Substituting (4.7) into (4.6) gives

$$F\left[\boldsymbol{X}\right] = \langle \log p(\boldsymbol{X}, \boldsymbol{W}) \rangle_{\prod_i p'(w_i)} + \sum_i \mathcal{H}\left[w_i\right] \tag{4.8}$$

The optimal form for $p'(\boldsymbol{W})$ is found by differentiating (4.8) w.r.t. $p'(w_i)$

$$\frac{\partial F}{\partial p'(\boldsymbol{w}_i)} = \langle \log p(\boldsymbol{X}, \boldsymbol{W}) \rangle_{\prod_{j \neq i} p'(\boldsymbol{w}_{j \neq i})} - \log p'(\boldsymbol{w}_i) - 1 + \lambda_i \tag{4.9}$$

where the Lagrange multiplier λ_i is a constant that ensures that $p'(\boldsymbol{w_i})$ is normalised. Setting (4.9) to zero and rearranging gives

$$p'(\boldsymbol{w}_i) = \frac{1}{Z_i} \exp\left[\langle \log p(\boldsymbol{X}, \boldsymbol{W}) \rangle_{\prod_{j \neq i} p'(\boldsymbol{w}_{j \neq i})} \right] \tag{4.10}$$

where Z_i is the normalising factor defined by the constants in (4.9). If conjugate forms for the prior and data likelihoods are chosen, then (4.10) will be analytic and the posterior will have a similar functional form. For example, if a Gaussian is chosen such that $p(w_i|\mathcal{M}) = \mathcal{N}(w_i; \mu_i, \beta_i)$, and similarly for the data likelihood, then $p'(w_i) = \mathcal{N}(w_i; \hat{\mu}_i, \hat{\beta}_i)$ where the posterior parameters are functions of the terms on the RHS of (4.10). In general, all the posterior parameters will be given in terms of the corresponding prior parameters, data, and expectations of hidden variables under other posteriors. Equation (4.10) will

therefore have to be iterated for all i. This procedure can be shown to increase the NFE monotonically and is guaranteed to converge [110].

All the ingredients are now ready to mix and bake in Thomas Bayes' oven. The following section gives the recipe.

4.3 Variational Method for Generative Models

The variational approach for inference (which, as previously stated, includes learning), in a model $\mathcal{M}$ represented by a graphical model $\mathcal{G} = \{V, E\}$, then proceeds as follows:

1. Specify the structure $\mathcal{G}$. Identify which nodes are visible ($\boldsymbol{X}$), which are hidden ($\boldsymbol{H}$) and which are parameters ($\boldsymbol{\Theta}$). Let the vector $\boldsymbol{W}$ represent both $\boldsymbol{H}$ and $\boldsymbol{\Theta}$.

2. Read off the joint probability density, $P(\boldsymbol{X}, \boldsymbol{W}|\mathcal{M})$, from the structure using (3.9). The conditional term $\mathcal{M}$ represents beliefs and assumptions (e.g. the form of the structure, the hyper-parameters etc.).

3. Choose a form for the approximating posterior $p'(\boldsymbol{W})$

$$p'(\boldsymbol{W}) = \prod_i p'(w_i) \tag{4.11}$$

4. Substitute $P(\boldsymbol{X}, \boldsymbol{W}|\mathcal{M})$ and $p'(\boldsymbol{W})$ into (4.6), and maximise by solving (4.10) for all i, cycling until convergence.

5. The final posteriors can be substituted into (4.6) to compute the free energy bound, These scores can then be used for model comparison via (3.22).

4.4 An Example - Mixture of Gaussians

This section presents an example of variational Bayesian learning. A useful tool in pattern recognition - and in the next Chapter - is the Mixture of Gaussians (MoG) model. If a sequence of numbers is exhaustively described by its probability distribution, then describing this distribution is an important part of

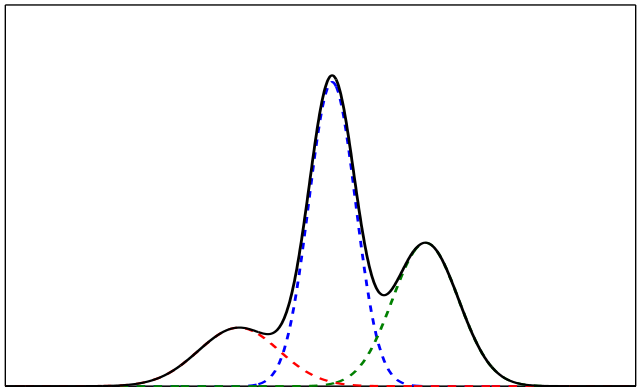


Figure 4.1: Density Modelling using a MoG.

analysing that sequence. MoGs are particularly useful for density modelling as their Gaussian nature makes them straightforward and efficient to work with. Furthermore, Gaussians are to probability densities as sines and cosines are to periodic signals - potentially any distribution can be captured given enough Gaussians. Figure 4.1 shows how just three Gaussians can model a complex, multi-modal distribution. For this reason, MoGs will be fundamental in the powerful and flexible ICA model developed in the next Chapter. This example concentrates on 1-dimensional Gaussians; it is straightforward to extend the formalism to multivariate Gaussians [113].

4.4.1 The model

The probability of generating a data point s^t from a m-component mixture model given assumptions $\mathcal{M}$ is:

$$p(s^t|\mathcal{M}) = \sum_{q=1}^{m} p(q|\mathcal{M}_0)p(s^t|q,\mathcal{M}_q) \tag{4.12}$$

A data vector is generated by choosing one of the m components stochastically under $p(q|\mathcal{M}_0)$ and then drawing from $p(s^t|\mathcal{M}_q, q)$. $\mathcal{M} = \{\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_m\}$ is the vector of component model assumptions, $\mathcal{M}_q$, and assumptions about the mixture process, $\mathcal{M}_0$. The assumptions represent everything that essentially defines the model - values of fixed parameters, model structure, details of the component switching method, any prior information etc. $p(s^t|\mathcal{M})$ is the evi-

dence for model $\mathcal{M}$ and quantifies the likelihood of the observed data under model $\mathcal{M}$. If the mixture is adapted through a maximum likelihood approach then $\mathcal{M}$ represents a list of point estimates for the corresponding parameters. In a Bayesian setting, the assumptions represent information concerning the *distribution* of parameters.

The variable q indicates which component of the mixture model is chosen to generate a given data point s. If $p(q|\mathcal{M}_0)$ is a vector of probabilities and each component $p(s^t|\mathcal{M}_q, q)$ is a Gaussian, then (4.12) describes a Mixture of Gaussians model

$$\begin{aligned} p(s^t|\theta) &= \sum_{q=1}^{m} p(q^t = q|\boldsymbol{\pi}) p(s^t|q^t, \mu_q, \beta_q) \\ &= \sum_{q=1}^{m} \pi_q \mathcal{N}(s^t; \mu_q, \beta_q) \end{aligned} \tag{4.13}$$

where the explicit dependence on the model $\mathcal{M}$ has been dropped for brevity. The variable q is an indicator variable indicating which Gaussian component is chosen for generating s and takes on values of $\{q = 1, q = 2, \ldots, q = m\}$. π_q are the mixing proportions and are equivalent to $p(q^t = q|\boldsymbol{\pi})$, the prior probability of choosing component q. The mean and precision of component q is μ_q and β_q respectively. The vectors of component parameters can be written as $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_m]$, $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_m]$ and $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_m]$. The complete parameter set is $\theta = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\beta}\}$. Figure 4.2 shows a schematic diagram for the MoG model. Circular nodes represent random variables, square nodes are assumptions and rounded rectangles represent the Gaussian components.

The probability of state q being chosen and generating observation sequence $\boldsymbol{S} = \{s^1, \ldots, s^T\}$ is

$$\begin{aligned} p(\boldsymbol{S}, \boldsymbol{q}|\theta) &= \prod_{t=1}^{T} p(q^t = q|\boldsymbol{\pi}) p(s^t|q, \mu_q, \beta_q) \\ &= \prod_{t=1}^{T} \pi_q \mathcal{N}(s^t; \mu_q, \beta_q) \end{aligned} \tag{4.14}$$

where $\boldsymbol{q} = \{q^1, \ldots, q^T\}$.

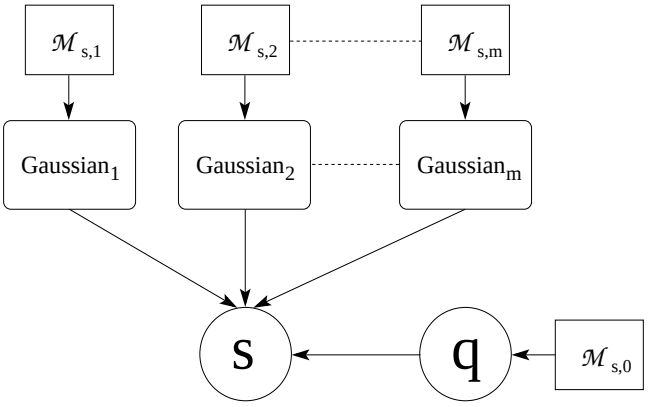


Figure 4.2: Mixture of Gaussians model.

4.4.2 Variational Bayesian learning for a MoG

First, the distribution over the hidden variables and the priors over the parameters are stated. The distribution over the MoG component indicator variables is

$$p(\boldsymbol{q}|\boldsymbol{\pi}) = \prod_{t=1}^{T} \pi_q \tag{4.15}$$

The prior over the model parameters is

$$p(\theta) = p(\boldsymbol{\pi})p(\boldsymbol{\mu})p(\boldsymbol{\beta}) \tag{4.16}$$

The prior over the mixture proportions is a symmetric Dirichlet ($\lambda_q = \lambda_0$ for all q)

$$p(\boldsymbol{\pi}) = \mathcal{D}(\boldsymbol{\pi}; \lambda_0) \tag{4.17}$$

The prior over the means is a product of Gaussians

$$p(\boldsymbol{\mu}) = \prod_{q=1}^{m} \mathcal{N}(\mu_q; m_0, \tau_0) \tag{4.18}$$

The prior over the precisions is a product of Gammas

$$p(\boldsymbol{\beta}) = \prod_{q=1}^{m} \mathcal{G}(\beta_q; b_0, c_0) \tag{4.19}$$

Figure 4.3 shows a graphical model for this Bayesian MoG.

The objective function to be maximised is the negative free energy, F:

$$F = \langle \log p(\boldsymbol{S}, \boldsymbol{W}) \rangle_{p'(\boldsymbol{W})} + \mathcal{H}\left[p'(\boldsymbol{W})\right] \tag{4.20}$$

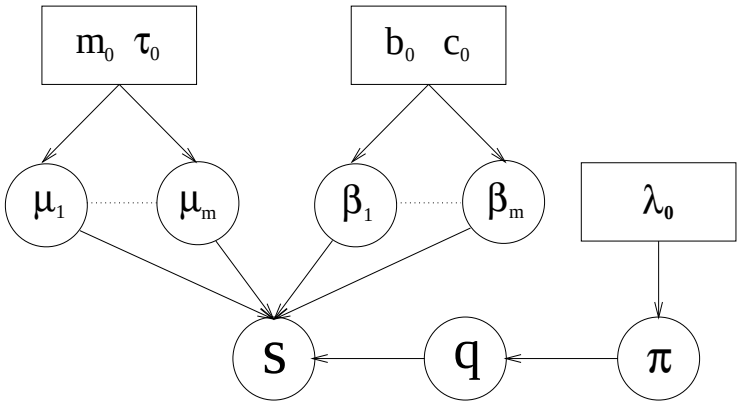


Figure 4.3: Graphical model of Bayesian MoG.

where $\boldsymbol{W} = \{\boldsymbol{q}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\beta}\}$, the ensemble of hidden variables and parameters. By choosing $p'(\boldsymbol{W})$ such that it factorises over the ensemble of hidden variables, $\boldsymbol{W}$, terms in each hidden variable can be maximised individually

$$p'(\boldsymbol{W}) = p'(\boldsymbol{q})p'(\boldsymbol{\pi})p'(\boldsymbol{\mu})p'(\boldsymbol{\beta}) \tag{4.21}$$

By substituting $p(\boldsymbol{S}, \boldsymbol{W})$ and (4.21) into (4.20), one can obtain an expression for the negative free energy of the model. The term $p(\boldsymbol{S}, \boldsymbol{W})$ is straight forward to write down from inspection of the graphical model

$$p(\boldsymbol{S}, \boldsymbol{W}) = p(\boldsymbol{S}|\boldsymbol{q}, \boldsymbol{\mu}, \boldsymbol{\beta})p(\boldsymbol{q}|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\mu})p(\boldsymbol{\beta}) \tag{4.22}$$

where

$$p(\boldsymbol{S}|\boldsymbol{q}, \boldsymbol{\mu}, \boldsymbol{\beta}) = \prod_{t=1}^{T} p(s^t|q^t, \mu_q, \beta_q) \tag{4.23}$$

The negative free energy for the model is now

$$\begin{aligned} F \quad = \quad & \langle \log p(\boldsymbol{S}|\boldsymbol{q}, \boldsymbol{\mu}, \boldsymbol{\beta}) \rangle_{p'(\boldsymbol{q})p'(\boldsymbol{\mu})p'(\boldsymbol{\beta})} \\ & + \langle \log p(\boldsymbol{q}|\boldsymbol{\pi}) \rangle_{p'(\boldsymbol{q})p'(\boldsymbol{\pi})} + \mathcal{H}\left[p'(\boldsymbol{q})\right] \\ & + \langle \log p(\boldsymbol{\pi}) \rangle_{p'(\boldsymbol{\pi})} + \mathcal{H}\left[p'(\boldsymbol{\pi})\right] \\ & + \langle \log p(\boldsymbol{\mu}) \rangle_{p'(\boldsymbol{\mu})} + \mathcal{H}\left[p'(\boldsymbol{\mu})\right] \\ & + \langle \log p(\boldsymbol{\beta}) \rangle_{p'(\boldsymbol{\beta})} + \mathcal{H}\left[p'(\boldsymbol{\beta})\right] \end{aligned} \tag{4.24}$$

4.4.3 Optimising the posteriors

The optimum posteriors are found by using (4.10). The factorisation defined by (4.21) allows each posterior to be optimised individually.

$p'(\boldsymbol{\mu})$

Using (4.10), the posterior over the component means $\boldsymbol{\mu}$ is given by

$$\log p'(\boldsymbol{\mu}) \propto \langle \log p(\boldsymbol{S}|\boldsymbol{q}, \boldsymbol{\mu}, \boldsymbol{\beta}) \rangle_{p'(\boldsymbol{q})p'(\boldsymbol{\beta})} + \log p(\boldsymbol{\mu}) \tag{4.25}$$

Substituting (4.23) and (4.18) into (4.25) gives

$$\begin{aligned} \log p'(\boldsymbol{\mu}) &\propto \sum_{t=1}^{T}\sum_{q=1}^{m} p'(\boldsymbol{q}) \left[\frac{1}{2}\langle \log \beta_q \rangle - \frac{\langle \beta_q \rangle}{2} \left(s^{t\,2} - 2s^t\mu_q + \mu_q^2 \right) \right] \\ &+ \sum_{q=1}^{m} \frac{1}{2}\log \tau_0 - \frac{\tau_0}{2}\left(\mu_q^2 - 2\mu_q m_0 + m_0^2\right) \end{aligned} \tag{4.26}$$

where $\langle a \rangle$ is the expectation w.r.t. $p'(a)$. Collecting together terms in μ_q and defining

$$p'(q^t = q) \doteq \hat{\gamma}_q^t \tag{4.27}$$

gives

$$\begin{aligned} \log p'(\boldsymbol{\mu}) &\propto \sum_{q=1}^{m} -\frac{1}{2}\left[\left(\tau_0 + \langle \beta_q \rangle \sum_{t=1}^{T} \hat{\gamma}_q^t \right) \mu_q^2 - 2\left(\tau_0 m_0 + \langle \beta_q \rangle \sum_{t=1}^{T} \hat{\gamma}_q^t s^t \right) \mu_q \right] \\ &\propto \sum_{q=1}^{m} -\frac{1}{2}\left(\tau_0 + \langle \beta_q \rangle \sum_{t=1}^{T} \hat{\gamma}_q^t \right) \left[\mu_q^2 - 2\frac{\tau_0 m_0 + \langle \beta_q \rangle \sum_t \hat{\gamma}_q^t s^t}{\tau_0 + \langle \beta_q \rangle \sum_t \hat{\gamma}_q^t} \mu_q \right] \end{aligned} \tag{4.28}$$

As $\log p'(\boldsymbol{\mu})$ is a sum of m quadratics in μ_q, (4.28) implies $p'(\boldsymbol{\mu})$ is a product of m Gaussian densities

$$p'(\boldsymbol{\mu}) = \prod_{q=1}^{m} \mathcal{N}(\mu_q; \hat{m}_q, \hat{\tau}_q) \tag{4.29}$$

where

$$\hat{m}_q = \frac{1}{\hat{\tau}_q}\left(\tau_0 m_0 + \langle \beta_q \rangle \sum_{t=1}^{T} \hat{\gamma}_q^t s^t \right) \tag{4.30}$$

$$\hat{\tau}_q = \tau_0 + \langle \beta_q \rangle \sum_{t=1}^{T} \hat{\gamma}_q^t \tag{4.31}$$

The expectation of μ_q is $\hat{m}_q$. Note that in the limit of uninformative priors ($\tau_0 \to 0$) and/or infinite data, the above update equations yield the maximum likelihood solutions

$$\begin{aligned} \langle \mu_q \rangle &\to \frac{\sum_{t=1}^{T} \hat{\gamma}_q^t s^t}{\sum_{t=1}^{T} \hat{\gamma}_q^t} && (4.32) \\ \langle \mu_q^2 \rangle &\to \langle \mu_q \rangle^2 && (4.33) \end{aligned}$$

The maximum likelihood solution is also the limiting case for the rest of the updates.

$p'(\boldsymbol{\beta})$

The posterior over the component precisions $\boldsymbol{\beta}$ is given by

$$\log p'(\boldsymbol{\beta}) \propto \langle \log p(\boldsymbol{S}|\boldsymbol{q}, \boldsymbol{\mu}, \boldsymbol{\beta}) \rangle_{p'(\boldsymbol{q})p'(\boldsymbol{\mu})} + \log p(\boldsymbol{\beta}) \tag{4.34}$$

Substituting (4.23) and (4.19) into (4.34) gives

$$\begin{aligned} \log p'(\boldsymbol{\beta}) &\propto \sum_{t=1}^{T} \sum_{q=1}^{m} p'(\boldsymbol{q}) \left[\frac{1}{2} \log \beta_q - \frac{\beta_q}{2} \left(s^{t\,2} - 2 s^t \langle \mu_q \rangle + \langle \mu_q^2 \rangle \right) \right] \\ &+ \sum_{q=1}^{m} (c_0 - 1) \log \beta_q - \frac{\beta_q}{b_0} \end{aligned} \tag{4.35}$$

Collecting together terms in β_q and using (4.27) gives

$$\begin{aligned} \log p'(\boldsymbol{\beta}) &\propto \sum_{q=1}^{m} \left[\left(c_0 + \frac{1}{2} \sum_{t=1}^{T} \hat{\gamma}_q^t \right) - 1 \right] \log \beta_q \\ &\quad - \left[\frac{1}{b_0} + \frac{1}{2} \sum_{t=1}^{T} \hat{\gamma}_q^t \langle \left(s^t - \mu_q \right)^2 \rangle \right] \beta_q \end{aligned} \tag{4.36}$$

The functional form of (4.36) implies $p'(\boldsymbol{\beta})$ is a product of m Gamma distributions

$$p'(\boldsymbol{\beta}) = \prod_{q=1}^{m} \mathcal{G}(\beta_q; \hat{b}_q, \hat{c}_q) \tag{4.37}$$

where

$$\begin{aligned} \hat{b}_q &= \left[\frac{1}{b_0} + \frac{1}{2} \sum_{t=1}^{T} \hat{\gamma}_q^t \langle \left(s^t - \mu_q \right)^2 \rangle \right]^{-1} && (4.38) \\ \hat{c}_q &= c_0 + \frac{1}{2} \sum_{t=1}^{T} \hat{\gamma}_q^t && (4.39) \end{aligned}$$

$p'(\boldsymbol{\pi})$

The posterior over the indicator priors $\boldsymbol{\pi}$ is given by

$$\log p'(\boldsymbol{\pi}) \propto \langle \log p(\boldsymbol{q}|\boldsymbol{\pi}) \rangle_{p'(\boldsymbol{q})} + \log p(\boldsymbol{\pi}) \tag{4.40}$$

Substituting (4.15) and (4.17) into (4.40) gives

$$\begin{aligned} \log p'(\boldsymbol{\pi}) &\propto \sum_{q=1}^{m} \langle \log \pi_q \rangle_{p'(\boldsymbol{q})} + \sum_{q=1}^{m} (\lambda_0 - 1) \log \pi_q \\ &\propto \sum_{q=1}^{m} \left[\left(\lambda_0 + \sum_{t=1}^{T} \hat{\gamma}_q^t \right) - 1 \right] \log \pi_q \end{aligned} \tag{4.41}$$

The functional form of (4.41) implies a non-symmetric Dirichlet for $p'(\boldsymbol{\pi})$

$$p(\boldsymbol{\pi}) = \mathcal{D}(\boldsymbol{\pi}; \hat{\boldsymbol{\lambda}}_{1:m}) \tag{4.42}$$

where for component q

$$\hat{\lambda}_q = \lambda_0 + \sum_{t=1}^{T} \hat{\gamma}_q^t \tag{4.43}$$

$p'(\boldsymbol{q})$

Using (4.10), the posterior over the indicator variable q is given by

$$\log p'(\boldsymbol{q}) \propto \langle \log p(\boldsymbol{S}|\boldsymbol{q}, \boldsymbol{\mu}, \boldsymbol{\beta}) \rangle_{p'(\boldsymbol{\mu}) p'(\boldsymbol{\beta})} + \langle \log p(\boldsymbol{q}|\boldsymbol{\pi}) \rangle_{p'(\boldsymbol{\pi})} \tag{4.44}$$

Substituting (4.14) into (4.44) gives

$$\begin{aligned} \log p'(\boldsymbol{q}) &\propto \sum_{t=1}^{T} \frac{1}{2} \langle \log \beta_q \rangle - \frac{\langle \beta_q \rangle}{2} \left({s^t}^2 - 2 s^t \langle \mu_q \rangle + \langle \mu_q^2 \rangle \right) \\ &+ \sum_{t=1}^{T} \langle \log \pi_q \rangle \end{aligned} \tag{4.45}$$

Exponentiating (4.45) yields the posterior over $\boldsymbol{q}$

$$p'(\boldsymbol{q}) = \prod_{t=1}^{T} \hat{\gamma}_q^t \tag{4.46}$$

where

$$\gamma_q^t = \tilde{\pi}_q \tilde{\beta}_q^{\frac{1}{2}} \exp \left[-\frac{\langle \beta_q \rangle}{2} \langle \left(s^t - \mu_q \right)^2 \rangle \right] \tag{4.47}$$

$$\hat{\gamma}_q^t = \frac{\gamma_q^t}{\sum_{q'} \gamma_{q'}^t} \tag{4.48}$$

Equation (4.48) ensures that $\sum_q \hat{\gamma}_q^t = 1$. The tilded variables are exponentiated versions of $\langle \log \beta_q \rangle$ and $\langle \log \pi_q \rangle$ (under their respective posteriors) and are given by

$$\tilde{\pi}_q = \exp\left[\Psi(\hat{\lambda}_q) - \Psi(\sum_{q'} \hat{\lambda}_{q'})\right] \tag{4.49}$$

$$\tilde{\beta}_q = \hat{b}_q \exp\left[\Psi(\hat{c}_q)\right] \tag{4.50}$$

where $\Psi(.)$ is the Digamma function and is defined as

$$\Psi(x) \doteq \frac{\partial}{\partial x} \log \Gamma(x) \tag{4.51}$$

Appendix A shows how the expectations under relevant densities are computed.

Although no functional forms for the approximating posteriors were assumed, the resultant posteriors are conjugate with the priors. Intuitively, the (approximate) posterior is equivalent to shifting the prior to a position which satifies both the demands of the data and the constraints of the prior.

The update equations (4.30)-(4.48) are coupled and therefore must be solved iteratively. This is achieved by starting with initial guesses of the variables and cycling through the update equations using the moments calculated until convergence. Often in practice, the updates are performed in a 'variational Expectation-Maximisation' way. The expectations of hidden variables are first calculated using an initial guess of the parameter values. In the MoG example above, this means computing $p'(\boldsymbol{q})$. These expectations are then used in estimating the hidden parameters. For the MoG, this is equivalent to cycling through $p'(\boldsymbol{\mu})$, $p'(\boldsymbol{\beta})$ and $p'(\boldsymbol{\pi})$ given above.

4.4.4 Evaluating the negative free energy

The negative free energy in (4.24) can be rewritten using (1.18) in terms of the Kullback-Liebler divergences between the prior and posterior parameter distributions

$$\begin{aligned} F &= \langle \log p(\boldsymbol{S}, \boldsymbol{q}|\boldsymbol{\theta}) \rangle_{p'(\boldsymbol{q})p'(\boldsymbol{\theta})} + \mathcal{H}\left[p'(\boldsymbol{q})\right] - KL\left[p'(\boldsymbol{\theta}) \| p(\boldsymbol{\theta})\right] \\ &= L_{av} + \mathcal{H}\left[p'(\boldsymbol{q})\right] - KL[\boldsymbol{\pi}] - KL[\boldsymbol{\mu}] - KL[\boldsymbol{\beta}] \end{aligned} \tag{4.52}$$

where L_{av} is the average joint-likelihood of the observations and the hidden variables

$$L_{av} = \langle \log p(\boldsymbol{S}, \boldsymbol{q}|\boldsymbol{\theta}) \rangle_{p'(\boldsymbol{q})p'(\boldsymbol{\theta})} \tag{4.53}$$

Substituting the optimised posteriors and the appropriate KL-divergences from Appendix A into (4.52) yields an expression for the negative free energy, F.

Average Likelihood

The joint-likelihood averaged over the posteriors is straight forward to write down

$$\begin{aligned} L_{av} &= \langle \log p(\boldsymbol{S}, \boldsymbol{q}|\boldsymbol{\theta}) \rangle_{p'(\boldsymbol{q})p'(\boldsymbol{\theta})} \\ &= \left\langle \sum_{t=1}^{T}\sum_{q=1}^{m} \log \pi_q + \frac{1}{2}\log \beta_q - \frac{\beta_q}{2}\langle \left(s^t - \mu_q\right)^2 \rangle - \frac{1}{2}\log 2\pi \right\rangle_{p'(\boldsymbol{q})p'(\boldsymbol{\theta})} \\ &= \sum_{t=1}^{T}\sum_{q=1}^{m} \hat{\gamma}_q^t \left[\log \tilde{\pi}_q + \frac{1}{2}\log \tilde{\beta}_q - \frac{\langle \beta_q \rangle}{2}\langle \left(s^t - \mu_q\right)^2 \rangle_{p'(\boldsymbol{\mu})} \right] \\ &- \frac{T}{2}\log 2\pi \end{aligned} \tag{4.54}$$

Substituting (4.49) and (4.50) into (4.54) gives

$$L_{av} = \mathcal{A} + \mathcal{B} - \mathcal{C} - \frac{T}{2}\log 2\pi \tag{4.55}$$

where

$$\mathcal{A} = \sum_{t=1}^{T}\sum_{q=1}^{m} \hat{\gamma}_q^t \left[\Psi\left(\hat{\lambda}_q\right) - \Psi\left(\sum_{q'} \hat{\lambda}_{q'}\right) \right] \tag{4.56}$$

$$\mathcal{B} = \frac{1}{2}\sum_{t=1}^{T}\sum_{q=1}^{m} \hat{\gamma}_q^t \left[\Psi\left(\hat{c}_q\right) + \log \hat{b}_q \right] \tag{4.57}$$

$$\mathcal{C} = \frac{1}{2}\sum_{t=1}^{T}\sum_{q=1}^{m} \hat{\gamma}_q^t \langle \beta_q \rangle \langle \left(s^t - \mu_q\right)^2 \rangle_{p'(\boldsymbol{\mu})} \tag{4.58}$$

KL[$\boldsymbol{\pi}$]

The KL divergence for a Dirichlet can be found in Appendix A.3

$$KL[\boldsymbol{\pi}] = \log \frac{\Gamma(\sum_{q'} \hat{\lambda}_{q'})}{\Gamma\left(m\lambda_0\right)} - \sum_{q=1}^{m} \log \frac{\Gamma(\hat{\lambda}_{q'})}{\Gamma\left(\lambda_0\right)}$$

$$+ \quad \sum_{q=1}^{m}(\hat{\lambda}_q - \lambda_0)\left[\Psi\left(\hat{\lambda}_q\right) - \Psi\left(\sum_{q'}\hat{\lambda}_{q'}\right)\right] \tag{4.59}$$

Substituting (4.43) into $(\hat{\lambda}_q - \lambda_0)$ in (4.59) gives

$$KL[\boldsymbol{\pi}] = \mathcal{A} + \left[\log\frac{\Gamma(\sum_{q'}\hat{\lambda}_{q'})}{\Gamma(m\lambda_0)} - \sum_{q=1}^{m}\log\frac{\Gamma(\hat{\lambda}_q)}{\Gamma(\lambda_0)}\right] \tag{4.60}$$

KL[$\boldsymbol{\beta}$]

The KL divergence for a Gamma can be found in Appendix A.2

$$\begin{aligned} KL[\boldsymbol{\beta}] &= \sum_{q=1}^{m}\hat{c}_q\left(\frac{\hat{b}_q}{b_0} - 1\right) + (\hat{c}_q - c_0)\left[\Psi(\hat{c}_q) + \log\hat{b}_q\right] \\ &- \log\frac{\Gamma(\hat{c}_q)\hat{b}_q^{\hat{c}_q}}{\Gamma(c_0)b_0^{c_0}} \end{aligned} \tag{4.61}$$

Substituting (4.38) and (4.39) into (4.61) gives

$$KL[\boldsymbol{\beta}] = \mathcal{B} - \mathcal{C} - \sum_{q=1}^{m}\log\frac{\Gamma(\hat{c}_q)\hat{b}_q^{\hat{c}_q}}{\Gamma(c_0)b_0^{c_0}} \tag{4.62}$$

KL[$\boldsymbol{\mu}$]

The KL divergence for a Gaussian can be found in Appendix A.1

$$KL[\boldsymbol{\mu}] = \sum_{q=1}^{m}\frac{1}{2}\left[\left(\frac{\tau_0}{\hat{\tau}_q} - 1\right) - \log\frac{\tau_0}{\hat{\tau}_q} + \tau_0\left(\hat{m}_q - m_0\right)^2\right] \tag{4.63}$$

Entropy of $\boldsymbol{q}$

The entropy of $\boldsymbol{q}$ is simply given by

$$\mathcal{H}\left[p'(\boldsymbol{q})\right] = -\sum_{t=1}^{T}\sum_{q=1}^{m}\hat{\gamma}_q^t\log\hat{\gamma}_q^t \tag{4.64}$$

Energy

Substituting (4.55), (4.60), (4.62), (4.63) and (4.64) into (4.52) gives a much simplified expression for the negative free energy

$$F = \sum_{q=1}^{m}\log\frac{\Gamma(\hat{\lambda}_q)}{\Gamma(\lambda_0)} - \log\frac{\Gamma(\sum_{q'}\hat{\lambda}_{q'})}{\Gamma(m\lambda_0)}$$

$$
\begin{aligned}
&+ \quad \sum_{q=1}^{m} \log \frac{\Gamma(\hat{c}_q)\hat{b}_q^{\hat{c}_q}}{\Gamma(c_0)b_0^{c_0}} \\
&- \quad \frac{1}{2}\sum_{q=1}^{m}\left[\left(\frac{\tau_0}{\hat{\tau}_q} - 1\right) - \log\frac{\tau_0}{\hat{\tau}_q} + \tau_0\left(\hat{m}_q - m_0\right)^2\right] \\
&- \quad \sum_{t=1}^{T}\sum_{q=1}^{m}\hat{\gamma}_q^t \log \hat{\gamma}_q^t - \frac{T}{2}\log 2\pi
\end{aligned} \tag{4.65}
$$

The negative free energy F simplifies to a set of normalising constants, an entropy term and a dataset size term. The entropy term of $\hat{\gamma}_q^t$ measures how well the model fits the data, as can be seen from the definition of γ_q^t in (4.47). The last term relates to how much data is seen as the more data there is, the less sure a model can be that it explains *all* the data. This can be ignored if comparing models trained on the same amount of data. The NFE consists of coupled variables so although non-decreasing in principle, in practice the optimisation is cyclic and moves through the parameter space one dimension at a time. Consequently, the NFE may decrease in short periods where there is rapid change in the model parameters. This tends to be for the first 5-10 iterations, with subsequent monotonic behaviour.

Equation (4.65) is a very pleasing simplification. Essentially, the negative free energy reduces to a difference between the (log) normalising constants of the prior and posterior densities and a data misfit. This is akin to MacKay's 'Occam Factor' in [73]. The penalty for over-complex and/or overfitted models is implicit in this formula. The normalising constants measure the volume covered by a particular (unnormalised) density. Complex models will have prior densities over many variables and thus a large prior weight space. This has a penalising effect on F. Similarly, over-tuned models will have very narrow posteriors, also penalising F. Over-simple models will be penalised by the data-fit measure implicit in the entropy term. Only a model that does not overfit and is just complex enough to explain the data avoids these penalties, indicated by a maximum in F across models.

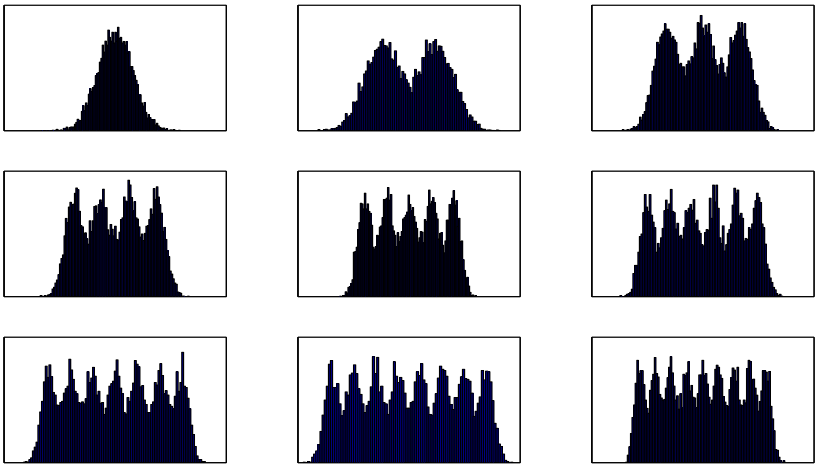


Figure 4.4: Test Distributions.

4.4.5 Results

The variational Bayesian algorithm for a Mixture of Gaussians (vbMoG) was first tested on 9 datasets. These 9 datasets were generated from 9 known MoG models, comprising 1-9 Gaussian components. Their data distributions are shown in Figure 4.4. A range of MoG models with 1-10 components were trained on each of the 9 datasets. The models were initialised using K-means clustering [1] and trained via the vbMoG algorithm on 10000, 1000 and 500 data points until the negative free energy changed by less than 0.1% (typically less than 20 iterations). The priors were chosen to be broad and weak to let the data 'do the talking' - $b_0 = 1000$, $c_0 = 0.001$, $m_0 = 0$, $\tau_0 = 1000$, $\lambda_0 = 5$. The final energy score was used to select the most likely MoG that best represented the data. The variational Bayes method was compared with Expectation-Maximisation penalised using the Bayesian Information Criterion.

The negative free energy plots over the 10 MoG models for each dataset is shown in Figure 4.5. Each plot shows three curves corresponding to training on 10000, 1000 and 500 points. The NFE has been normalised such that the maximum is unity for each curve. In all but one plot, the correct model is picked out by the energy maximum. The exception is the bottom right-hand

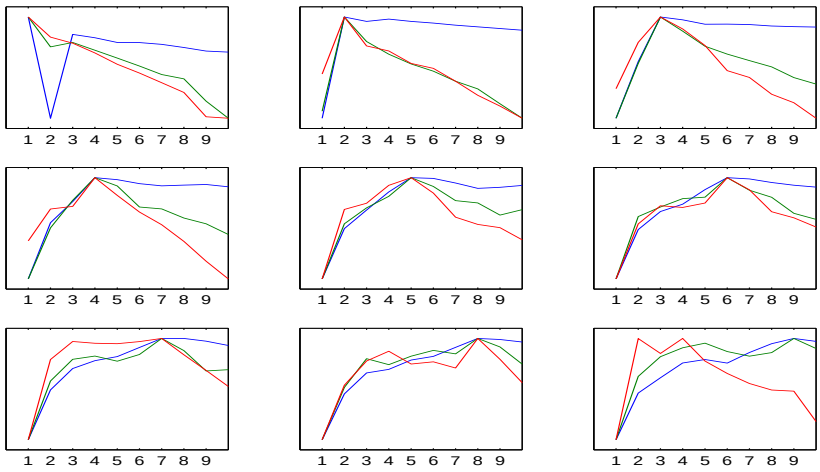


Figure 4.5: Negative free energy plots. Blue - 10000, Green - 1000, Red - 500.

plot corresponding to data generated from a 9 component MoG. The blue (10000 points) and green (1000) curves select the correct model order, but the red (500) curve favours fewer components. This is because 500 points is not enough data to resolve all 9 peaks in the distribution. The most favoured MoG models for 1000 data points are plotted in Figure 4.6.

As a comparison, the BIC plots for MoGs learnt by penalised EM are shown in Figure 4.7. The BIC curves of MoGs trained on 10000 points are mostly identical to the equivalent free energy curves for vbMoG. The curves for 1000 training points pick out the correct model order for 1-6 components, but systematically choose lower model orders for 7-9 components. Similarly for 500 training points, the penalised EM regime gets into trouble at model order 5. It can be shown [109] that BIC is a special case of the variational Bayes framework in the infinite data limit. Noting the form of (3.30), one would expect BIC to over-penalise for smaller datasets particularly as the number of parameters increases. The free energy, however, suffers from no such pathologies and only breaks down when there is insufficient data to calculate the sufficient statistics of the modelling distributions.

To simulate a more realistic situation, the vbMoG algorithm was used to train a model on the image shown in Figure 4.8(a). The image is 127 $\times$ 127

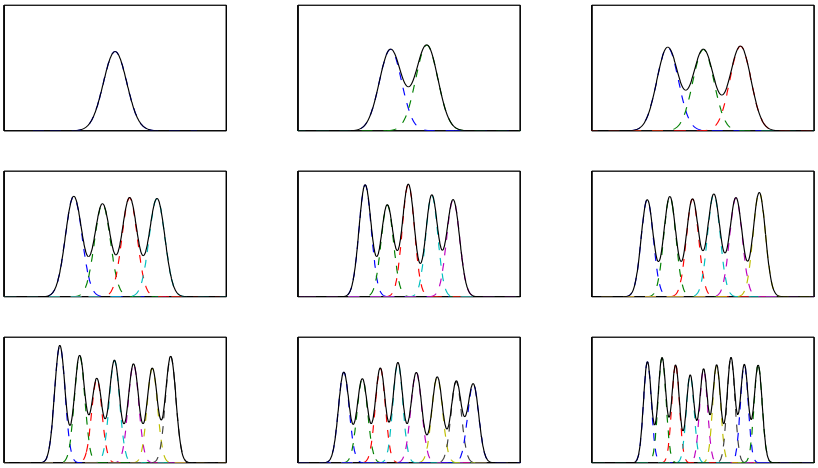


Figure 4.6: Models.

pixels and has the complex distribution plotted in Figure 4.8(b). A range of MoGs with 1-20 components were initialised using K-Means and trained by vbMoG on 1000 samples randomly drawn from the 16129-long image vector. Training continued until the energy changed by less than 0.01% (typically 30-100 iterations). The negative free energy for the 20 models is plotted in Figure 4.8(c). A MoG with 5 components is considered the best-fit model, shown in Figure 4.8(d). Although there is no sense of what is correct or incorrect in the same way as there is for the synthetically generated data, the important point to note is that Bayesian model comparison picks out the most *appropriate* model for the data *from* the data. It is this powerful concept that will be used in the following Chapter to infer the most appropriate number of sources that best describes given observations.

4.4.6 The effect of priors

As one would expect, the choice of priors has an effect on the computed posteriors and on model selection. Each of the update equations in section 4.4.3 is the sum of a prior term and a data term. If the priors are weak - as in the example above - then the data term dominates. If the priors are strong, however, or there is little data, then the prior contribution dominates and acts as a regulariser.

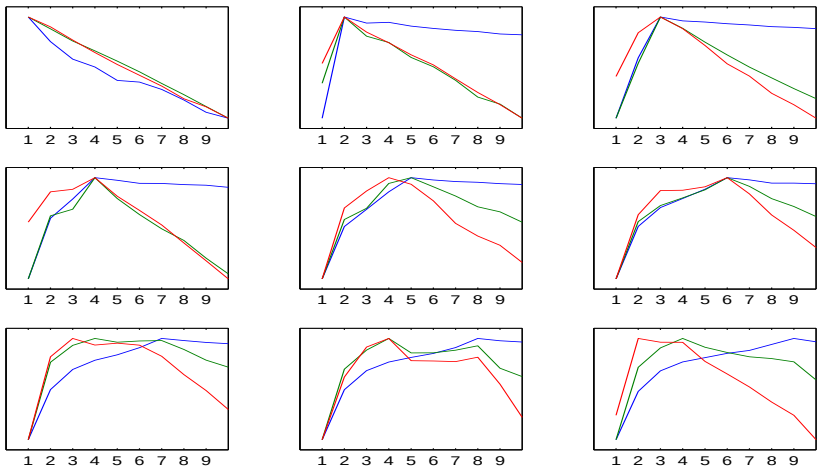
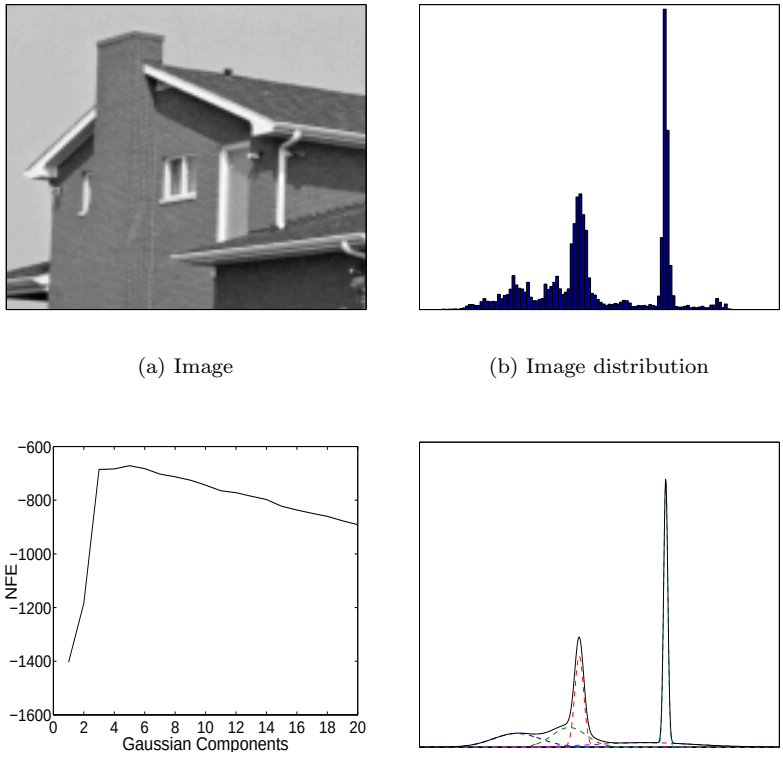


Figure 4.7: Bayesian Information Criterion plots. Blue - 10000, Green - 1000, Red - 500.

This obviously has an effect on the learnt model - set the priors too strong and the model will not adapt. Figure (4.9) illustrates the effect different priors can have. As explained previously, the prior over the MoG component means $\boldsymbol{\mu}$ is a Gaussian with mean m_0 and precision τ_0. In the example presented above, m_0 was set at 0 indicating that if there is no supportive data, set all MoG components to have a mean of 0. The strength of this prior is determined by τ_0. This was set at 100 giving a very broad distribution over $\boldsymbol{\mu}$. Therefore, $\boldsymbol{\mu}$ is given alot of room to manoeuvre. Figure 4.9(a) shows how the component means each go their own way. If τ_0 is set to 10^{-3}, however, 2 of the 5 components are effectively suppressed. Figure 4.9(b) shows how 2 of the means are forced to $\hat{m}_q \approx m_0$ as there is not enough data to support them. If τ_0 is turned up even further, to $\tau_0 = 10^{-5}$, then all the components get suppressed in Figure 4.9(c). Figures 4.9(e) and 4.9(f) plot the resultant models.

Strengthening priors also has an effect on model selection, as shown by Figure 4.9(d). The values of prior density parameters are a part of the model assumptions $\mathcal{M}$. While the loosely confined model picks 5 components as the most likely number of Gaussians, tighter constraints encourage simpler models. Stricter priors require more support from the data to overcome them. Having



(a) Image

(b) Image distribution

(c) Model selection

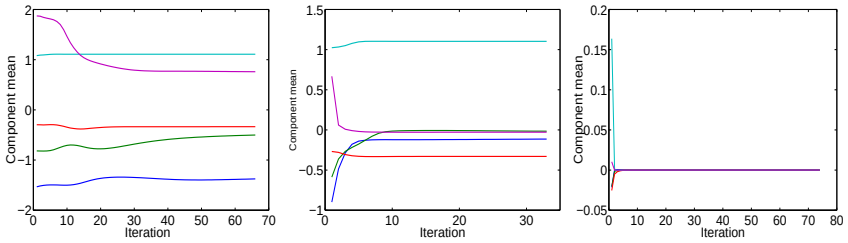
(d) MoG model

Figure 4.8: Image data.

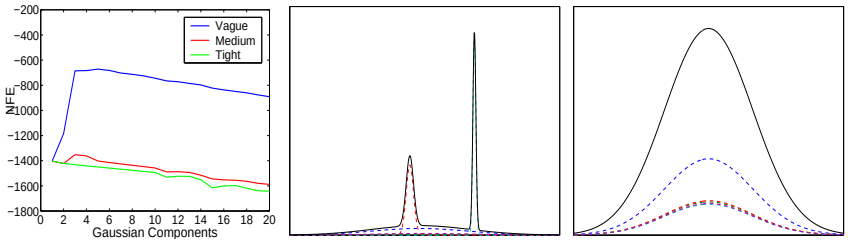
effectively killed 2 of the components, the $\tau = 10^{-3}$ negative free energy chooses a 3 component model as the most likely. Similarly, the demands of $\tau = 10^{-5}$ lead to a single component preference. The ability of priors to effectively kill unsupported components implies priors can be used for automatically determining model structure, provided they are chosen correctly. This idea will be re-examined in the next Chapter.

The Phantom Menace

On the face of it, this sensitivity of models to the choice of prior strength seems problematic. If the model and model evidence vary depending on the priors set, how can one possibly find the one 'correct' model? The simple answer is, one cannot: there is no such thing. At the risk of becoming philosophical, a



(a) Weak Prior (b) Medium Prior (c) Strong Prior



(d) Energy (e) MoG model - medium (f) MoG model - strong

Figure 4.9: Effect of Priors.

model is a model and nothing more. It is not real and is not what generated the observed data. A model just codifies postulates and assumptions about what is observed and what caused it. The choice of priors are a just another part of the gamut of all model assumptions. If the priors are poorly chosen, they will reveal themselves in the subsequent model evidence, as is clearly illustrated in Figure 4.9(d). The very fact that a change in priors reveals itself in a change in the evidence is just another arrow in Bayes' quiver.

Models are compared via (3.22). The assumptions are split into those which are questioned and which are not. It is clear that this can continue indefinately as the number of assumptions, postulates, priors - and therefore models - are infinite. The best one can ever do is compare a small-subset of plausable models. Bayes' theorem is not a panacea to the world's problems - it just helps with some ambiguous decisions. It is advisable to keep that in mind.

4.5 Summary

A central problem in ICA is working out the most appropriate number of sources to best model some observations. This is equivalent to finding the latent dimensionality of the data manifold. A probablity distribution over candidate dimensionalities can be obtained using Bayesian inference and subsequent model comparison via (3.22). If all candidates are given equal prior weight, then this is equivalent to evaluating the evidence of each model using (3.12). The evidence is of fundamental importance to probabilistic models, and measures the support for the data given model hypotheses. MacKay elegantly shows in [73] that the evidence is a measure of the effective degrees of freedom in a model. It implicitly measures the number of parameters supported by the data, and has a peak at the optimum number [73]. Consequently, it is an excellent indication of the inherent structure needed in a model.

Therefore, in principle, a method exists for infering the latent dimensionality of a data distribution. In practice, however, and as discussed earlier, the evidence is a difficult quantity to measure. The space over which the integration or summation in (3.12) is carried out is exponential in the number of weights - this is the oft talked about 'curse of dimensionality' [1]. Using the variational Bayes approximation, though, allows the evidence to be approximated in polynomial time. This framework allows efficient Bayesian inference and learning of a model's weights. Once the weights' posterior distributions have been calculated, they can be substituted into (4.3) to yield a strict lower bound on the evidence called the negative free energy. The use of the free-energy to infer latent structure has been shown in section 4.4 and its robustness compared with the widely-used Bayesian Information Criterion/Maximum Description Length methods of model comparison. It has been shown to be more robust, particularly as the number of observations decrease.

With the variational Bayes approximation, there exists a framework within which inference and learning can be performed in a closed way, and which

allows comparisons of structure to be made using information gleaned from data. Chapter 5 applies these concepts to Independent Component Analysis to yield a new and powerful analytic tool.

Chapter 5

Variational Bayesian Independent Component Analysis

In this Chapter, the variational Bayesian framework is applied to Independent Component Analysis. The posterior distributions over both the latent variables and parameters of the model are inferred. Current research is extended by using a wide variety of priors over model parameters, including noise, and inferring the source distribution as part of the learning procedure. The source distribution is unknown *a priori*, so a powerful and flexible source model is used to model a wide variety of potential distributions.

The Chapter starts with a brief summary of the current state of the art in Independent Component Analysis. This is followed by a derivation of the variational Bayesian Independent Component Analysis (vbICA) model. Two algorithms are derived based on two different factorisations of the variational posterior. The model's ability to infer complex source distributions and to decompose noisy data is explored and contrasted with traditional ICA formulations. Bayesian model comparison is used to interrogate model structure and thus infer the most likely dimensionality of the data manifold. The two vbICA algorithms are compared in section 5.4.3 and their respective merits and demerits highlighted. In section 5.5, the ability of Bayesian methods to code prior constraints is used to extend the model by incorporating a method for automat-

ically determining the latent dimensionality of the data using a method known as *Automatic Relevance Determination* (ARD). Additional prior constraints are also explored for analysing non-negative data. The Chapter concludes with a demonstration on real data and a discussion of potential applications and problems.

5.1 ICA - The State of the Art

It is useful to quickly summarise the current state of play in ICA research. ICA has traditionally been performed in the noiseless limit [45, 65], with noise often being dealt with as an extra source. Recently, however, Attias [46] extended ICA and incorporated full covariance noise into the ICA framework. The model, dubbed by Attias as Independent Factor Analysis (IFA), was subsequently learnt through a maximum likelihood EM algorithm.

Lappalainen introduced a variational Bayesian formalism for ICA in [76], where the posterior over the hidden variables and parameters was approximated. The priors over the model parameters were all Gaussians and the approximating distribution to the posterior was itself taken to be a Gaussian. This simplifies the learning process, but is limited if the posterior is not well approximated by a Gaussian. A similar variational formalism was used in [77], but where a richer variety of functional forms for the priors was used. Unlike Lappalainen, no functional form for the approximating distribution to the posterior was assumed - the forms for the posteriors expressed themselves in the optimisation procedure (see section 4.2). Crucially, however, the source model used in [77] was kept fixed and only the parameters of the observation model (mixing matrix and isotropic noise model) were learnt. As discussed in section 2.7, if the source model is not well matched to the true source distributions an incorrect and ill-fitting model is learnt and the recovered sources are sub-optimal. Miskin and Mackay [78] relaxed this constraint by utilising an adaptable mixture of Gaussians source model, albeit limited to unimodal form.

This Chapter takes a similar approach to [77], but with a multi-modal source

model that is learnt from the data. A fully adaptable factorial Mixture of Gaussians will serve as the source model and the noise covariance will be extended to a non-isotropic form. A variational Bayesian approach to a single 1-dimensional Mixture of Gaussians was explored in [98]. In this Chapter, the work of [77] is combined with [98] to produce a full adaptable variational Bayesian learning formalism for ICA.

5.2 The Proposed Model

As discussed in Chapter 2, a generative model is used. The observed variables $\boldsymbol{x}$, of dimension M, are modelled as a linear combination of statistically independent latent variables $\boldsymbol{s}$, of dimension L, with added Gaussian noise

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{n} \tag{5.1}$$

where $\boldsymbol{A}$ is an $M \times L$ mixing matrix and $\boldsymbol{n}$ is M-dimensional additive noise. In signal processing nomenclature, M is the number of (observed) sensors and L is the number of (hidden) sources.

The noise is assumed to be Gaussian, with zero mean and diagonal precision matrix $\boldsymbol{\Lambda}$

$$p(\boldsymbol{n}|\boldsymbol{\Lambda}, \mathcal{M}) = \mathcal{N}(\boldsymbol{n}; \boldsymbol{0}, \boldsymbol{\Lambda}) \tag{5.2}$$

Since the sources $\boldsymbol{s} = \{s_1, \ldots, s_L\}$ are mutually independent, the distribution over $\boldsymbol{s}$ for data point t can be written as

$$p(\boldsymbol{s}^t|\boldsymbol{\theta}, \mathcal{M}) = \prod_{i=1}^{L} p(\boldsymbol{s}_i^t|\theta_i, \mathcal{M}) \tag{5.3}$$

where $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_L\}$ are the parameters governing the source distribution. The probability of observing data vector $\boldsymbol{x}^t$ given the generative model is then

$$p(\boldsymbol{x}^t|\boldsymbol{s}^t, \boldsymbol{A}, \boldsymbol{\Lambda}, \mathcal{M}) = \left|\det(\frac{1}{2\pi}\boldsymbol{\Lambda})\right|^{\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}^t - \boldsymbol{A}\boldsymbol{s}^t)^T\boldsymbol{\Lambda}(\boldsymbol{x}^t - \boldsymbol{A}\boldsymbol{s}^t)\right] \tag{5.4}$$

where ^{T} indicates transpose.

ICA attempts to uncover the hidden source vectors that give rise to a set of observed sensor vectors. In principle, this is achieved by calculating the

posterior over the latent variables (sources) given the observed variables (sensor signals) and the model

$$p(\boldsymbol{s}^n|\boldsymbol{x}^n, \mathcal{M}) = \frac{p(\boldsymbol{x}^n|\boldsymbol{s}^n, \mathcal{M})p(\boldsymbol{s}^n|\mathcal{M})}{p(\boldsymbol{x}^n|\mathcal{M})} \tag{5.5}$$

where $p(\boldsymbol{s}^n|\mathcal{M})$ is the source model and $p(\boldsymbol{x}^n|\mathcal{M})$ is the evidence for model $\mathcal{M}$.

5.2.1 Source Model

The choice of a flexible and mathematically attractive source model is crucial if a wide variety of source distributions are to be (tractably) modelled. In particular, the source model should be capable of encompassing complex, multi-modal distributions if vbICA is to move beyond traditional ICA.

One such model is a factorised mixture of Gaussians (MoG) with L factors (i.e. sources) and m_i Gaussian components per source

$$\begin{aligned} p(\boldsymbol{s}^t|\boldsymbol{\theta}) &= \prod_{i=1}^{L}\sum_{q_i=1}^{m_i} p(q_i^t = q_i|\boldsymbol{\pi}_i)p(s_i^t|q_i^t, \mu_{i,q_i}, \beta_{i,q_i}) \\ &= \prod_{i=1}^{L}\sum_{q_i=1}^{m_i} \pi_{i,q_i}\mathcal{N}(s_i^t; \mu_{i,q_i}, \beta_{i,q_i}) \end{aligned} \tag{5.6}$$

where the explicit dependence on the model $\mathcal{M}$ has been dropped for brevity. The variable q_i is an indicator variable signifying which Gaussian component of the i^{th} source is chosen for generating s_i^t and takes on values of $\{q_i = 1, q_i = 2, \ldots, q_i = m_i\}$. π_{i,q_i} are the mixing proportions and are equivalent to $p(q_i^t = q_i|\boldsymbol{\pi}_i)$, the prior probability of choosing component q_i of the i^{th} source. The mean and precision of component q_i in source i is μ_{i,q_i} and β_{i,q_i} respectively. The vectors of component parameters can be written as $\boldsymbol{\pi}_i = [\pi_{i,1}, \pi_{i,2}, \ldots, \pi_{i,m_i}]$, $\boldsymbol{\mu}_i = [\mu_{i,1}, \mu_{i,2}, \ldots, \mu_{i,m_i}]$ and $\boldsymbol{\beta}_i = [\beta_{i,1}, \beta_{i,2}, \ldots, \beta_{i,m_i}]$. The parameters of source i are $\boldsymbol{\theta}_i = \{\boldsymbol{\pi}_i, \boldsymbol{\mu}_i, \boldsymbol{\beta}_i\}$ and the complete parameter set of the source model is $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_L\}$. The complete collection of possible source states is denoted $\boldsymbol{q} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_{\boldsymbol{m}}\}$ and runs over all $\boldsymbol{m} = \prod_i m_i$ possible combinations of source states. The probability of state $\boldsymbol{q}^t$ being chosen and generating source vector $\boldsymbol{s}^t$ is

$$p(\boldsymbol{s}^t, \boldsymbol{q}^t|\boldsymbol{\theta}) = \prod_{i=1}^{L} p(q_i^t = q_i|\boldsymbol{\pi}_i)p(s_i^t|q_i^t, \mu_{i,q_i}, \beta_{i,q_i})$$

$$= \quad p(\boldsymbol{q}^t|\boldsymbol{\pi})p(\boldsymbol{s}^t|\boldsymbol{q}^t,\boldsymbol{\theta}) \tag{5.7}$$

where $\boldsymbol{\pi} = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_L\}$. Note that the product of L 1-dimensional MoGs in (5.6) is equivalent to a single MoG in L-dimensional space with $\boldsymbol{m}$ states.

The likelihood of the IID data $\boldsymbol{X} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^T\}$ given the model parameters can now be written as

$$p(\boldsymbol{X}|\boldsymbol{\Theta}) = \prod_{t=1}^{T} \sum_{\boldsymbol{q}=1}^{\boldsymbol{m}} \int p(\boldsymbol{x}^t|\boldsymbol{s}^t, \boldsymbol{A}, \boldsymbol{\Lambda}) p(\boldsymbol{s}^t|\boldsymbol{q}^t, \boldsymbol{\theta}) p(\boldsymbol{q}^t|\boldsymbol{\pi}) d\boldsymbol{s} \tag{5.8}$$

where $d\boldsymbol{s} = \prod_i ds_i$.

The parameters $\boldsymbol{\Theta} = \{\boldsymbol{A}, \boldsymbol{\Lambda}, \boldsymbol{\theta}\}$ of the model could, for example, now be learnt by maximising the likelihood in (5.8) using an approach such as the EM algorithm (see [46] for a comprehensive derivation of the EM algorithm with regard to this model). To integrate out the dependency on the model parameters, $\boldsymbol{\Theta}$, the variational Bayesian methodology will be applied.

5.3 Variational Bayes for ICA

The maximum likelihood approach ICA is well documented [70, 46, 35], but has severe shortcomings (see section 3.6.1). Therefore, the ICA model developed above will be learnt using Bayesian inference via the variational Bayesian approach developed in Chapter 4. First, the prior distributions over the hidden variables parameters are stated.

5.3.1 The Priors

Because of source independence, it follows that the distribution over the MoG component indicator variables is

$$p(\boldsymbol{q}|\boldsymbol{\pi}) = \prod_{t=1}^{T} \prod_{i=1}^{L} \pi_{i,q_i} \tag{5.9}$$

where $\boldsymbol{q} = \{\boldsymbol{q}^1, \ldots, \boldsymbol{q}^T\}$. For a given set of components, the distribution over the sources is

$$p(\boldsymbol{S}|\boldsymbol{q}^t, \boldsymbol{\theta}) = \prod_{t=1}^{T} \prod_{i=1}^{L} \mathcal{N}(s_i^t; \mu_{i,q_i}, \beta_{i,q_i}) \tag{5.10}$$

Now set priors over the parameters in question. The prior over the source model parameters is

$$p(\boldsymbol{\theta}) = p(\boldsymbol{\pi})p(\boldsymbol{\mu})p(\boldsymbol{\beta}) \tag{5.11}$$

where $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_L\}$ and $\boldsymbol{\beta} = \{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_L\}$. The prior over the mixture proportions is a product of symmetric Dirichlets

$$p(\boldsymbol{\pi}) = \prod_{i=1}^{L} \mathcal{D}(\boldsymbol{\pi}_i; \lambda_{i0}) \tag{5.12}$$

The prior over the means is a product of Gaussians

$$p(\boldsymbol{\mu}) = \prod_{i=1}^{L} \prod_{q_i=1}^{m_i} \mathcal{N}(\mu_{i,q_i}; m_{i0}, \tau_{i0}) \tag{5.13}$$

The prior over the precisions is a product of Gammas

$$p(\boldsymbol{\beta}) = \prod_{i=1}^{L} \prod_{q_i=1}^{m_i} \mathcal{G}(\beta_{i,q_i}; b_{i0}, c_{i0}) \tag{5.14}$$

The prior over the sensor noise precision is a product of Gammas

$$p(\boldsymbol{\Lambda}) = \prod_{j=1}^{M} \mathcal{G}(\Lambda_j; b_{\Lambda_j}, c_{\Lambda_j}) \tag{5.15}$$

The prior over the mixing matrix is a product of Gaussians

$$p(\boldsymbol{A}) = \prod_{i=1}^{L} \prod_{j=1}^{M} \mathcal{N}(A_{ji}|0, \alpha_{ji}) \tag{5.16}$$

Figure 5.1 shows the graphical model for vbICA. Now the priors have been specified, the variational Bayes methodology can be implemented.

5.3.2 Variational Methodology

The objective function to be maximised is the negative free energy, F:

$$F = \langle \log p(\boldsymbol{X}, \boldsymbol{W}) \rangle_{p'(\boldsymbol{W})} + \mathcal{H}\left[p'(\boldsymbol{W})\right] \tag{5.17}$$

where $\boldsymbol{W} = \{\boldsymbol{A}, \boldsymbol{\Lambda}, \boldsymbol{S}, \boldsymbol{q}, \boldsymbol{\theta}\}$ is the ensemble of hidden variables and parameters. The negative variational free energy forms a strict lower bound on the evidence of the model, with the difference being the Kullback-Leibler divergence between the true and approximating posteriors. Maximising this function is equivalent

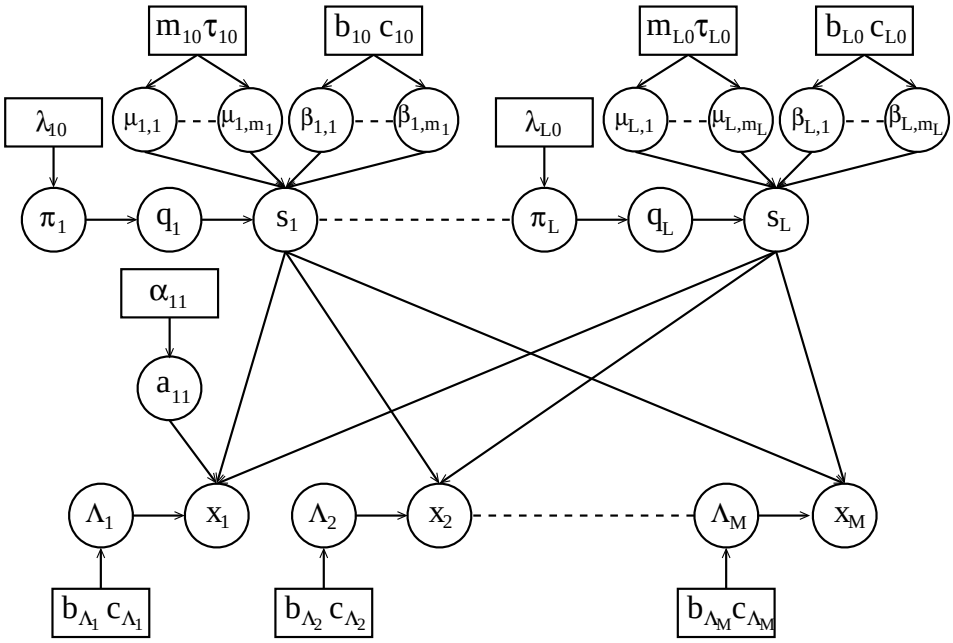


Figure 5.1: Bayesian Independent Component Analysis as a Graphical Model. Circles represent random variables and rectangles represent hyper-parameters.

to minimising the KL divergence between the true and approximate posteriors. By choosing $p'(\boldsymbol{W})$ such that it factorises over the ensemble of hidden variables, $\boldsymbol{W}$, terms in each hidden variable can be maximised individually. Two different factorisations are chosen to highlight the effect of different assumptions on the results

$$p'(\boldsymbol{W}) = p'(\boldsymbol{\Lambda})p'(\boldsymbol{A})p'(\boldsymbol{S})p'(\boldsymbol{q})p'(\boldsymbol{\theta}) \tag{5.18}$$

$$p'(\boldsymbol{W}) = p'(\boldsymbol{\Lambda})p'(\boldsymbol{A})p'(\boldsymbol{S}|\boldsymbol{q})p'(\boldsymbol{q})p'(\boldsymbol{\theta}) \tag{5.19}$$

where $p'(\boldsymbol{\theta}) = p'(\boldsymbol{\pi})p'(\boldsymbol{\mu})p'(\boldsymbol{\beta})$. The density $p'(\boldsymbol{S})$ in (5.18) will be shown to be of Gaussian form while the term $p'(\boldsymbol{S}|\boldsymbol{q})$ in (5.19) implies a mixture posterior source density for source i of the form

$$p'(s_i^t) = \sum_{q_i=1}^{m_i} p'(q_i^t = q_i)p'(s_i|q_i) \tag{5.20}$$

$$\doteq \sum_{q_i=1}^{m_i} \hat{\gamma}_{i,qi}^t \mathcal{N}(s_i^t; \hat{\mu}_{i,q_i}^t, \hat{\beta}_{i,q_i}^t) \tag{5.21}$$

where a posterior MoG density is implicit in the choice of a MoG prior over each source s. This seemingly small change will have repercussions when comparing the robustness of the two algorithms. The algorithms derived under (5.18) and (5.19) will be termed vbICA1 and vbICA2 respectively.

In a similar approach to [46], define the posteriors over the sources to factorise such that

$$p'(\boldsymbol{s}) = \prod_{i=1}^{L} p'(s_i) \tag{5.22}$$

This additional factorisation allows efficient scaling of computation with the number of hidden sources, with little loss of accuracy [112]. There are consequences for highly correlated data, however, as (5.22) is then too severe. This is discussed in more detail in section 5.7.1.

By substituting $p(\boldsymbol{X}, \boldsymbol{W})$ and either (5.18) or (5.19) into (5.17), one can obtain an expression for the negative free energy of the model. The term $p(\boldsymbol{X}, \boldsymbol{W})$ is straight forward to write down

$$p(\boldsymbol{X}, \boldsymbol{W}) = p(\boldsymbol{X}|\boldsymbol{A}, \boldsymbol{\Lambda}, \boldsymbol{S})p(\boldsymbol{S}|\boldsymbol{q}, \boldsymbol{\mu}, \boldsymbol{\beta})p(\boldsymbol{q}|\boldsymbol{\pi})p(\boldsymbol{\theta})p(\boldsymbol{A})p(\boldsymbol{\Lambda}) \tag{5.23}$$

where

$$p(\boldsymbol{X}|\boldsymbol{A}, \boldsymbol{\Lambda}, \boldsymbol{S}) = \prod_{t=1}^{T} p(\boldsymbol{x}^t|\boldsymbol{s}^t, \boldsymbol{A}, \boldsymbol{\Lambda}) \tag{5.24}$$

For vbICA1, the negative free energy is

$$\begin{aligned} F \quad = \quad & \langle \log p(\boldsymbol{X}|\boldsymbol{A}, \boldsymbol{\Lambda}, \boldsymbol{S}) \rangle_{p'(\boldsymbol{A})p'(\boldsymbol{\Lambda})p'(\boldsymbol{S})} \\ & + \langle \log p(\boldsymbol{S}|\boldsymbol{q}, \boldsymbol{\mu}, \boldsymbol{\beta}) \rangle_{p'(\boldsymbol{S})p(\boldsymbol{q})p'(\boldsymbol{\mu})p'(\boldsymbol{\beta})} + \mathcal{H}\left[p'(\boldsymbol{S})\right] \\ & + \langle \log p(\boldsymbol{q}|\boldsymbol{\pi}) \rangle_{p'(\boldsymbol{q})p'(\boldsymbol{\pi})} + \mathcal{H}\left[p'(\boldsymbol{q})\right] \\ & + \langle \log p(\boldsymbol{\pi}) \rangle_{p'(\boldsymbol{\pi})} + \mathcal{H}\left[p'(\boldsymbol{\pi})\right] \\ & + \langle \log p(\boldsymbol{\mu}) \rangle_{p'(\boldsymbol{\mu})} + \mathcal{H}\left[p'(\boldsymbol{\mu})\right] \\ & + \langle \log p(\boldsymbol{\beta}) \rangle_{p'(\boldsymbol{\beta})} + \mathcal{H}\left[p'(\boldsymbol{\beta})\right] \\ & + \langle \log p(\boldsymbol{A}) \rangle_{p'(\boldsymbol{A})} + \mathcal{H}\left[p'(\boldsymbol{A})\right] \\ & + \langle \log p(\boldsymbol{\Lambda}) \rangle_{p'(\boldsymbol{\Lambda})} + \mathcal{H}\left[p'(\boldsymbol{\Lambda})\right] \end{aligned} \tag{5.25}$$

The energy for vbICA2 is found similarly (see Appendix C).

One may now proceed by specifying functional forms of each of the approximating posteriors and using these in (5.17) as shown by [76]. As shown in section 4.2, however, there is no need to specify functional forms for the approximating posteriors as they 'fall-out' of the maximisation process.

5.3.3 The Posteriors

vbICA1

Maximising (5.25) yields the following marginal posteriors for vbICA1 (see Appendix B)

$$p'(\boldsymbol{q}) = \prod_{i=1}^{L}\prod_{t=1}^{T}\hat{\gamma}_{i,q_i}^{t} \tag{5.26}$$

$$p'(\boldsymbol{\pi}) = \prod_{i=1}^{L}\mathcal{D}(\boldsymbol{\pi}_i;\hat{\boldsymbol{\lambda}}_{i,q:m_i}) \tag{5.27}$$

$$p'(\boldsymbol{\mu}) = \prod_{i=1}^{L}\prod_{q_i=1}^{m_i}\mathcal{N}(\mu_{i,q_i};\hat{m}_{i,q_i},\hat{\tau}_{i,q_i}) \tag{5.28}$$

$$p'(\boldsymbol{\beta}) = \prod_{i=1}^{L}\prod_{q_i=1}^{m_i}\mathcal{G}(\beta_{i,q_i};\hat{b}_{i,q_i},\hat{c}_{i,q_i}) \tag{5.29}$$

$$p'(\boldsymbol{S}) = \prod_{i=1}^{L}\prod_{t=1}^{T}\mathcal{N}(s_i^t;\hat{\mu}_i^{(t)},\hat{\beta}_i^{(t)}) \tag{5.30}$$

$$p'(\boldsymbol{A}) = \prod_{i=1}^{L}\prod_{j=1}^{M}\mathcal{N}(A_{ji};\hat{m}_{A_{ji}},\hat{\alpha}_{ji}) \tag{5.31}$$

$$p'(\boldsymbol{\Lambda}) = \prod_{j=1}^{M}\mathcal{G}(\Lambda_j;\hat{b}_{\Lambda_j},\hat{c}_{\Lambda_j}) \tag{5.32}$$

The parameters of the posteriors are updated 'versions' of the parameters of the priors. The update equations are detailed below[1]:

Source model

- $p'(\boldsymbol{q})$

$$\gamma_{i,q_i}^{t} = \tilde{\pi}_{i,q_i}\tilde{p}_{i,q_i} \tag{5.33}$$

$$\hat{\gamma}_{i,q_i}^{t} = \frac{\gamma_{i,q_i}^{t}}{\sum_{q_i'}\gamma_{i,q_i'}^{t}} \tag{5.34}$$

[1]The term $\langle a\rangle$ is the expectation of a with respect to $p'(a)$.

where

$$\tilde{\pi}_{i,q_i} = \exp\left[\Psi(\hat{\lambda}_{i,q_i}) - \Psi(\sum_{q_i'}\hat{\lambda}_{i,q_i'})\right] \quad (5.35)$$

$$\tilde{p}_{i,q_i} = \tilde{\beta}_{i,qi}^{\frac{1}{2}}\exp\left[-\frac{\langle\beta_{i,q_i}\rangle}{2}\langle\left(s_i^t-\mu_{i,q_i}\right)^2\rangle\right] \quad (5.36)$$

$$\tilde{\beta}_{i,qi} = \hat{b}_{i,q_i}\exp\left[\Psi(\hat{c}_{i,q_i})\right] \quad (5.37)$$

and where (5.34) ensures that $\sum_{q_i}\hat{\gamma}_{i,q_i}^t = 1$. $\Psi(.)$ is the Digamma function.

- $p'(\boldsymbol{\pi})$

$$\hat{\lambda}_{i,q_i} = \lambda_{i0} + \sum_{t=1}^{t}\hat{\gamma}_{i,q_i}^t \quad (5.38)$$

- $p'(\boldsymbol{\mu})$

$$\hat{m}_{i,q_i} = \frac{1}{\hat{\tau}_{i,q_i}}\left(m_{i0} + \langle\beta_{i,q_i}\rangle\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^t\langle s_i^t\rangle\right) \quad (5.39)$$

$$\hat{\tau}_{i,q_i} = \tau_{i0} + \langle\beta_{i,q_i}\rangle\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^t \quad (5.40)$$

- $p'(\boldsymbol{\beta})$

$$\hat{b}_{i,q_i} = \left[\frac{1}{b_{i0}} + \frac{1}{2}\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^t\langle\left(s_i^t-\mu_{i,q_i}\right)^2\rangle\right]^{-1} \quad (5.41)$$

$$\hat{c}_{i,q_i} = c_{i0} + \frac{1}{2}\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^t \quad (5.42)$$

Observation model

- $p'(\boldsymbol{S})$

$$\hat{\mu}_i^{(t)} = \frac{1}{\hat{\beta}_i^{(t)}}\left[\bar{\mu}_i + \sum_{j=1}^{M}\langle\Lambda_j\rangle\langle A_{ji}\rangle\left(x_j^t - \langle\hat{x}_{j,k\neq i}^t\rangle\right)\right] \quad (5.43)$$

$$\hat{\beta}_i^{(t)} = \bar{\beta}_i + \sum_{j=1}^{M}\langle\Lambda_j\rangle\langle A_{ji}^2\rangle \quad (5.44)$$

where

$$\bar{\mu}_i = \sum_{q_i=1}^{m_i}\hat{\gamma}_{i,q_i}^t\langle\beta_{i,q_i}\rangle\langle\mu_{i,q_i}\rangle \quad (5.45)$$

$$\bar{\beta}_i = \sum_{q_1=1}^{m_i}\hat{\gamma}_{i,q_i}^t\langle\beta_{i,q_i}\rangle \quad (5.46)$$

$$\hat{x}^t_{j,k\neq i} = \sum_{k\neq i}^{L} A_{jk} s^t_k \tag{5.47}$$

$$\hat{x}^t_j = \sum_{i=1}^{L} A_{ji} s^t_i \tag{5.48}$$

Note the intuitive form of (5.43) and (5.44). Source i 'sees' data x^t_j at sensor j and works out what it can 'explain away' given information in the rest of the model, i.e. $\hat{x}^t_{j,k\neq i}$. The residual information is then used to update its own parameters in a bid to explain what is 'left over'. This is analogous to Pearl's message passing algorithms [93], whereby a node (s_i) updates its belief (encapsulated in $\hat{\mu}_i$ and $\hat{\beta}_i$) using messages passed to it by its parents (quantified by $\bar{\mu}_i$ and $\bar{\beta}_i$), its children (the sensors) and all other parents of its children (i.e. all other sources and quantified by $\hat{x}_{j,k\neq i}$). This use of information from a variable's Markov blanket runs through all the update equations.

- $p'(\boldsymbol{A})$

$$\hat{m}_{A_{ji}} = \frac{\langle \Lambda_j \rangle}{\hat{\alpha}_{ji}} \sum_{t=1}^{T} \langle s^t_i \rangle \left(x^t_j - \langle \hat{x}^t_{j,k\neq i} \rangle \right) \tag{5.49}$$

$$\hat{\alpha}_{ji} = \alpha_{ji} + \langle \Lambda_j \rangle \sum_{t=1}^{t} \langle s_i^{t\,2} \rangle \tag{5.50}$$

Noise model

- $p'(\boldsymbol{\Lambda})$

$$\hat{b}_{\Lambda_j} = \left[\frac{1}{b_{\Lambda_j}} + \frac{1}{2} \sum_{t=1}^{T} \langle \left(x^t_j - \hat{x}^t_j \right)^2 \rangle \right]^{-1} \tag{5.51}$$

$$\hat{c}_{\Lambda_j} = c_{\Lambda_j} + \frac{T}{2} \tag{5.52}$$

The relevant expectations are given by

$$\langle a \rangle = \text{mean}(a)$$

$$\langle a^2 \rangle = \text{mean}(a)^2 + \text{variance}(a)$$

and are detailed for the various distributions in Appendix A.

vbICA2

The posteriors under the vbICA2 factorisation have the same functional form as (5.26-5.32) except $p'(\boldsymbol{S})$ which is replaced by

$$p'(\boldsymbol{S}|\boldsymbol{q}) = \prod_{i=1}^{L}\prod_{t=1}^{T}\mathcal{N}(s_i^t; \hat{\mu}_{i,q_i}^{(t)}, \hat{\beta}_{i,q_i}^{(t)}) \tag{5.53}$$

This alters the update equations that depend on expectations of $\boldsymbol{S}$ (see Appendix C):

- $p'(\boldsymbol{S}|\boldsymbol{q})$

$$\hat{\mu}_{i,q_i}^t = \frac{1}{\hat{\beta}_{i,q_i}^t}\left[\langle\beta_{i,q_i}\rangle\langle\mu_{i,q_i}\rangle + \sum_{j=1}^{M}\langle\Lambda_j\rangle\langle A_{ji}\rangle\left(x_j^t - \langle\hat{x}_{j,k\neq i}^t\rangle\right)\right] \tag{5.54}$$

$$\hat{\beta}_{i,q_i}^t = \langle\beta_{i,q_i}\rangle + \sum_{j=1}^{M}\langle\Lambda_j\rangle\langle A_{ji}^2\rangle \tag{5.55}$$

  In practice, (5.54) has to be iterated for every i a number of times until $\hat{\mu}_{i,q_i}^t$ converges as it depends on every other $k \neq i$.

- $p'(\boldsymbol{q})$

  The update is the same as (5.33)/(5.34), but with (5.36) replaced with

$$\tilde{p}_{i,q_i} = \left(\frac{\tilde{\beta}_{i,qi}}{\hat{\beta}_{i,qi}^t}\right)^{\frac{1}{2}}\exp\left[\frac{1}{2}\left(\hat{\beta}_{i,qi}^t\hat{\mu}_{i,q_i}^{t^2} - \langle\beta_{i,qi}\rangle\langle\mu_{i,q_i}^2\rangle\right)\right] \tag{5.56}$$

- $p'(\boldsymbol{\mu})$

$$\hat{m}_{i,q_i} = \frac{1}{\hat{\tau}_{i,q_i}}\left(\tau_{i0}m_{i0} + \langle\beta_{i,q_i}\rangle\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^t\langle s_i^t|q_i^t\rangle\right) \tag{5.57}$$

$$\hat{\tau}_{i,q_i} = \tau_{i0} + \langle\beta_{i,q_i}\rangle\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^t \tag{5.58}$$

- $p'(\boldsymbol{\beta})$

$$\hat{b}_{i,q_i} = \left(\frac{1}{b_{i0}} + \frac{1}{2}\tilde{\sigma}_{i,q_i}\right)^{-1} \tag{5.59}$$

$$\hat{c}_{i,q_i} = c_{i0} + \frac{1}{2}\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^t \tag{5.60}$$

  where the average variance of component q_i in source i is defined as

$$\tilde{\sigma}_{i,q_i} = \sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^t\left(\langle s_i^{t^2}|q_i^t\rangle - 2\langle\mu_{i,q_i}\rangle\langle s_i^t|q_i^t\rangle + \langle\mu_{i,q_i}^2\rangle\right)$$

and where $\langle a|b\rangle$ is the expectation of a w.r.t. $p'(a|b)$. The posterior parameter updates for $\boldsymbol{A}$ and $\boldsymbol{\Lambda}$ remain the same, with the source expectations given by the following identities

$$\langle s_i^t \rangle = \sum_{q_i=1}^{m_i} p'(q_i^t = q_i) \langle {s_i}^t | q_i^t \rangle \tag{5.61}$$

$$\langle {s_i^t}^2 \rangle = \sum_{q_i=1}^{m_i} p'(q_i^t = q_i) \langle {s_i^t}^2 | q_i^t \rangle \tag{5.62}$$

where

$$p'(q_i^t = q_i) = \hat{\gamma}_{i,qi}^t \tag{5.63}$$

$$\langle s_i^t | q_i^t \rangle = \hat{\mu}_{i,q_i}^t \tag{5.64}$$

$$\langle {s_i^t}^2 | q_i^t \rangle = (\hat{\mu}_{i,q_i}^t)^2 + \frac{1}{\hat{\beta}_{i,q_i}^t} \tag{5.65}$$

5.3.4 Hierarchical Interpretation

The update equations for the vbICA source model parameters bear a striking resemblance to the update equations for the MoG example in section 4.4.3. The vbICA model acts as two separate networks operating in a hierarchy. During learning, the observation+noise model unmixes the observed data vectors into L 1-dimensional source signals using the current estimates for the various posteriors. These L signals are each fed to the collection of L MoG models. Each MoG source model 'sees' the relevant source expectations as data points and thus learns the distributions of these expectations. Therefore, the MoG updates are similar to those in section 4.4.3 with hidden source expectations $\langle s_i \rangle$ and $\langle s_i^2 \rangle$ taking the place of visible data.

5.3.5 Implementing vbICA

The update equations (5.33)-(5.52) are coupled and therefore must be solved iteratively. This is achieved by starting with initial guesses of the variables and cycling through the update equations using the moments calculated until convergence. The learning steps for vbICA1 may be conveniently implemented in the algorithm shown here in pseudo-code form in Table 5.1. The methodology

initialise;
WHILE ($\Delta F_{(ica)} <$ tolerance)

  WHILE ($\Delta F_{(obs)} <$ tolerance)
    compute (5.44) using current estimate of $\langle \boldsymbol{A} \rangle$ and $\langle \boldsymbol{\Lambda} \rangle$;
    compute source expectations by cycling through (5.43) for all i until convergence
    compute (5.50) using current estimate of $\langle \boldsymbol{S} \rangle$ and $\langle \boldsymbol{\Lambda} \rangle$;
    compute mixing matrix pdf by cycling through (5.43) for all i until convergence
    calculate $F^{new}_{(obs)}$;
    calculate $\Delta F_{(obs)} \doteq |F^{new}_{(obs)} - F^{old}_{(obs)}|$;
  END WHILE;

  FOR i=1:L
    WHILE ($\Delta F_{(MoG_i)} <$ tolerance)
      update MoG model i by computing (5.33)-(5.42);
      calculate $F^{new}_{(MoG_i)}$;
      calculate $\Delta F_{(MoG_i)} \doteq |F^{new}_{(MoG_i)} - F^{old}_{(MoG_i)}|$;
    END WHILE;
  END FOR;

  update ICA noise model using (5.51) and (5.52);
  calculate $F^{new}_{(ica)}$;
  calculate $\Delta F_{(ica)} \doteq |F^{new}_{(ica)} - F^{old}_{(ica)}|$;

END WHILE;

Table 5.1: Pseudo-code for vbICA updates.

for vbICA2 is similar, but with the appropriate changes in the update equations and with the vbICA2 versions of (5.33)/(5.34) computed only *once* during the MoG updates. The factorisation of $p'(\boldsymbol{S}|\boldsymbol{q})p'(\boldsymbol{q})$ strongly couple the source reconstructions and hidden source states, so this ensures stability.

Once trained, the model can be used to reconstruct hidden sources (to within a scaling and permutation) given a dataset by cycling through the updates for $p'(\boldsymbol{q})$ and $p'(\boldsymbol{S})/p'(\boldsymbol{S}|\boldsymbol{q})$ using the (now-fixed) model parameter posteriors until convergence.

Priors and Initialisation

As with the MoG model in Chapter 4, the choice of priors has an effect on the final model. Unless specific priors are required, the prior parameter values are best chosen from the ranges indicated below.

For the ICA model presented in this Chapter, a wide variety of priors ($10^{-6} \leq b \leq 10^{6}$, $10^{-6} \leq c \leq 10^{6}$ for all Gamma distributions, scale parameter = 5-5000 for all Dirichlets and precision = $10^{-6} - 10^{6}$ for Gaussians) were examined. For Gamma distributions, the variance is given by b^2c and the mean by bc. Large values of c tend to be highly peaked around the mode, tending towards Gaussianity. Values of $c < 1$ have no mode and are more like exponentials. Consequently, they encourage small values for the random variate while curtailing large values beyond the mean. $1 \leq b \leq 1000$ with $bc \approx 1$ were found to be a useful range for Gamma distributions. Gamma distributions govern scale parameters, in this case precisions. Values of $bc > 10$ have a narrowing effect on the posteriors, making them over-confident. Values of $bc < 10^{-1}$ with $c > 1$ have a smoothing effect on the posteriors, losing detail in the process. Similarly, precisions of the order of $10^{0} - 10^{-3}$ were found to be the most applicable. The value of Dirichlet scale parameters act as 'pseudo-counts', so high values are very constraining. Appropriate values depend on the number of data vectors. If T is the number of data vectors, then values between $0.01T$ and $0.1T$ stop components dying while allowing the data to set the posterior values accurately. If the algorithm was recast as an online scheme, however, these values would have to be reexamined as the results would necessarily become more sensitive to the choice of priors.

The choice of initialisation is also important. As with any ICA formalism, the vbICA algorithms find a local maximum in the parameter space and, as such, results depend on initialisation. There are many ways to initialise, but the most common for ICA are to intialise randomly or start at the PCA solution. For a data matrix that is $M \times T$, this is tantamount to performing SVD on the

data matrix

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{5.66}$$

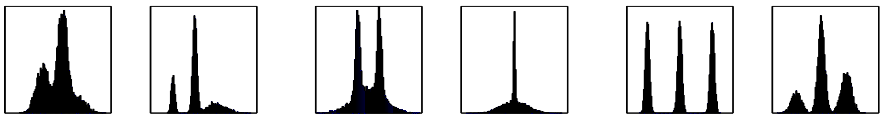
where $\boldsymbol{U}$ and $\boldsymbol{V}$ are othonormal matrices and $\boldsymbol{\Sigma}$ is the diagonal matrix of singular values. If SVD is used then the first L columns of $\boldsymbol{V}$ are used to initialise L MoGs using k-means, while residual singular values are used to intialise the noise covariance. If $d = L + 1 : M$, then the noise precision is initilised by

$$\boldsymbol{\Lambda}_{\text{init}} = \frac{T}{\sum_d \Sigma_{dd}}\boldsymbol{I} \tag{5.67}$$

If mixing is square (i.e. $L = M$), then noise is initialised at 5 percent of the mean variance across the data dimensions. The first L columns of $\boldsymbol{U}$ are used to initialise the mixing matrix.

Both PCA and random intialisation have their own merits and demerits. Initialising using PCA yields a repeatable and quicker result, but vbICA can have problems with resolving directions in data densities exhibiting high correlation, often staying at - or close to - the PCA result. Random intialisation can overcome this to a certain extent, but different intialisations may yield different results if the free-energy has multiple stationary points. If this is the case, the model may have to be intialised a number of times to find the final result with the lowest free-energy (highest value for F). The correlation problem can sometimes be overcome by decorrelating the data first. This is discussed further in section 5.7.1.

One must also choose the number of components in each source MoG. This can be determined by monitoring the contibution to F_{ICA} by each source (F_{MoG_i}) as a function of components. This can sometimes be time consuming. The number of components is not a fundamental quantity of interest so may be fixed to a sufficient number. For most datasets, 3-5 components have been found to be enough, although more may be needed for complex image data. Although by no means necessary, centring and normalising the data to unit variance avoids potential numerical problems during computation. This is carried out for all experiments presented here and in subsequent Chapters.



(a) Group 1 (b) Group 2 (c) Group 3

Figure 5.2: Original source distributions.

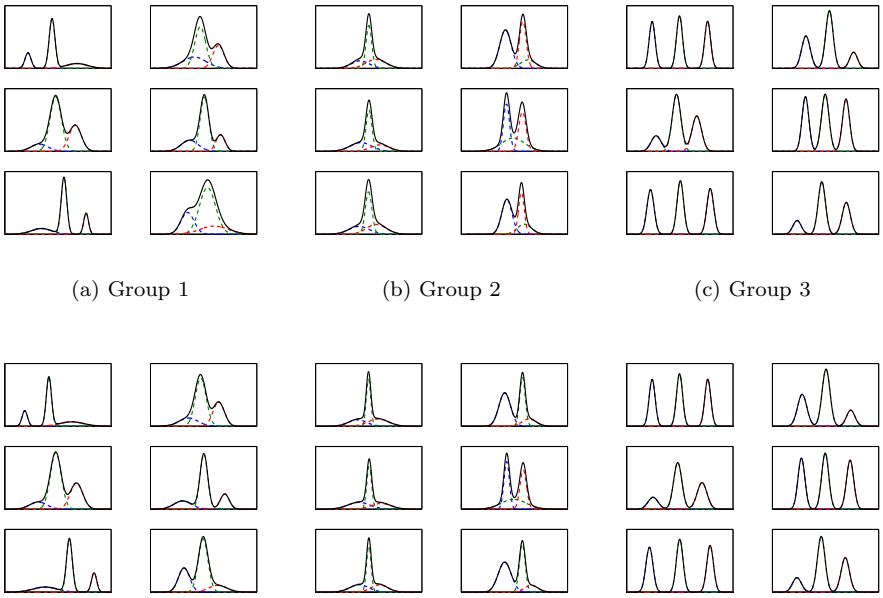
5.4 Results

This initial results section is presented in three parts. The two algorithms will be first demonstrated on some toy data to illustrate their modelling abilities. They will then be compared with current ICA algorithms in section 5.4.2. Both algorithms will be shown to be superior at separating image mixtures and producing interpretable results. Following this, section 5.4.3 will explore the differences between vbICA1 and vbICA2, showing the latter to be more accurate and robust under noise.

5.4.1 Toy Data

Both vbICA1 and vbICA2 were tested on nine different datasets, grouped into three sets of three. Each group consisted of three different random projections of the same two-source distribution. Three different source distributions were used, giving a total of nine datasets. The distributions of each of the sources is shown in Figure 5.2. 1000 points were drawn from each group of sources, then centred and normalised to unit-variance to allow ease of comparison later on. Each group of points was projected 3 times by mixing matrices randomly drawn from a Gaussian generator. The projections were centred and normalised to unit-variance before 5 % Gaussian noise was added.

The 9 datasets were used to train 9 vbICA models with vbICA1 and 9 models with vbICA2. Each model was intialised using SVD, with 3 Gaussian components per source. Training continued until the negative free energy (NFE), F, converged to within 0.01 %. Training took typically took 50-150 iterations.



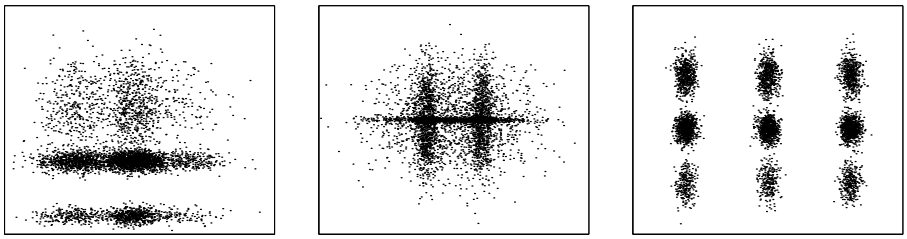
(a) Group 1 (b) Group 2 (c) Group 3

Figure 5.3: Learnt source distributions. Top: vbICA1, Bottom: vbICA2.

Source models

Figure 5.3 shows the source models learnt by vbICA. All are very close to the original source distributions in Figure 5.2. Figure 5.4 show how 2-D models typically compare with the original distributions. In particular, note how the multi-modal nature has been captured, something not possible using traditional ICA. The vbICA models also compared well against the (non-Bayesian) variational version of Independent Factor Analysis [46] (IFA). IFA utilises MoGs as source models, so is capable of capturing multi-modal sources. In simulations, however, IFA required 2-3 times the data and 250-300 iterations to model groups 1 and 3 as accurately as vbICA, whilst it struggled with group 2 even with 3 times the data and 300 iterations. With less than 1000 data points, there is the added problem of Gaussian components in the MoGs sometimes collapsing onto single source points as this increases the likelihood, something that vbICA avoids due to the natural way Bayesian priors regularise the learning process.

Although the distributions are modelled well, it is clear some models are



(a) Group 1 (b) Group 2 (c) Group 3

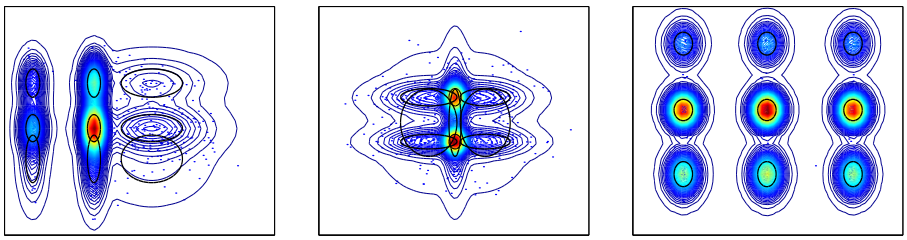
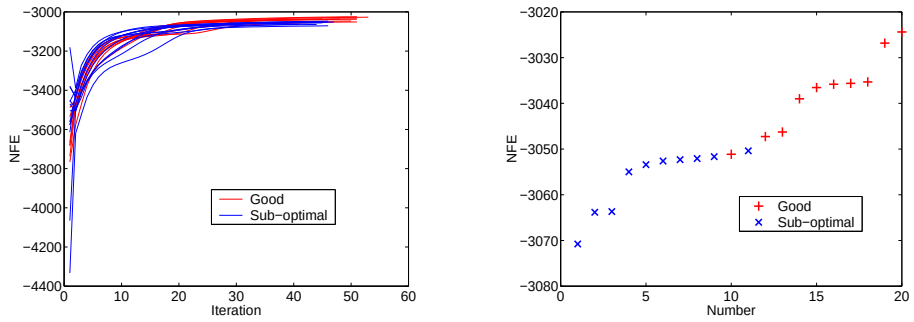


Figure 5.4: 2-D source distributions. Top: Original, Bottom: vbICA models. Contours plot $0.001 < p(\boldsymbol{s}) < 1$ in 250 intervals. Blue-red $\equiv$ low-high $p(\boldsymbol{s})$. Note how the first two source models are rotated 90 degress w.r.t. the original distributions. This is the unavoidable source permutation ambiguity of ICA.

better than others. In particular, vbICA2 has captured the sources of group 1 better across the projections than vbICA1. Both, however, have difficulty with group 2 as these sources are particularly complex and two of the projections were highly correlated. Both vbICA models fared better if they were initialised at random a number of times. Those with higher NFEs were found to correctly model these complicated distributions. Figure 5.5(a) shows the increase in F as 20 different models are learnt from random intialisations. Of the 20, 9 converged to accurate models, while the rest were sub-optimal. Although all free-energies converge to similar values, if these values are plotted in ascending order, as shown in Figure 5.5(b), all but one of the 'good' models have higher NFEs than the sub-optimal ones. Therefore, if one suspects the data are highly correlated, random intialisation followed by selection via F will increase the likelihood of obtaining an accurate model over initialisation by SVD.



(a) NFE for 20 runs

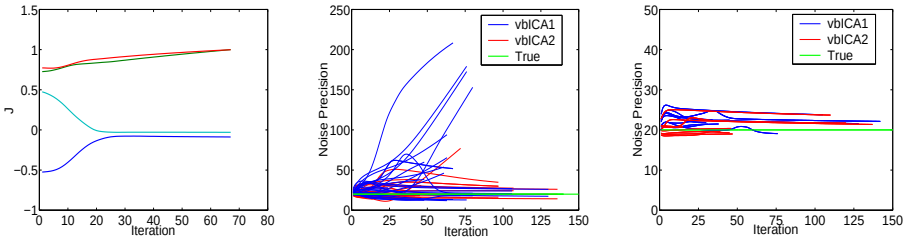
(b) NFE in ascending order

Figure 5.5: NFE for 'good' and 'sub-optimal' models.

The optimisation landscape

Figure 5.5 also gives an indication of the shape of the energy landscape. The ideal optimisation landscape is one in which there is only one maximum (or, equivalently, minimum). In real problems, this is rarely - if ever - the case. ICA's permutation ambiguity equates to symmetries in the landscape such that for L sources there are $L!$ equally high maxima. Furthermore, there are parallel straight ridges that pass through these peaks due to the scaling ambiguity. Therefore, the ideal optimisation is one in which a model is intialised somewhere on this landscape, 'moves' up the nearest ridge during learning and stops when it gets to the top.

With noise and multi-modal sources, however, extra ridges, plateaux and valleys are introduced due to the extra degrees of freedom. Two source densities that look similar to each other will have ridges of similar height, even though one is 'correct' and the other is not. For example, the densities in groups 1 and 2 (Figure 5.2) can mostly be described using only two Gaussian components. These less detailed densities would provide ridges of their own, albeit not as high as the correct ones. A model intialised near one of these would move up this lesser ridge first. Usually, there will be a smooth transition between this ridge and the correct (higher) one as there are densities that interpolate between the



(a) Mixing matrix (b) Non-isotropic noise (c) Isotropic noise

Figure 5.6: Parameter evolution.

two. Sometimes there may be a shallow plateau, as indicated by the bottom most curve in Figure 5.5(a). In most instances, the model will eventually reach the top of the correct ridge. In rare cases, though, there maybe a valley between the closest ridge and the correct one so the model will stop at the top of the first ridge. This is the case with two of the projections in Group 2 of Figure 5.3 and is due to the highly correlated nature of the projections. Such projections - in effect - warp some of the structure to the point where a different source distribution is 'better' at representing this structure. In extreme cases (see section 5.7.1), the best source density may be a Gaussian. Such behaviour, however, is very rare and can be avoided using random initialisations to start at different points in the landscape. As shown by Figure 5.5(b), the correct solution will have the highest NFE.

Mixing matrix and noise

Let $\boldsymbol{A}^0$ represent an original mixing matrix (rescaled, of course, to allow for the projection normalisation). Let $\hat{\boldsymbol{A}}$ represent a mixing matrix learnt from the data, which is also rescaled to give unit-variance source reconstructions. Then the product

$$\boldsymbol{J} = \left(\hat{\boldsymbol{A}}^T \hat{\boldsymbol{A}}\right)^{-1} \hat{\boldsymbol{A}}^T \boldsymbol{A}^0 \tag{5.68}$$

where $\hat{\boldsymbol{A}}^T$ indicates transpose, equals the identity matrix (to within a permutation) , of dimension L, for $\hat{\boldsymbol{A}} = \boldsymbol{A}^0$. The majority of the $\boldsymbol{J}$ matrices were found

to be close to the identity matrix, although the $\boldsymbol{J}$ learnt from highly correlated data were further away. Figure 5.6(a) shows the evolution of one such 2×2 $\boldsymbol{J}$ for vbICA2. It must be noted that the mixing matrix and noise precision take longer to converge than the source model. In this thesis, the source signals are the fundamental quatities to be inferred whereas the mixing matrix and noise statistics are less so. As such, the covergence criterion using $\delta F \leq 0.01\%$ was chosen to reflect that. If more accurate estimates for the mixing matrix and noise precision are needed, then $\delta F \leq 0.001\%$ or would be better, or a criterion based on the rate of change of $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{\Lambda}}$ themselves.

Both vbICA1 and vbICA2 can, in principle, estimate a non-isotropic diagonal noise precision matrix. In practice, this estimation is not robust. Figure 5.6(b) plots the evolution of the noise precisions for both vbICA1 and vbICA2. Although 5 % isotropic noise was added (equivalent to a noise precision of 20), vbICA1 does not always estimate the noise correctly; indeed, Figure 5.6(b) shows that many of the estimations are drastically overestimated. vbICA2 is more accurate, with only one estimate of one sensor being drastically overestimated. This would indicate that a single Gaussian source posterior is too severe an approximation for adequate parameter estimation. Both algorithms were found to be much better if forced to learn isotropic noise by replacing (5.51) and (5.52) above with

$$\hat{b}_{\boldsymbol{\Lambda}} = \left[\frac{1}{b_{\Lambda_0}} + \frac{1}{2} \sum_{t=1}^{T} \sum_{j=1}^{M} \langle \left(x_j^t - \hat{x}_j^t \right)^2 \rangle \right]^{-1} \tag{5.69}$$

$$\hat{c}_{\boldsymbol{\Lambda}} = c_{\Lambda_0} + \frac{TM}{2} \tag{5.70}$$

Figure 5.6(c) shows how both vbICA1 and vbICA2 are much better at estimating isotropic noise, with no drastic over-estimates, and much less variance across models. Using a full-covariance noise model similar to IFA, but with a Wishart prior distribution, may address the problem as correlated sensor noise could act as a constraint. However, the need to infer extra parameters together with the associated matrix inversions would slow down the learning, particularly for

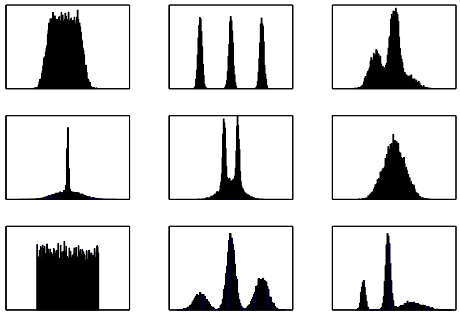


Figure 5.7: The test source densities for model selection.

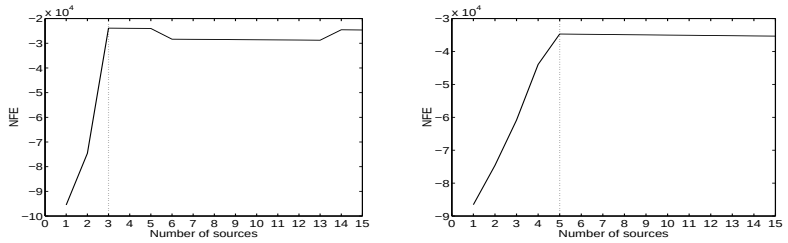
high-dimensional data. Therefore, all learning henceforth will use the isotropic noise model.

Model selection

Bayesian model selection was tested by inferring the most likely number of sources underlying nine given datasets. Each dataset was a 15-dimensional randomly-drawn mixing of 1-9 sources randomly chosen from the pdfs in Figure 5.7. Fifteen models of latent dimensionality 1-15 were trained on 5000 points drawn from each of the nine datasets, giving 135 models in total. The models were initialised using SVD and training continued until the NFE converged to within 0.01%.

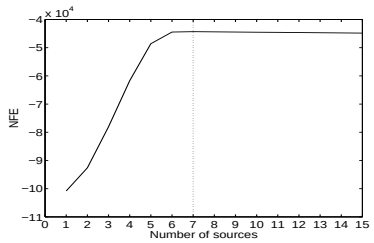
The NFE plots across models correctly inferred all dimensionalities. Figure 5.8 shows 4 of the 9 NFE curves for true latent dimensionalities of 3,5,7 and 9. Each curve has a maximum at the correct number of sources. The NFE is of the order of -10^4, so when the NFE is exponentiated, the models' posterior probabilities equal unity at the correct dimensionality.

The comparison of models can even be extended to inferring the most appropriate number of Gaussian components in each source MoG. The two source distributions shown in Figure 5.9(a) were generated using 3-component MoG models. These were mixed using a randomly generated 2×2 mixing matrix. A range of models were trained on 500 randomly drawn samples, covering 1-6 Gaussian components for each source model, giving 36 models in total. The

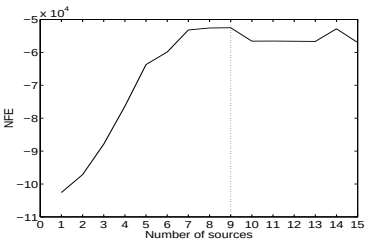


(a) 3 source

(b) 5 source



(c) 7 source



(d) 9 source

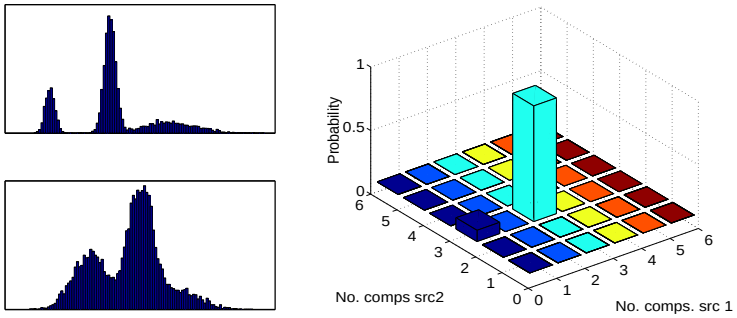
Figure 5.8: Model selection for number of sources.

posterior model probability is plotted in Figure 5.9(b). The correct 3-and-3 configuration is inferred with a probability of 0.92.

5.4.2 Comparison with other methods

The two vbICA algorithms were compared with traditional ICA algorithms. These were the entropy-maximising InfoMax (IMAX) [45], the kurtosis-based JADE [60], the negentropy-maximising FastICA [15] and the likelihood-maximising DecICA [35].

The algorithms were used to blindly separate 8 mixtures of 4 images. The original images are shown in Figure 5.10(a). These were mixed by an 8×4 mixing matrix, normalised to unit variance and centred, then 1% Gaussian noise was added. The resultant observation images are shown in Figure 5.10(b). Each image is 127×127 pixels, vectorising into 16129 data points. The dataset was thus 8 dimensional with 16129 observation vectors. Each ICA algorithm was trained on 2000 randomly drawn samples, and the complete dataset was unmixed with



(a) Source distributions

(b) Model posterior

Figure 5.9: Model selection for number of source MoG components.

the learnt unmixing matrix (or inferring the latent source signals using the learnt mixing matrix in the case of vbICA). The source reconstructions by the traditional ICA methods are shown in Figure 5.11(a)-(d). These reconstructions clearly suffer from large amounts of cross-talk and are little better than the original mixtures. Both IMAX and JADE have effectively fixed unimodal source distributions and so poorly represent the original source distributions (Figure 5.14). FastICA has a choice of non-linearities (effectively source distributions - see section 2.6.2), but all produced similar reconstructions. The results shown are those for $g(u) = u \exp\left(-a\frac{u^2}{2}\right)$ which produced the smallest reconstruction error. DecICA has an adaptable source model based on generalised exponential densities. Although still only unimodal, this limited adaptation allows DecICA to produce the most faithful reconstructions of the source images, as shown in Figure 5.11(d).

The vbICA model was trained on the same 2000 data vectors using the vbICA1 and vbICA2 algorithms. Training was intialised using SVD - with 5 Gaussian components per source for vbICA - and continued until the negative free energy, F, converged to within 0.01 %. The complete dataset was then umixed by inferring the latent source vectors using the learnt source model and (now fixed) density over the parameters. A range of models with 1-8 sources

(a) Source Images

(b) Image mixtures

Figure 5.10: Original source images and observed image mixtures.

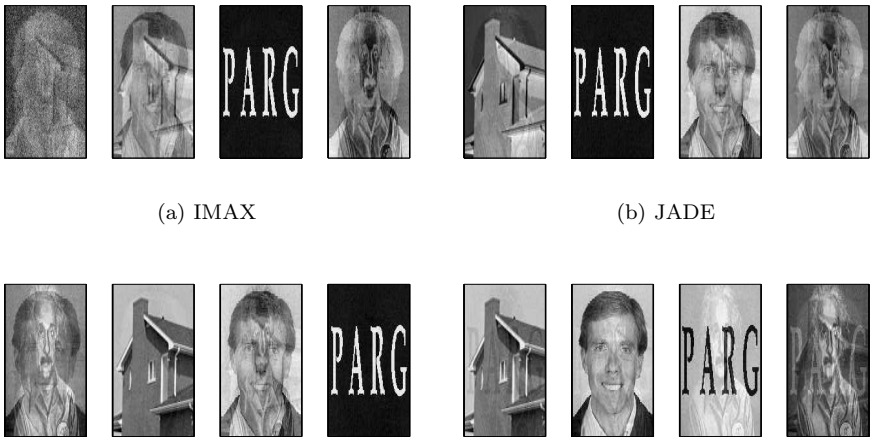
were trained; the NFE curve picked out 4 sources (see Figure 5.18(b)). The reconstructions by both vbICA1 and vbICA2 4-source models are shown in Figure 5.12. All four images have been reconstructed extremely well, with only a little cross-talk between 'House' and 'PARG' (although, interestingly, this disappears as the observation noise goes up - see section 5.4.3). At 1% sensor noise, there is no difference in the quality of reconstruction between vbICA1 and vbICA2.

To quantitatively compare between the various algorithms, the reconstruction error and the residual cross-talk were measured. The reconstruction error is quantified by the mean-square error (MSE) between the reconstructions and the original images

$$\text{MSE} = \frac{1}{LT} \sum_{i=1}^{L} \sum_{t=1}^{T} (\hat{s}_i - s_i)^2 \tag{5.71}$$

and measures how much the reconstruction differs from the original (in a Euclidean sense). For perfect separation, MSE is zero. The residual cross-talk (Xtalk) measures the quality of separation

$$\text{Xtalk} = \frac{1}{T(L^2 - L)} \sum_{i=1}^{L} \sum_{j \neq i=1}^{L} \sum_{t=1}^{T} \hat{s}_i s_j - s_i s_j \tag{5.72}$$



(a) IMAX (b) JADE

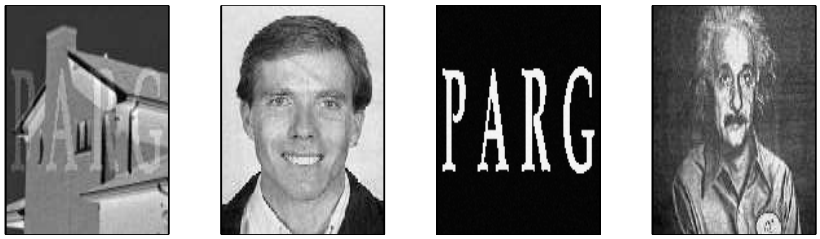
(c) FastICA (d) DecICA

Figure 5.11: Blind source separation of images by traditional ICA.

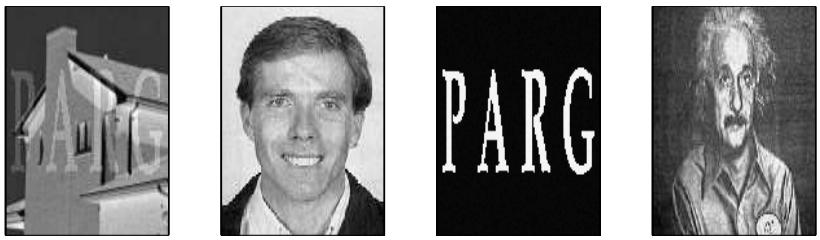
where the correlation of the original source signals is deducted to ensure their Xtalk is zero. The reconstructions were permuted according to their correlation with the original source images and rescaled and normalised where necessary to match the original source statistics.

Figure 5.13(a) plots the MSE for each image under each ICA algorithm. Apart from image 'House', both vbICA algorithms have by far the smallest MSE. The anomaly of 'House' is due to the Xtalk between 'House' and 'PARG'. Overall, vbICA has a total MSE of 0.0254, over 5 times smaller than the next best, DecICA at 0.1365. Figure 5.13(b) paints a similar picture. JADE has the smallest Xtalk of the traditional methods, with a value of 0.160. Both vbICA1 and vbICA2 have a Xtalk 0f 0.051, some 3 times smaller.

The reason for this great difference in performance is due to the flexibility of the vbICA source MoGs. Although vbICA also a proper noise model, running vbICA with unimodal MoG sources produces results similar to DecICA. This underlines the importance of having a highly adaptable source model. Whereas the traditional methods have either fixed or moderately adaptable source models, vbICA represents each source as a Mixture of Gaussians. The versatility of



(a) vbICA1

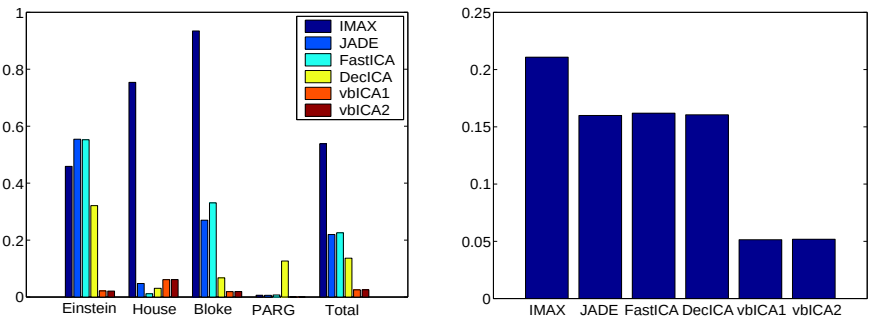


(b) vbICA2

Figure 5.12: Blind source separation of images by vbICA.

this representation is particularly highlighted for sources which have complex, multi-modal densities such as images. Figure 5.14 plots the original source densities with those learnt by vbICA, where the learnt densities have been appropriately scaled to aid comparison. The pdfs vary considerably, from the quantised pdf of 'Einstein', to the minimalist density of the two-toned 'PARG'. In all cases, the adaptability and multi-modal nature of a MoG-based source model has been fundamental in accurately modelling the true source densities and subsequent quality of reconstruction and separation. The reason for the Xtalk between 'House' and 'PARG' is due to a small bump between the two dominant peaks of the 'House' pdf which is not present in the original. This is some probablity mass that has 'leaked' from 'PARG' to 'House', leading to the Xtalk.

The source densities learnt by vbICA1 and vbICA2 differ slightly, most noticeably in their representations of 'Einstein' and 'Bloke'. Recall that vbICA1



(a) Reconstruction error

(b) Cross-talk

Figure 5.13: Comparison of reconstruction error and cross-talk.

employs a Gaussian approximation to the source posterior, while vbICA2 uses a MoG approximation. This seems to result in vbICA2 capturing more detail in the pdf structure than vbICA1, which smooths the pdfs. This is most evident in 'Einstein', where vbICA1 captures the overall envelope rather than the quantisation spikes. vbICA2, on the other hand, is more 'bumpy' in the middle and has a different tail to the left. Similarly, there are small differences for 'Bloke'. One suspects that, if enough Gaussians were stipulated, vbICA2 could capture the quantisation of 'Einstein'. This is one of the differences explored in the following section.

5.4.3 Comparison of vbICA algorithms

The difference between vbICA1 and vbICA2 lies in the posterior over $\boldsymbol{S}$: $p'(\boldsymbol{S})$ for vbICA1 and $p'(\boldsymbol{S}|\boldsymbol{q})$ for vbICA2. The explicit conditioning of $\boldsymbol{S}$ on $\boldsymbol{q}$ changes the nature of the posterior. The factorisation in (5.18) gives a Gaussian posterior over s while (5.19) gives a MoG posterior. The desire for such a posterior are two-fold. Firstly, it allows arbitrary posterior densities to be captured, something not possible under a Gaussian posterior. Secondly, comparison of the updates derived from the two factorisations implies that the vbICA2 factorisation is more robust under uncertainty. To understand this more clearly, consider the update equations for $p'(s_i^t)$ and $p'(s_i^t|\boldsymbol{q}^t)$

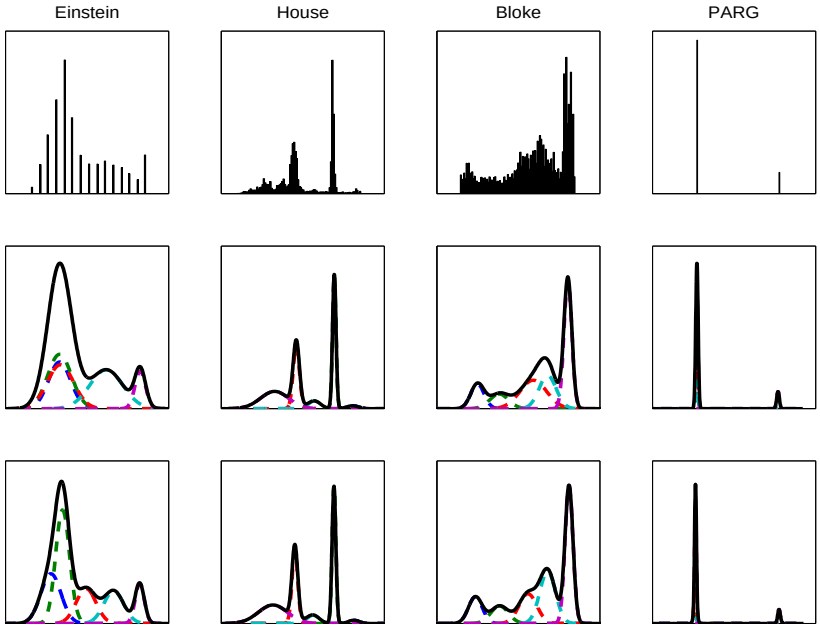


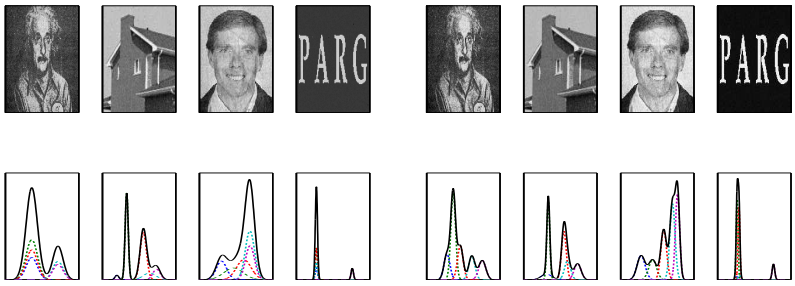
Figure 5.14: Image PDFs and vbICA models. Top: Original, Middle: vbICA1, Bottom: vbICA2.

$$\hat{\mu}_i^t \propto \sum_{q_i=1}^{m_i} \hat{\gamma}_{i,q_i}^t \langle \beta_{i,q_i} \rangle \langle \mu_{i,q_i} \rangle + \sum_{j=1}^{M} \langle \Lambda_j \rangle \langle A_{ji} \rangle \left(x_j^t - \langle \hat{x}_{j,k \neq i}^t \rangle \right) \tag{5.73}$$

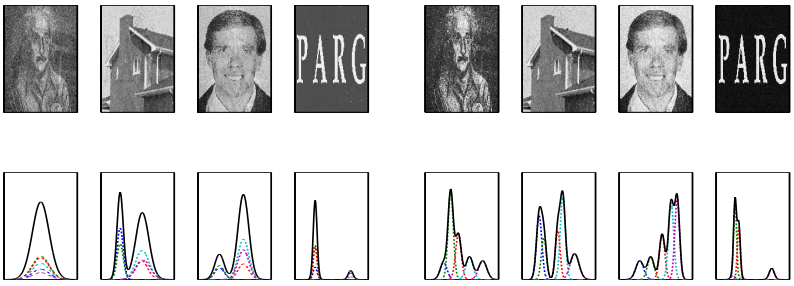
$$\hat{\mu}_{i,q_i}^t \propto \langle \beta_{i,q_i} \rangle \langle \mu_{i,q_i} \rangle + \sum_{j=1}^{M} \langle \Lambda_j \rangle \langle A_{ji} \rangle \left(x_j^t - \langle \hat{x}_{j,k \neq i}^t \rangle \right) \tag{5.74}$$

Equation (5.73) is the update for the Gaussian mean under a Gaussian source posterior while (5.74) is the update for the Gaussian mean for component q_i under a MoG posterior. The important feature to note is that the updates consist of prior (i.e. current) information from the MoG source model plus new data information. This information is combined and fed back up to the MoG source models. The MoGs then use this information to update their parameters.

In uncertain situations (i.e. high noise), $\langle \Lambda_j \rangle$ tends towards 0, drastically down-weighting the data term. In vbICA1, the MoG source models get fed a *weighted average* across MoG components of the current component parameters. This leads the MoG components to update their parameters towards *common* values. If there is little data and/or data support, the MoGs will evolve towards the centroids of their respective densities rather than staying static. Over a



(a) Source models for 10% sensor noise. Left: vbICA1, Right: vbICA2.



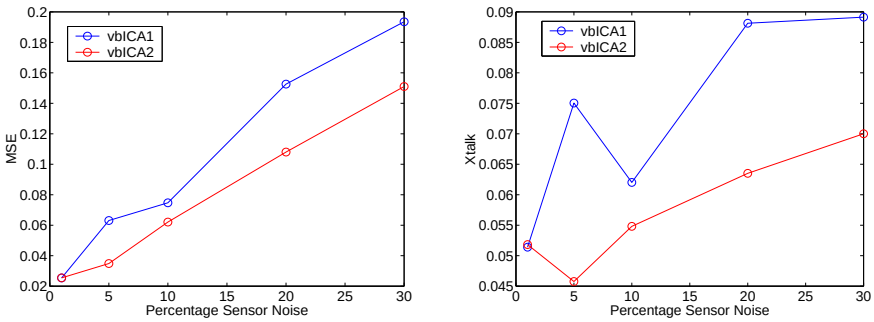
(b) Source models for 30% sensor noise. Left: vbICA1, Right: vbICA2.

Figure 5.15: Source reconstructions and models.

number of iterations, they will effectively become the same single Gaussian.

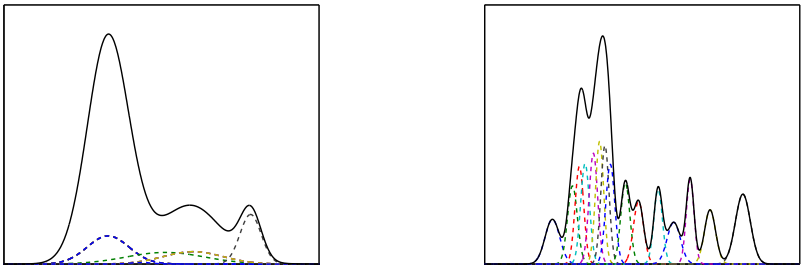
If the source *posterior* is itself a MoG, then each component of the posterior MoG is responsible for its equivalent in the source *prior* MoG. In (5.74), as $\langle \Lambda_j \rangle \to 0$, separate information is fed back to each component q_i - essentially their current parameter values. In this case, if there is little data and/or data support the source MoGs will remain static.

The two vbICA algorithms were tested for robustness under noise. The mixed images in Figure 5.10(b) with 1%, 5%, 10%, 20% and 30% Gaussian noise comprised the dataset. Both algorithms were initialised using SVD, with 5 components per source, and trained on 2000 randomly drawn points until the NFE converged to within 0.01 %. The complete source images were inferred by



(a) MSE (b) Cross-talk

Figure 5.16: Comparison of MSE and Xtalk.

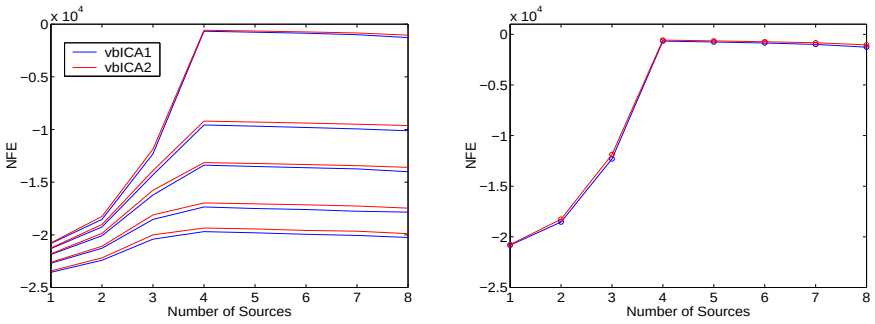


(a) vbICA1 (b) vbICA2

Figure 5.17: Source model detail.

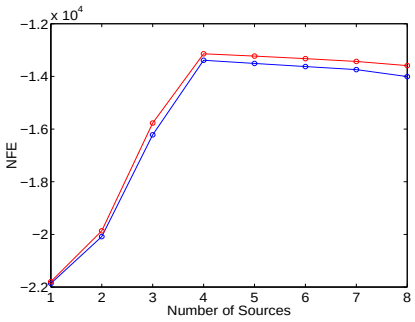
using the learnt model on the whole dataset. Figure 5.15 shows the image reconstructions together with there respective MoG models. By inspection alone, the images reconstructed by vbICA2 appear cleaner than those reconstructed by vbICA1. Certainly, the source models for vbICA2 are closer to the original densities (see Figure 5.14), particularly for 30% sensor noise. Note in particular how the 'Einstein' MoG for vbICA1 has collapsed to a single Gaussian, as previously discussed. These qualitative results are supported by quantitative measurements. Figure 5.16 plots the mean-square error and cross-talk of the sources for varying amounts of noise. vbICA2 is shown to be systematically better for both measures.

The more flexible posterior approximation in vbICA2 also allows it to cap-

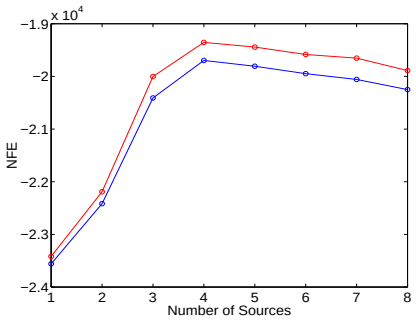


(a) All

(b) 1% noise



(c) 10% noise



(d) 30% noise

Figure 5.18: NFE for varying sensor noise.

ture more detail in the source models. Figure 5.17 shows the MoG models learnt by vbICA1 and vbICA2 for (quantised) 'Einstein' if 15 Gaussian components are stipulated. The vbICA1 model in Figure 5.17(a) has modelled the overall envelope of the original pdf (see Figure 5.14), but not the quantisation. The vbICA2 model in 5.17(b) has captured more of the quantisation, although - admittedly - not perfectly. It is this extra detail that contributes to the more accurate reconstructions by vbICA2.

These results are confirmed by plotting the NFE plots for both algorithms across models of varing order in Figure 5.18. Figure 5.18(a) shows the NFE for vbICA2 is greater than that of vbICA1 across all values of sensor noise. In particular, note how the difference between the two increases in Figures

5.18(b)-(d) as the noise increases. Also note how the NFE picks the correct number of sources, no matter how noisy the observations. This robustness under uncertainty shows that vbICA2 is the algorithm of choice for the discerning Independent Component Analyst.

The down side is speed. While vbICA1 calculates $2LT$ parameters for $p(\boldsymbol{S})$, vbICA2 calculates this amount *for each component in the MoG*. This reduction in speed is most apparent for large, high-dimensional datasets. For low dimensional, large datasets (with little noise) vbICA1 was found to be adequate. However, vbICA1 often fails to converge for high-dimensional and/or small datasets, particularly if sensor noise is above 5%. As such, the pros of vbICA2 outway its cons and is more universally applicable.

5.5 Using Prior Constraints

One of the advantages of the Bayesian formalism is the principled way in which prior knowledge and constraints can be incorporated into statistical models. This section will showcase two such constraints. The first will actively adjust the strength of the prior over the mixing matrix, allowing the number of sources to be inferred as part of the learning process. The second constraint will substitute the priors over the mixing and sources with positive-only distributions, guaranteeing that only non-negative sources and/or mixings are learnt.

5.5.1 Automatic Relevance Determination

The ability to select between candidate models - almost the *raison d'etre* of the vbICA model - is a very powerful asset. This process can be slow and cumbersome, however, as a number of individual models have to be trained before the model comparison can be carried out. If the primary motive for model order selection in ICA is to ascertain the most likely number of sources that produced the observations, then this full-scale model comparison can be short-cut by the use of *heirarchical* priors. This practice is known as Automatic Relevance Determination (ARD) [114].

Recall from section 4.4.6 the effect priors had on Gaussian components in a MoG that were not sufficiently supported by the data. Under weak priors, all components played a role in the model. As the strength of the prior increased, however, components with the least support were 'killed-off'. It is this phenomenon that is exploited in ARD. The priors over the mixing matrix elements in (5.16) have zero-mean, with precision α_{ji} over each element. This precision quantifies how strongly peaked the prior is around zero and, therefore, how harsh it is.

Note that each of the L columns of the mixing matrix is associated with one of the L source models and consider the following (constrained) prior over the mixing matrix

$$p(\boldsymbol{A}|\boldsymbol{\alpha}) = \prod_{i=1}^{L}\prod_{j=1}^{M}\mathcal{N}(A_{ji}|0,\alpha_i) \tag{5.75}$$

where the precision, α_{ji}, over each element has been replaced by a precision, α_i, over each column. Each of the L columns of $\boldsymbol{A}$ now has a precision α_i associated with it. The prior over the mixing matrix elements has a mean of zero i.e. assume the elements of $\boldsymbol{A}$ are zero unless the data says otherwise. The precisions $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_L\}$ define how strong this prior assumption is. If $\boldsymbol{\alpha}$ is too high, the prior over $\boldsymbol{A}$ is too strong so the elements will remain at zero. On the other hand, if $\boldsymbol{\alpha}$ is too low, the elements will be sensitive to noise and outliers in the data. By defining a *prior* over $\boldsymbol{\alpha}$ as well as $\boldsymbol{A}$, the values $\boldsymbol{\alpha}$ takes depends on both the prior and the data. The prior is optimised as part of the learning process, but as it is 'further up the chain', it is much less sensitive to noise. Consequently, the strength of $\boldsymbol{\alpha}$ on each of the columns of $\boldsymbol{A}$ will be a much better reflection of whether the data supports that dimension or not. Thus, the number of relevent sources may be determined automatically using ARD, as has been demonstrated by Lawrence and Bishop in their variational Bayesian ICA model with fixed MoG sources [77].

Define the prior over the precisions $\boldsymbol{\alpha}$ as a product of Gamma distributions

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^{L}\mathcal{G}(\alpha_i; b_{\alpha_i}, c_{\alpha_i}) \tag{5.76}$$

and let the factorisation over the posterior approximation be (for vbICA2)

$$p'(\boldsymbol{W}) = p'(\boldsymbol{\Lambda})p'(\boldsymbol{A})p'(\boldsymbol{\alpha})p'(\boldsymbol{S}|\boldsymbol{q})p'(\boldsymbol{q})p'(\boldsymbol{\theta}) \tag{5.77}$$

The posteriors over the mixing matrix and $\boldsymbol{\alpha}$ are the same for both vbICA1 and vbICA2 (see Appendix D.1 for derivations)

$$p'(\boldsymbol{A}) = \prod_{i=1}^{L}\prod_{j=1}^{M}\mathcal{N}(A_{ji}; \hat{m}_{A_{ji}}, \hat{\alpha}_{ji}) \tag{5.78}$$

$$p'(\boldsymbol{\alpha}) = \prod_{i=1}^{L}\mathcal{G}(\alpha_i; \hat{b}_{\alpha_i}, \hat{c}_{\alpha_i}) \tag{5.79}$$

where $\hat{m}_{A_{ji}}$ is given by (5.49) and where

$$\hat{\alpha}_{ji} = \langle\alpha_i\rangle + \langle\Lambda_j\rangle\sum_{t=1}^{t}\langle s_i^{t\,2}\rangle \tag{5.80}$$

$$\hat{b}_{\alpha_i} = \left(\frac{1}{b_{\alpha_i}} + \frac{1}{2}\sum_{j=1}^{M}\langle A_{ji}^2\rangle\right)^{-1} \tag{5.81}$$

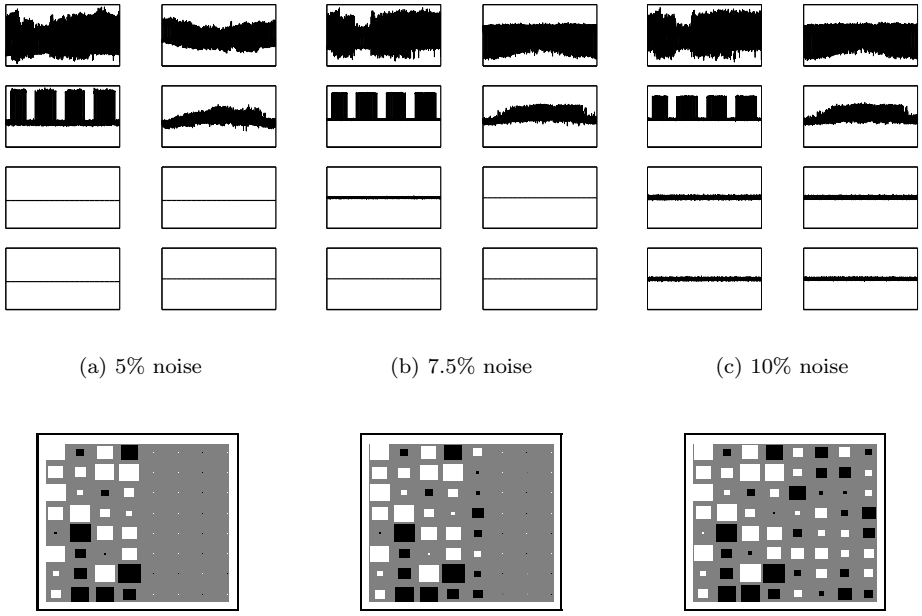
$$\hat{c}_{\alpha_i} = c_{\alpha_i} + \frac{M}{2} \tag{5.82}$$

The posterior precision, $\hat{\alpha}_{ji}$, over the mixing matrix elements now depends on a prior precision that is adapted during the learning process, embodied in $\langle\alpha_i\rangle$. If this is large, then the posterior over mixing matrix column i will be dominated by the prior density, effectively setting the elements of column i to zero. This will result in heavy supression of the i^{th} source signal. By monitoring the variance of each source signal - or, equivalently, the values of $\langle\alpha_i\rangle$ - the most likely number of sources supported by the observation data can be ascertained.

Results

The ARD extension was tested on the image mixtures in Figure 5.10 with 1%, 5%, 7.5%, 10%, 20% and 30% Gaussian noise. The vbICA2 algorithm was initialised and run under the same conditions as previously until the NFE converged to within 0.01%.

Figure 5.19 shows the source signal reconstructions by vbICA models trained on data with 5%, 7.5% and 10% noise, together with Hinton plots of their respective mixing matrices [115]. These represent the magnitude of matrix elements

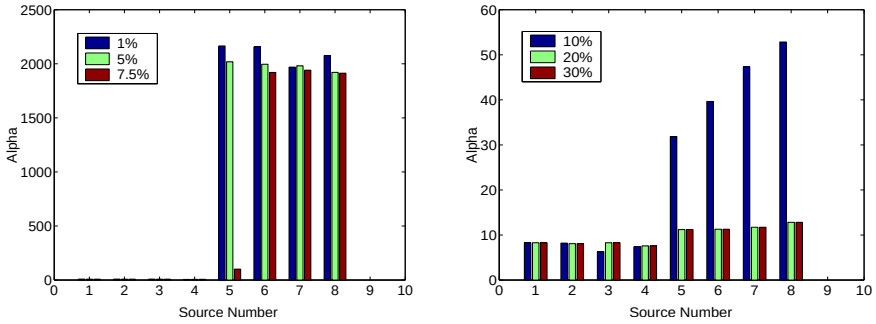


(a) 5% noise (b) 7.5% noise (c) 10% noise

Figure 5.19: Top: ARD source signal reconstructions, Bottom: Mixing matrices Hinton diagrams.

proportionally as squares of differing size, with white representing positive entries and black negative entries. ARD has supressed 4 of the 8 stipulated sources in the 5% case, clear both in the reconstructions and the mixing matrix. With 7.5% noise, one of the 'unnecessary' sources has picked up some of the noise, while all 4 superfluous sources have picked up noise in the 10% case.

Figure 5.20 plots the ARD coefficients (the $\langle \alpha_i \rangle$s) for all the models. For low noise ($< 5\%$), the alphas corresponding to unsupported sources are orders of magnitude larger than those for the supported sources, thereby stongly favouring 4 sources. For medium noise (5%-10%), this disparity is less distinct, but still differentiable. ARD is not so clever when it comes to high noise situations ($>$ 10%), suggesting that model selection via ARD is best employed in low-medium noise levels ($< 10\%$). Such situations can be ascertained from noting the learnt noise precision, although as some of the noise is syphoned off as spurious sources, the noise precision estimate, $\hat{\mathbf{\Lambda}}$, will be over-determined - the model estimates less noise than there really is (see Figure 5.21). In high noise situations, model



(a) Low-med noise

(b) Med-high noise

Figure 5.20: ARD alphas for different sensor noise.

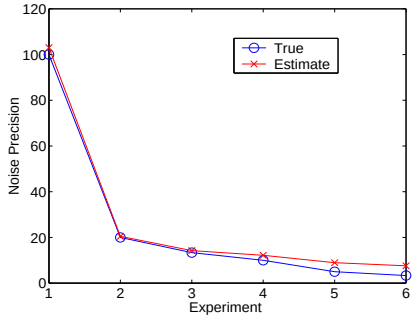


Figure 5.21: Noise precision estimates when ARD is employed.

order selection can still be carried out using the NFE, as shown in Figure 5.18. A prudent strategy would be to use ARD to narrow down the search, then use individual model selection via the NFE over the model orders of interest.

5.5.2 Positivity

A further constraint that can be imposed is that of positivity, enforcing only positive source signals and mixing. This is particularly useful when decomposing strictly non-negative data, such as word-document cocurrences in a corpus. As shown by Miskin [112], such a constraint is enforced by installing a mixture of exponentials or mixture of truncated Gaussians source model ('Mixture of Positives'), and stipulating exponential or truncated Gaussian (tGaussian) priors on the mixing matrix (see Appendix A for definitions).

A Mixture of Positives (MoP) source model is similar in form to a MoG, although without location parameters to ensure tractability

$$p(\boldsymbol{s}^n|\boldsymbol{\theta}) = \prod_{i=1}^{L}\sum_{q_i=1}^{m_i}\pi_{i,q_i}\left(\frac{\beta_{i,q_i}}{Z}\right)^{\frac{1}{\eta}}\exp\left[-\frac{\beta_{i,q_i}}{\eta}(s_i^n)^\eta\right] \tag{5.83}$$

where $\eta = Z = 1$ for exponentials and $\eta = 2$, $Z = \frac{\pi}{2}$ for truncated Gaussians. The MoGS in vbICA are substituted with (5.83), together with exponential or tGaussian priors over the mixing matrix, and the prior over the means $\boldsymbol{\mu}$ is removed. If the standard vbICA model is termed vbICA-MoG, then this model will be named vbICA-MoP.

A word of explanation regarding the sensor noise model. Due to its probabilistic nature, a noise model for vbICA must be stipulated, preferably a non-negative one. Unfortuantely, only the standard Gaussian noise model allows mathematical tractability. In principle, this violates the strictly positive constraints of the source model and mixing scheme. In practice, however, it seems to makes no difference and is a convenient way of rejecting errant negative values that might creep into the otherwise positive data.

Once the necessary modifications are made to vbICA, the vbICA1 factorisation yields the following posteriors

$$p'(\boldsymbol{S}) = \prod_{i=1}^{L}\prod_{t=1}^{T}\mathcal{N}^{(tr)}(s_i^t;\hat{\mu}_i^{(t)},\hat{\beta}_i^{(t)}) \tag{5.84}$$

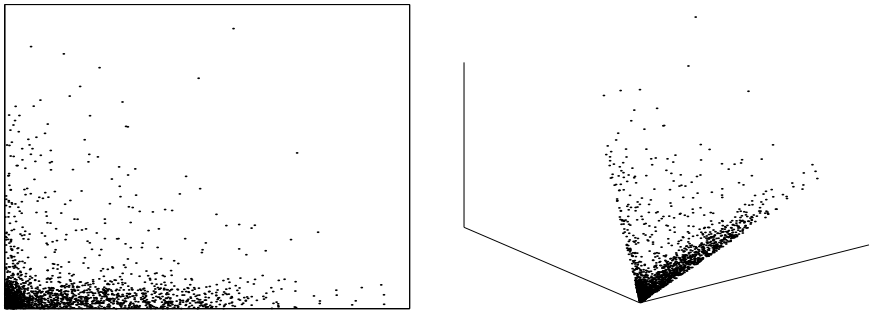
$$p'(\boldsymbol{A}) = \prod_{i=1}^{L}\prod_{j=1}^{M}\mathcal{N}^{(tr)}(A_{ji};\hat{m}_{A_{ji}},\hat{\alpha}_{ji}) \tag{5.85}$$

where $\mathcal{N}^{(tr)}(.)$ represents truncated Gaussians. The posterior updates are

$$\hat{\mu}_i^{(t)} = \frac{1}{\hat{\beta}_i}\left[-\bar{\beta}_i^{\text{expo}} + \sum_{j=1}^{M}\langle\Lambda_j\rangle\langle A_{ji}\rangle\left(x_j^t - \langle\hat{x}_{j,k\neq i}^t\rangle\right)\right] \tag{5.86}$$

$$\hat{\beta}_i = \bar{\beta}_i^{\text{tGauss}} + \sum_{j=1}^{M}\langle\Lambda_j\rangle\langle A_{ji}^2\rangle \tag{5.87}$$

where $\bar{\beta}_i^{(.)}$ is given by (5.46) only for the appropriate source model and is zero otherwise. The updates for vbICA2 are similar, with the summation in (5.46) removed. The updates for the mixing matrix are given by (5.49) and (5.50) (or



(a) Source distribution (b) Data distribution

Figure 5.22: Non-negative data.

equivalent if using ARD) if the priors are tGaussians, otherwise for exponential priors they are

$$\hat{m}_{A_{ji}} = \frac{1}{\hat{\alpha}_{ji}}\left[-\langle\alpha_i\rangle + \langle\Lambda_j\rangle\sum_{t=1}^{T}\langle s_i^t\rangle\left(x_j^t - \langle\hat{x}_{j,k\neq i}^t\rangle\right)\right] \tag{5.88}$$

$$\hat{\alpha}_{ji} = \langle\Lambda_j\rangle\sum_{t=1}^{t}\langle s_i^{t\,2}\rangle \tag{5.89}$$

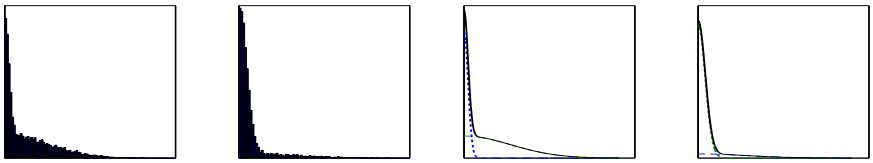
Note that the precision of the mixing matrix prior appears in the *mean* of the posterior, and *not* in the posterior precision. This implies that ARD is not suitable for non-negative ICA with an exponential mixing prior. This turns out to be the case and in fact the results are poor if ARD is attempted. ARD works normally if a truncated Gaussian mixing prior is chosen. Without location parameters in the source model, the source model updates for $\boldsymbol{q}$ and $\boldsymbol{\beta}$ are also different

$$\tilde{p}_{i,q_i} = \tilde{\beta}_{i,qi}^{\frac{1}{\eta}}\exp\left(-\frac{\langle\beta_{i,q_i}\rangle}{\eta}\langle s_i^{t^\eta}\rangle\right) \tag{5.90}$$

$$\hat{b}_{i,q_i} = \left[\frac{1}{b_{i0}} + \frac{1}{\eta}\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^t\langle s_i^{t^\eta}\rangle\right]^{-1} \tag{5.91}$$

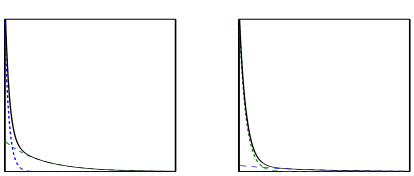
$$\hat{c}_{i,q_i} = c_{i0} + \frac{1}{\eta}\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^t \tag{5.92}$$

These updates may be substituted straight into the relevant lines in Table 5.1 without modification. Initialisation via SVD for non-negative models is not

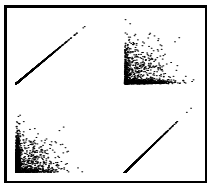


(a) Original Source pdfs

(b) Learnt MoPs - tGaussian



(c) Learnt MoPs - exponential



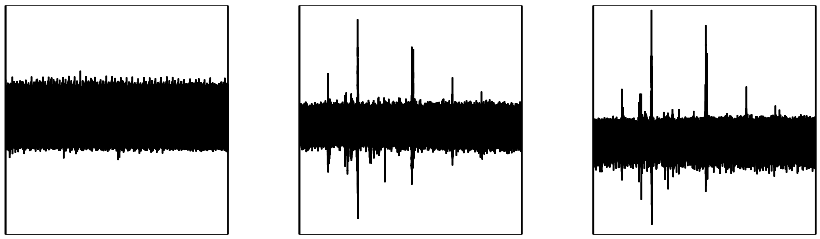
(d) Reconstruction scatter plot

Figure 5.23: Non-negative ICA results.

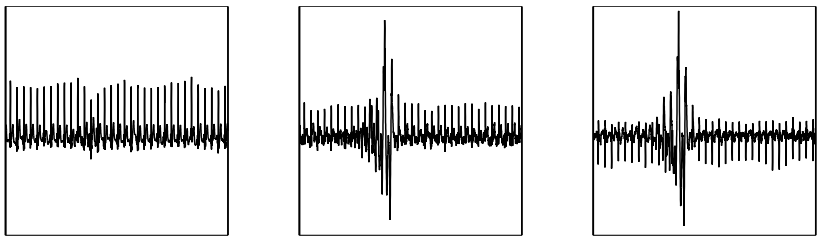
possible as SVD has no non-negative constraints, so the model must be intialised to random values.

Results

The vbICA-MoP model was tested on a simple 3-dimensional toy example. A 2-dimensional non-negative source signal was mixed into a 3-dimensional data signal. No noise was added. Figure 5.22 plots the original source distribution together with the data distribution. 2000 data vectors were drawn at random, and a truncated Gaussian model and an exponential model were trained until the NFE converged to within 0.01%. Figure 5.23 compares the learnt source models with the original source distributions, showing accurate models. The source data was originally generated from a 2-component mixture of truncated Gaussians so the truncated Gaussian-based source model in Figure 5.23(b) is particularly well matched. This is further confirmed by the NFE with $F_{\text{tGauss}} = 10952$ compared with $F_{\text{expo}} = 10161$. The tGaussian reconstructions are plotted scattered with the original source signals in Figure 5.23(d); Xtalk is only 0.0013, while MSE is vitually nil at 2×10^{-6}. For the interested reader, non-negative



(a) ECG data



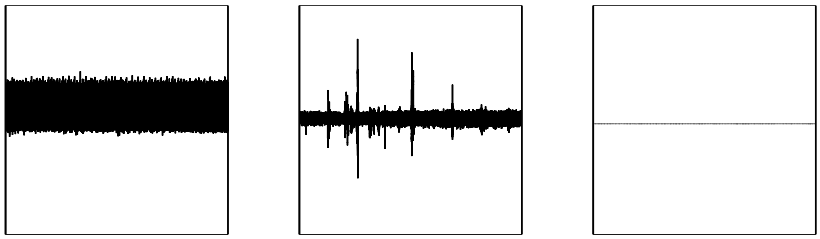
(b) ECG data - subset

Figure 5.24: Electrocardiogram signals with artifacts.

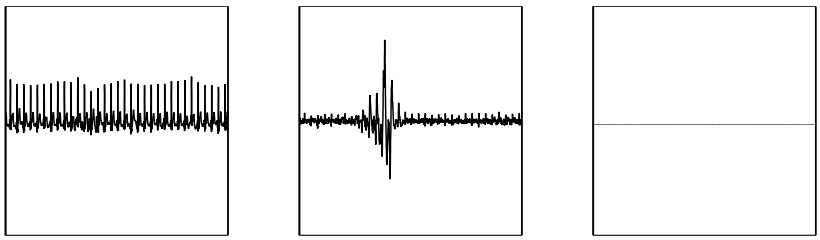
ICA (in vbICA1 guise) is explored more fully in Miskin's thesis [112].

5.6 Real Data - Removing Signal Artifacts

The analysis of biosignals is an increasingly important area of research for pattern recognition, particularly for developing automated patient monitoring and diagnosis. Real hospitals and real patients do not produce pristine signals, so these signals must be cleaned and processed before they can be analysed. An important step in this preprocessing is the removal of spurious 'artifacts' from the signal. Figure 5.24 shows 3 electrocardiogram signals collected from a patient. Each shows a varying degree of unwanted artifacts, typical 'bursts' of signal as shown in Figure 5.24(b). These artifacts are the result of electrodes being jolted. It is reasonable to assume that the process that generates these artifacts (e.g. patient rolling over, equipment being jolted etc.) is independent of the process that generates the ECG signals (i.e. the heart). Therefore, ICA



(a) vbICA sources



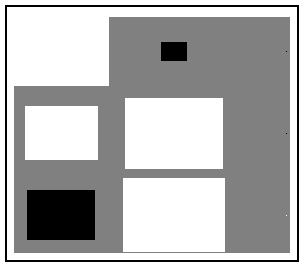
(b) vbICA sources - subset

Figure 5.25: Separated electrocardiogram and artifact signals.

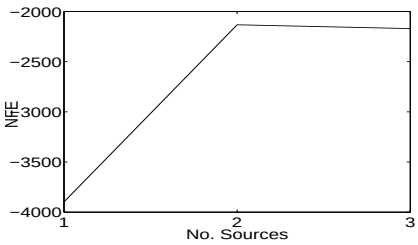
can be used to remove these artifacts.

The whole signal is 100000 samples long. A vbICA model with 3 sources and 3-components per source MoG was trained using the vbICA2 algorithm on 1000 randomly drawn vectors until the NFE converged to within 0.01 %. This took 46 iterations. The network was then used to unmix the whole dataset. The reconstructions are shown in Figure 5.25. ARD has supressed one of the sources, leaving two relevant sources - the clean ECG signal and an artifact channel. Figure 5.25(b) shows these sources in more detail by plotting the subset of the reconstructions responsible for Figure 5.24(b). This separation is not perfect as there is some cross-talk between the two sources, but nevertheless, the ECG is much cleaner. Although this may be because one of the data signals in Figure 5.24(a) is much cleaner than the other, exactly the same results were achieved by learning on only the 2 more noisy signals.

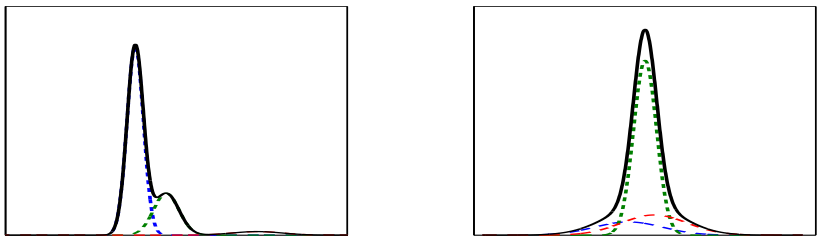
Figure 5.26(a) shows how ARD has killed one of the columns in the mixing



(a) Mixing matrix



(b) Negative free energy



(c) vbICA source MoGs

Figure 5.26: Model selection and source MoGs.

matrix, favouring 2 sources. This is confirmed by plotting the NFE for vbICA models with 1-3 sources. The MoG models for the relavent sources are plotted in Figure 5.26(c). The ECG source exhibits multi-modal structure, while the artifact source is a super-Gaussian unimodal density.

5.7 Discussion

The vbICA suite of algorithms has been shown to outperform all traditional forms of ICA. The flexible nature of the MoG sources allows vbICA to separate noisy image mixtures, which these other methods were unable to do. The Bayesian formulation allows the most likely number of components to be inferred, and different models to be quantifiably compared. This formalism also allows constraints to be imposed, as shown by ARD and positivity extensions. Of the two vbICA algorithms, vbICA2 was shown to be more accurate and robust than vbICA1. The vbICA2 model was used to remove artifacts from ECG

data, picking up 2 sources, the underlying (clean) ECG signal plus artifacts.

5.7.1 The Problem of Correlated Data

As discussed in section 5.3.5, the choice of initialisation can have an effect on the final solution. SVD intialisation always starts at the same place for a given dataset so always ends at the same solution. In practice, different random intialisations tend to lead to the same solution as well. However, if parts of the data density are highly correlated, the vbICA model will stay close to the PCA solution if initialised using SVD. Different random initialisations may lead to different soultions, or may converge onto the PCA/FA solution. This has also been found to be the case using other ICA algorithms.

ICA can be seen as a generalisation of Principle Component Analysis and Factor Analysis. As shown by Attias [46], noiseless ICA with Gaussian sources is equivalent to PCA, while FA is the same with non-isotropic noise. This means the PCA/FA solution is a subset of all possible ICA solutions for a given dataset. As a particular cluster becomes more and more correlated (or, equivalently, its independent directions become less orthogonal), the second moment starts to dominate so the cluster becomes easier to describe by a Gaussian density. This means the PCA/FA maxima in the ICA solution space become more prominent so more and more intialisation points will lead to them. The limit of correlation that vbICA can succesfully resolve seems to be about $\pm$ 0.67 and - on average - 5 random initialisations are needed before vbICA 'latches-on', as indicated by a significantly higher value for F after 1 iteration. This limit may be improved by noting the factorial approximation in (5.22) is no longer a good approximation for highly correlated data, so dropping this will also solve much of the problem, albeit with a corresponding reduction in speed. Another way of overcoming this is to decorrelate the data first.

Preprocessing by Decorrelation

One possible solution is to decorrelate or 'whiten' (also 'sphere') the observation data, then commence learning on these whitened data. Decorrelating normalises

the data and ensures the data covariance matrix equals the identity matrix. First perform an eigenvalue decomposition of the data covariance

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{E}\boldsymbol{D}\boldsymbol{E}^T \tag{5.93}$$

where N is the number of data points, ^{T} means transpose, and $\boldsymbol{E}$ and $\boldsymbol{D}$ are the eigenvectors and eigenvalues respectively. Now linearly transform the data

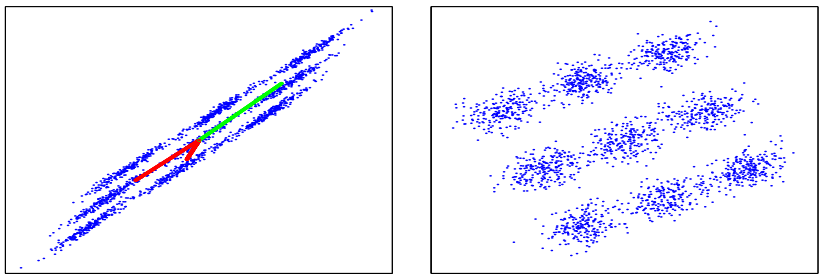
$$\tilde{\boldsymbol{X}} = \boldsymbol{E}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{E}^T\boldsymbol{X} \tag{5.94}$$

ICA may now be performed on this new decorrelated data, $\tilde{\boldsymbol{X}}$. The learnt mixing matrix, $\tilde{\boldsymbol{A}}$, is related to the 'true' matrix, $\boldsymbol{A}$, by $\tilde{\boldsymbol{A}} = \boldsymbol{E}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{E}^T\boldsymbol{A}$.

Figure 5.27(a) shows highly correlated data that cannot be resolved using vbICA. In green are the axes found - note that only the first principal component is discovered. Figure 5.27(b) plots the data distribution after decorrelation. The directions found by vbICA are plotted in red in Figure 5.27(a) - they are the correct independent directions. Decorrelation is a simple preprocessing step, and can be performed on any data. Computationally, this is more desirable than using an explicitly correlated approximation of the source posterior, although it cannot be used in the case of non-negative ICA as it won't preserve positivity. Also, decorrelation has the effect of equalising variance in *all* directions, so the latent dimensionality of the manifold may not be correctly inferred in cases where there is appreciable noise. In repeated tests, the dimensionality has been correctly inferred for most correlations *before* whitening, therefore model selection can be carried out prior to preprocessing when necessary.

5.7.2 Further Applications and Extensions

The vbICA model may be used wherever ICA is currently utilised. The advantage of vbICA over current methods is most apparent when component distributions may be multi-modal, and where the most appropriate number of components is sought. The blind separation of images can only be carried out using an ICA model with multi-modal sources. The coding of images may also



(a) Data and vbICA axes found. Green: Straight data Red: Decorrelated data

(b) Decorrelated data

Figure 5.27: Improvement when preprocessing by decorrelation.

be one such application, where knowing the ideal number of components is important in keeping the code efficient. Adaptive codes based on ICA have been shown to be more efficient than those constructed using other methods [116].

The ICA model presented in this Chapter may be extended by utilising other source models, such as binary or autoregressive sources. It may also be extended by bolting together models to form a mixture of ICAs to model multiple manifolds, or by incorporating dynamic submodels to learn posible temporal information giving rise to a better separation. Dynamic ICA is developed in Chapter 7, while mixture of ICAs is derived in the next Chapter.

Chapter 6

Mixtures of Independent Component Analysers

The goal of pattern analysis and recognition is to extract information from some data. In order for this information to be useful, the distribution of data must be represented in some meaningful way. In many cases, insight may be gained by dividing the data into self-similar areas and analysing each of these clusters under some informative framework, for example using some understanding of the assumed data generating process. One such method is to model the data as being produced by a mixture of data generators (a.k.a. analysers), where each component generator is responsible for generating a particular cluster. The problems to overcome in this *mixture modelling* are to decide how many generators are needed, where to place them, and how to adjust them to best represent the data.

Mixtures of Gaussians (MoG) are widely used throughout the fields of machine learning and statistics for data modelling, where each generator is a Gaussian density. Despite their popularity, however, MoGs suffer from two serious drawbacks. The first is that, as the dimensionality M of the problem space increases, the size of each covariance matrix, M^2, becomes prohibitively large. This can be dealt with by assuming isotropic Gaussians (i.e. ignoring the covariance structure) but this greatly reduces the flexibility of the model class. This problem has been solved by Tipping and Bishop [117] who replaced each

Gaussian with a probabilistic Principal Component Analyser (PCA) which allowed the dimensionality of each covariance to be effectively reduced whilst maintaining the richness of the model class. The model was formulated under a maximum-likelihood framework which - although efficient - does not allow one to infer the optimum number of generators needed. This mixture was modified into a Mixture of Factor Analysers (MoFA) [111] where variational Bayesian inference was used to infer the optimum number of analysers.

The second problem with MoGs is that each component is a Gaussian, a strong assumption which is often violated in many natural clustering problems [118]. Although MoGs are capable of modelling most distributions given enough components, the problem still remains of automatically grouping Gaussians which together describe some larger-scale feature. It is this second problem which is addressed in this Chapter. A solution is reached by extending the mixtures of probabilistic PCA/FA model to a Mixture of Independent Component Analysers model. Previous work [118, 119] is improved by incorporating a very flexible ICA model that can generate arbitrary densities using MoGs, and by bringing the formalism into the Bayesian arena. Bayesian inference is used to infer the optimum number of ICAs needed and automatically determine their ideal dimensionalities.

6.1 Why?

Decomposing and representing data using ICA assumes the whole data distribution is adequately described by one coordinate frame. This may not be appropriate for many problems, however. In the BSS problem, ICA assumes each data signal carries a constant mixing of source signals. Consider the scenario proposed by Lee *et al.* in [120]. There are two people talking to each other, but never at the same time, while there is music in the background. This cacophany is picked up by two microphones. At any one time, the microphones pick up a mixture of one voice and the music, or the other voice and the music, but never all three at the same time. Clearly in this case, standard ICA is an

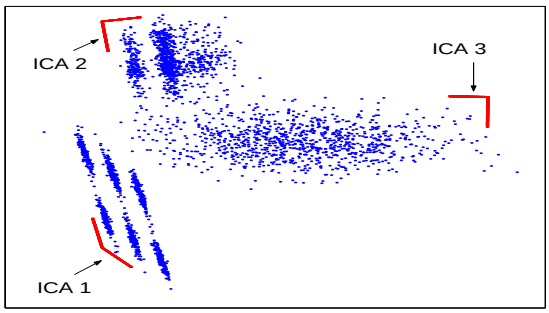


Figure 6.1: Example data distribution where multiple ICAs are better.

inadequate model. As shown in [120], a mixture of ICAs is a more appropriate generative model. More generally, if the sensor density in Figure 3.2(a) consists of various self-similar, non-Gaussian manifolds, enforcing a single, global representation is not appropriate and will produce a sub-optimal representation. A more intuitive way of representing the data is the use of a number of local coordinate frames, each constructed under local conditions, as shown in Figure 6.1. This leads to a mixture of Independent Component Analysers.

ICA mixture models were first formulated by Lee *et al.* in [65]. This model used the extended Infomax algorithm [79] to switch the source model between sub-Gaussian and super-Gaussian regimes. The model was learnt via maximum likelihood using gradient ascent. Although well demonstrated, the source model could only switch between Laplacian and bimodal densities and thus lacked flexibility. This was relaxed in [119] by utilising generalised exponential sources which can model a wide variety of unimodal densities by the adjustment of a parameter. A basic method of model selection was also incorporated using the Bayesian Information Criterion (BIC) to infer the number of hidden sources (i.e. the intrinsic dimensionality of the local manifold). Due to the problem formulation used, only 2-dimensional or higher manifolds could be modelled. Although more flexible, the densities could only be unimodal and the learning scheme was also maximum likelihood.

In this Chapter, a Mixture of ICAs model (MoICA) is presented trained using variational Bayesian methods. Each ICA component in the mixture will

be based on the ICA network developed in Chapter 5. Essentialy, each ICA component will model self-similar areas as a mixture of Gaussian sub-features. Monitoring the variational free energy of the model allows the optimum number of ICA components needed in the ICA mixture model to be inferred. Automatic Relevance Determination is used to suppress unsupported sources and thus effectively infer the local dimensionality of each ICA component as part of the learning process. This leads to the variational Bayes Mixture of Independent Component Analysers (vbMoICA) model.

6.2 The Generative Mixture Model

The probability of generating a data vector $\boldsymbol{x}^t$ from a C-component mixture model given assumptions $\mathcal{M}$ is:

$$p(\boldsymbol{x}^t|\mathcal{M}) = \sum_{c=1}^{C} p(c|\mathcal{M}_0)p(\boldsymbol{x}^t|\mathcal{M}_c, c) \tag{6.1}$$

A data vector is generated by choosing one of the C components stochastically under $p(c|\mathcal{M}_0)$ and then drawing from $p(\boldsymbol{x}^t|\mathcal{M}_c, c)$. $\mathcal{M} = \{\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_C\}$ is the vector of component model assumptions, $\mathcal{M}_c$, and assumptions about the mixture process, $\mathcal{M}_0$. The assumptions represent everything that essentially defines the model - values of fixed parameters, model structure, details of the component switching method, any prior information etc.. $p(\boldsymbol{x}^t|\mathcal{M})$ is the evidence for model $\mathcal{M}$ and quantifies the likelihood of the observed data under model $\mathcal{M}$.

The variable c indicates which component of the mixture model is chosen to generate a given data vector $\boldsymbol{x}$. If $p(c|\mathcal{M}_0)$ is a vector of probabilities and each component $p(\boldsymbol{x}^t|\mathcal{M}_c, c)$ is a Gaussian, then (6.1) simply describes a MoG. If the MoG is adapted through a maximum likelihood approach then $\mathcal{M}$ represents a list of point estimates for the corresponding parameters. In the ICA mixture model presented here, however, each component has a non-Gaussian density derived from the ICA model presented in the previous Chapter, and $\mathcal{M}$ represents assumptions concerning the distribution of possible parameter values. Figure

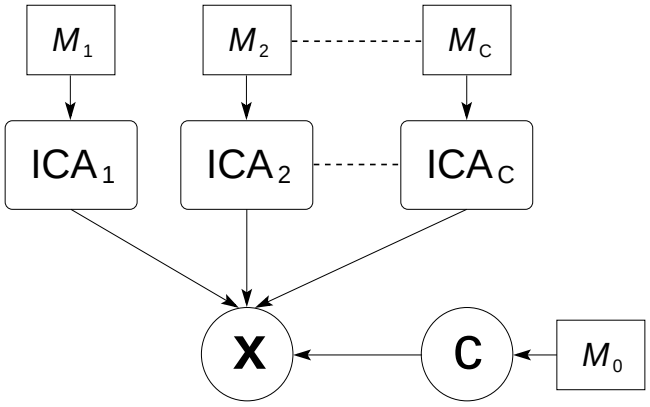


Figure 6.2: ICA mixture model.

6.2 shows a generative model for a data vector $\boldsymbol{x}$. Circular nodes represent random variables, square nodes are assumptions and rounded rectangles represent the ICA sub-networks.

As the diagram in Figure 6.2 implies, each ICA component has access to its own set of sources. This means that each manifold in the data-space is described by its own locally-adapted coordinate system, with a different distribution of points along each coordinate axis (if need be for independence). Such a mixture model will class together observations that share features with each other, but not with others. The dataset will be partitioned into C mutually exclusive classes, albeit with probabilistic memberships.

An alternative formulation would be to impose a global set of sources. Data manifolds will again be described by separate coordinate frames, but each adapted such that the distribution of points along each axis *is the same*. This has its problems, though, due to the scale indeterminancy in ICA source reconstruction. This is discussed further in section 6.5.3.

6.3 Variational Bayesian Mixture of ICAs

The observed variables, $\boldsymbol{x}$, of dimension M are modelled as a linear combination of statistically independent latent variables, $\boldsymbol{s}_c$, of dimension L_c with added

Gaussian noise

$$\boldsymbol{x} = \boldsymbol{A}_c \boldsymbol{s}_c + \boldsymbol{y}_c + \boldsymbol{n}_c \tag{6.2}$$

where $\boldsymbol{y}_c$ is an M-dimensional *bias* vector, $\boldsymbol{n}_c$ is S-dimensional additive noise and c represents the c^{th} ICA model. and L_c is the number of latent (hidden) sources.

Equation (6.2) acts as a complete description for cluster c in the data density. The bias vector, $\boldsymbol{y}_c$, defines the position of the cluster in the M-dimensional data space, $\boldsymbol{A}_c$ describes its orientation and $\boldsymbol{s}_c$ describes the underlying manifold. The noise, $\boldsymbol{n}_c$, is assumed to be Gaussian and isotropic

$$p(\boldsymbol{n}_c|0, \boldsymbol{\Lambda}_c) = \mathcal{N}(\boldsymbol{n}_c; 0, \Lambda_c I) \tag{6.3}$$

and essentially absorbs any (isotropic) Gaussianity present in the cluster.

The probability of observing data vector $\boldsymbol{x}^t$ under component c is then

$$p(\boldsymbol{x}^t|\theta_c, c) = \left(\frac{\Lambda_c}{2\pi}\right)^{\frac{M}{2}} \exp[-E_c] \tag{6.4}$$

where $\theta_c = \{\boldsymbol{A}_c, \boldsymbol{s}_c^t, \boldsymbol{\Lambda}_c\}$ and where

$$E_c = \frac{\Lambda_c}{2}(\boldsymbol{x}^t - \boldsymbol{A}_c \boldsymbol{s}_c^t - \boldsymbol{y}_c)^T (\boldsymbol{x}^t - \boldsymbol{A}_c \boldsymbol{s}_c^t - \boldsymbol{y}_c) \tag{6.5}$$

Since the sources $\boldsymbol{s}_c = \{s_{c,1}, .., s_{c,i}, .., s_{c,L_c}\}$ are - by definition - mutually independent, the distribution over $\boldsymbol{s}_c$ for data point t can be written as

$$p(\boldsymbol{s}_c^t|\mathcal{M}_{\boldsymbol{s}_c}, c) = \prod_{i=1}^{L_c} p(s_{c,i}^t|\mathcal{M}_{s_{c,i}}, c) \tag{6.6}$$

where the product runs over the L_c sources of component c, and $\mathcal{M}_{\boldsymbol{s}_c}$ is the vector of source model assumptions.

$p(\boldsymbol{s}_c^t|\mathcal{M}_{\boldsymbol{s}_c})$ is the source model for ICA component c. The source model used by [65, 118] utilised methods proposed for switching between super- and sub-Gaussian regimes [64, 79], although these depend on heuristic stability analyses. As discussed in Chapters 4 and 5, Mixtures of Gaussians allow a wide variety distribution to be modelled, so these will be used in the formalism presented here.

6.3.1 ICA Source Model

The source model for each ICA component is a factorised mixture of 1-dimensional Gaussians with L_c factors (i.e. sources) and m_i components per source (see Figure 4.2)

$$\begin{aligned} p(\boldsymbol{s}_c^t|\boldsymbol{\varphi}_c, c) &= \prod_{i=1}^{L_c} \sum_{q_i=1}^{m_i} p(q_i^t = q_i|\boldsymbol{\pi}_i, c) p(s_{c,i}^t|\varphi_{c,i}, c) \\ &= \prod_{i=1}^{L_c} \sum_{q_i=1}^{m_i} \pi_{i,q_i} \mathcal{N}(s_{c,i}^t; \mu_{i,q_i}, \beta_{i,q_i}) \end{aligned} \tag{6.7}$$

where, for brevity, the ICA component subscript c has been dropped from parameters which can be seen to belong to ICA c from context. From now on, all subscripted parameters should be assumed to belong to the c^{th} ICA model, unless otherwise stated. Equation (6.7) essentially describes the local features of cluster c - μ_{i,q_i} is the position of feature q_i w.r.t. the cluster centre, β_{i,q_i} is its size, and π_{i,q_i} its 'prominance' w.r.t. other features.

By integrating and summing over the hidden variables, $\{\boldsymbol{s}_c, \boldsymbol{q}_c\}$, the likelihood of the IID data $\boldsymbol{X} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^t\}$ given the model parameters $\boldsymbol{\Theta}_c = \{\boldsymbol{A}_c, \boldsymbol{y}_c, \Lambda_c, \boldsymbol{\varphi}_c\}$ can now be written as

$$p(\boldsymbol{X}|\boldsymbol{\Theta}_c, c) = \prod_{t=1}^{T} \sum_{\boldsymbol{q}=1}^{\boldsymbol{m}} \int p(\boldsymbol{x}^t, \boldsymbol{s}_c^t, \boldsymbol{q}_c^t|\boldsymbol{\Theta}_c, c) d\boldsymbol{s}_c \tag{6.8}$$

where $d\boldsymbol{s}_c = \prod_i ds_{c,i}$.

If a form for $p(c|\mathcal{M}_0)$ in (6.1) is stipulated as

$$p(\boldsymbol{X}|\mathcal{M}, \boldsymbol{\kappa}, \boldsymbol{\Theta}) = \sum_{c=1}^{C} p(c|\boldsymbol{\kappa}) p(\boldsymbol{X}|\boldsymbol{\Theta}_c, c) \tag{6.9}$$

where $p(c|\boldsymbol{\kappa}) = \{p(c=1) = \kappa_1, p(c=2) = \kappa_2, \ldots, p(c=C) = \kappa_C\}$ and $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \ldots, \boldsymbol{\Theta}_C\}$, then (6.4) - (6.6) and (6.7) can be substituted into (6.9) to yield a maximum likelihood model. This can then be learnt through an iteratrive process such as the Expectation-Maximisation algorithm [119] or gradient descent [65, 118]. In this Chapter, however, variational Bayesian learning is used to integrate out the parameters $\{\kappa, \boldsymbol{\Theta}\}$ in (6.9).

6.3.2 Variational Learning for vbMoICA

In the vbMoICA model, the ensemble of weights is $\boldsymbol{W} = \{\boldsymbol{c}, \boldsymbol{s}, \boldsymbol{q}, \boldsymbol{\kappa}, \boldsymbol{\Theta}\}$. The priors over each weight are those set in Chapter 5, with factorisation over the c ICA components, $p(\boldsymbol{\Theta}) = \prod_c p(\boldsymbol{\Theta}_c)$. The prior over bias $\boldsymbol{y}_c$ is a product of M zero-mean Gaussians with precisions $\boldsymbol{\tau}_{\boldsymbol{y}_c} = \{\tau_{y_1}, \ldots, \tau_{y_M}\}$. The prior over the ICA mixture indicator variables, $\boldsymbol{c} = \{c^1, c^2, \ldots, c^t\}$, simply factorises over the T data vectors

$$p(\boldsymbol{c}|\boldsymbol{\kappa}) = \prod_{t=1}^{T} \kappa_{c^t} \tag{6.10}$$

The prior over the ICA mixture coefficients κ is a symmetric Dirichlet

$$p(\kappa) = \mathcal{D}(\kappa; \iota_0) \tag{6.11}$$

The following factorisation for the approximating posteriors is chosen

$$p'(\boldsymbol{W}) = p'(\boldsymbol{c})p'(\boldsymbol{s}_c|\boldsymbol{q}_c, c)p'(\boldsymbol{q}_c|c)p'(\boldsymbol{\kappa})p'(\boldsymbol{y})p'(\boldsymbol{\Lambda})p'(\boldsymbol{A})p'(\boldsymbol{\alpha})p'(\boldsymbol{\varphi}) \tag{6.12}$$

where $p'(\boldsymbol{\varphi}) = p'(\boldsymbol{\pi})p'(\boldsymbol{\mu})p'(\boldsymbol{\beta})$ and $p'(a|b)$ is the approximating density of $p(a|b, \boldsymbol{X})$. As in section 5.3.2, the posterior over the sources and hidden states factorises

$$p'(\boldsymbol{s}_c, \boldsymbol{q}_c|c) = \prod_{i=1}^{L_c} p'(\boldsymbol{q}_i|c)p'(\boldsymbol{s}_{c,i}|q_i, c) \tag{6.13}$$

The posterior expectations of the sources given component c under this factorisation are

$$\langle s_{c,i}^t|c\rangle = \sum_{q_i=1}^{m_i} p'(q_i^t = q_i|c)\langle {s_{c,i}}^t|q_i, c\rangle \tag{6.14}$$

$$\langle {s_{c,i}^t}^2|c\rangle = \sum_{q_i=1}^{m_i} p'(q_i^t = q_i|c)\langle {s_{c,i}^t}^2|q_i, c\rangle \tag{6.15}$$

where

$$p'(q_i^t = q_i|c) = \hat{\gamma}_{i,q_i}^t \tag{6.16}$$

$$\langle s_{c,i}^t|q_i^t, c\rangle = \hat{\mu}_{i,q_i}^t \tag{6.17}$$

$$\langle {s_{c,i}^t}^2|q_i^t, c\rangle = (\hat{\mu}_{i,q_i}^t)^2 + \frac{1}{\hat{\beta}_{i,q_i}^t} \tag{6.18}$$

By substituting $p(\boldsymbol{X}, \boldsymbol{W})$ and (6.12) into (4.3), the negative variational free energy, F, for the model is

$$F_{\text{tot}} = F_{\text{mixture}} + \sum_{c=1}^{C} F_{\text{ICA}_c} \tag{6.19}$$

where

$$\begin{aligned} F_{\text{mixture}} &= F\left[\boldsymbol{c}, \boldsymbol{\kappa}\right] \\ F_{\text{ICA}_c} &= F\left[\boldsymbol{s}_c, \boldsymbol{q}_c, \boldsymbol{y}_c, \Lambda_c, \boldsymbol{A}_c, \boldsymbol{\alpha}_c, \boldsymbol{\varphi}_c\right] \end{aligned} \tag{6.20}$$

The energies in (6.20) can be further factorised into energy contributions from each parameter. By monitoring subsets of F, for example F_{ICA_c}, the effect of local assumptions can be used as well as ARD to infer the most likely number of sources in ICA component c.

Maximisation proceeds as before. As shown in section 4.2, the optimal form for each posterior is simply given by

$$p'(W_k) \propto p(W_k) \exp\left[\langle \log p(\boldsymbol{X}, \boldsymbol{W}) \rangle_{\prod_{l \neq k} p'(W_l)}\right] \tag{6.21}$$

where the index k refers to the k^{th} parameter in $\boldsymbol{W}$.

6.3.3 The Posteriors

The parameter posteriors are given by

$$p'(\boldsymbol{s}_c | \boldsymbol{q}_c, c) = \prod_{t=1}^{T} \prod_{i=1}^{L_c} \mathcal{N}(s_{c,i}^t; \hat{\mu}_{i,q_i}^t, \hat{\beta}_{i,q_i}^t) \tag{6.22}$$

$$p'(\boldsymbol{A}) = \prod_{c=1}^{C} \prod_{i=1}^{L_c} \prod_{j=1}^{M} \mathcal{N}(A_{ji}; \hat{m}_{A_{ji}}, \hat{\alpha}_{ji}) \tag{6.23}$$

$$p'(\boldsymbol{\alpha}) = \prod_{c=1}^{C} \prod_{i=1}^{L_c} \mathcal{G}(\alpha_i; \hat{b}_{\alpha_i}, \hat{c}_{\alpha_i}) \tag{6.24}$$

$$p'(\boldsymbol{q}_c | c) = \prod_{t=1}^{T} \prod_{i=1}^{L_c} \hat{\gamma}_{i,q_i}^t \tag{6.25}$$

$$p'(\boldsymbol{\pi}) = \prod_{c=1}^{C} \prod_{i=1}^{L_c} \mathcal{D}(\boldsymbol{\pi}_i; \hat{\boldsymbol{\lambda}}_{i,1:m_i}) \tag{6.26}$$

$$p'(\boldsymbol{\mu}) = \prod_{c=1}^{C} \prod_{i=1}^{L_c} \prod_{q_i=1}^{m_i} \mathcal{N}(\mu_{i,q_i}; \hat{m}_{i,q_i}, \hat{\tau}_{i,q_i}) \tag{6.27}$$

$$p'(\boldsymbol{\beta}) = \prod_{c=1}^{C}\prod_{i=1}^{L_c}\prod_{q_i=1}^{m_i}\mathcal{G}(\beta_{i,q_i};\hat{b}_{i,q_i},\hat{c}_{i,q_i}) \tag{6.28}$$

$$p'(\boldsymbol{y}) = \prod_{c=1}^{C}\prod_{j=1}^{M}\mathcal{N}(y_j;\hat{m}_{y_j},\hat{\tau}_{y_j}) \tag{6.29}$$

$$p'(\boldsymbol{\Lambda}) = \prod_{c=1}^{C}\mathcal{G}(\Lambda_c;\hat{b}_{\Lambda_c},\hat{c}_{\Lambda_c}) \tag{6.30}$$

$$p'(\boldsymbol{c}) = \prod_{t=1}^{T}\prod_{c=1}^{C}\hat{\eta}_c^t \tag{6.31}$$

$$p'(\boldsymbol{\kappa}) = \mathcal{D}(\kappa;\hat{\boldsymbol{\iota}}_{1:C}) \tag{6.32}$$

Updated hyper-parameters are hatted versions of the original parameters and $\langle a|b\rangle$ are expectations taken w.r.t. $p'(a|b)$.

Observation Model

- $p'(\boldsymbol{s}_c|\boldsymbol{q}_c,c)$

$$\hat{\mu}_{i,q_i}^t = \frac{1}{\hat{\beta}_{i,q_i}^t}\left[\langle\beta_{i,q_i}\rangle\langle\mu_{i,q_i}\rangle + \hat{\eta}_c^t\langle\Lambda_c\rangle\sum_{j=1}^{S}\langle A_{ji}\rangle\left(x_j^t - \langle\hat{x}_{j,k\neq i}^t|c\rangle - \langle y_j\rangle\right)\right] \tag{6.33}$$

$$\hat{\beta}_{i,q_i}^t = \langle\beta_{i,q_i}\rangle + \hat{\eta}_c^t\langle\Lambda_c\rangle\sum_{j=1}^{S}\langle A_{ji}^2\rangle \tag{6.34}$$

where

$$\hat{x}_{j,k\neq i}^t = \sum_{k\neq i}^{L_c} A_{jk}s_{c,k}^t$$

$$\langle\hat{x}_{j,k\neq i}^t|c\rangle = \sum_{k\neq i}^{L_c}\langle A_{jk}\rangle\langle s_{c,k}^t|c\rangle$$

and define

$$\hat{x}_j^t = \sum_{i=1}^{L_c} A_{ji}s_{c,i}^t$$

$$\langle\hat{x}_j^t|c\rangle = \sum_{i=1}^{L_c}\langle A_{ji}\rangle\langle s_{c,i}^t|c\rangle$$

In practice, (6.33) has to be iterated for every i a number of times until $\hat{\mu}_{i,q_i}^t$ converges as it depends on every other $k \neq i$.

- $p'(\boldsymbol{A}_c)$

$$\hat{m}_{A_{ji}} = \frac{\langle \Lambda_c \rangle}{\hat{\alpha}_{ji}} \sum_{t=1}^{T} \hat{\eta}_c^t \langle s_i^t | c \rangle \left(x_j^t - \langle \hat{x}_{j,k\neq i}^t | c \rangle - \langle y_j \rangle \right) \tag{6.35}$$

$$\hat{\alpha}_{ji} = \langle \alpha_i \rangle + \langle \Lambda_c \rangle \sum_{t=1}^{T} \hat{\eta}_c^t \langle {s_i^t}^2 | c \rangle \tag{6.36}$$

In practice, (6.35) has to be iterated for every i a number of times until $\hat{m}_{A_{ji}}$ converges.

- $p'(\boldsymbol{\alpha}_c)$

$$\hat{b}_{\alpha_i} = \left(\frac{1}{b_{\alpha_i}} + \frac{1}{2} \sum_{j=1}^{M} \langle A_{ji}^2 \rangle \right)^{-1} \tag{6.37}$$

$$\hat{c}_{\alpha_i} = c_{\alpha_i} + \frac{M}{2} \tag{6.38}$$

Source Model

- $p'(\boldsymbol{q}_c)$

$$\gamma_{i,q_i}^t = \tilde{\pi}_{i,q_i} \tilde{p}_{i,q_i} \tag{6.39}$$

$$\hat{\gamma}_{i,q_i}^t = \frac{\gamma_{i,q_i}^t}{\sum_{q_i'} \gamma_{i,q_i'}^t} \tag{6.40}$$

where

$$\tilde{\pi}_{i,q_i} = \exp\left[\Psi(\hat{\lambda}_{i,q_i}) - \Psi(\sum_{q_i'} \hat{\lambda}_{i,q_i'}) \right]$$

$$\tilde{p}_{i,q_i} = \left(\frac{\tilde{\beta}_{i,qi}}{\hat{\beta}_{i,qi}^t} \right)^{\frac{1}{2}} \exp\left[\frac{1}{2} \left(\hat{\beta}_{i,qi}^t \hat{\mu}_{i,q_i}^{n^2} - \langle \beta_{i,qi} \rangle \langle \mu_{i,q_i}^2 \rangle \right) \right]$$

$$\tilde{\beta}_{i,qi} = \hat{b}_{i,q_i} \exp\left[\Psi(\hat{c}_{i,q_i}) \right]$$

- $p'(\boldsymbol{\pi}_c)$

$$\hat{\lambda}_{i,q_i} = \lambda_{i,q_i} + \sum_{t=1}^{T} \hat{\gamma}_{i,q_i}^t \tag{6.41}$$

- $p'(\boldsymbol{\mu}_c)$

$$\hat{m}_{i,q_i} = \frac{1}{\hat{\tau}_{i,q_i}} \left(\tau_{i0} m_{i0} + \langle \beta_{i,q_i} \rangle \sum_{t=1}^{T} \hat{\gamma}_{i,q_i}^t \langle s_i^t | q_i^t, c \rangle \right) \tag{6.42}$$

$$\hat{\tau}_{i,q_i} = \tau_{i0} + \langle \beta_{i,q_i} \rangle \sum_{t=1}^{T} \hat{\gamma}_{i,q_i}^t \tag{6.43}$$

- $p'(\boldsymbol{\beta}_c)$

$$\hat{b}_{i,q_i} = \left(\frac{1}{b_{i0}} + \frac{1}{2}\tilde{\sigma}_{i,q_i}\right)^{-1} \tag{6.44}$$

$$\hat{c}_{i,q_i} = c_{i0} + \frac{1}{2}\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^t \tag{6.45}$$

where the average variance of component q_i in source i is defined as

$$\tilde{\sigma}_{i,q_i} = \sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^t\left(\langle {s_i^t}^2|q_i^t,c\rangle - 2\langle\mu_{i,q_i}\rangle\langle s_i^t|q_i^t,c\rangle + \langle\mu_{i,q_i}^2\rangle\right)$$

Noise Model

- $p'(y_c)$

$$\hat{m}_{y_j} = \frac{\langle\Lambda_c\rangle}{\hat{\tau}_{y_j}}\sum_{t=1}^{T}\hat{\eta}_c^t\left(x_j^t - \langle\hat{x}_j^t|c\rangle\right) \tag{6.46}$$

$$\hat{\tau}_{y_j} = \tau_{y_j} + \langle\Lambda_c\rangle\sum_{t=1}^{T}\hat{\eta}_c^t \tag{6.47}$$

- $p'(\Lambda_c)$

$$\hat{b}_{\Lambda_c} = \left[\frac{1}{b_{\Lambda_c}} + \frac{1}{2}\sum_{j=1}^{M}\sum_{t=1}^{T}\hat{\eta}_c^t\langle\left(x_j^t - \hat{x}_j^t - y_j\right)^2|c\rangle\right]^{-1} \tag{6.48}$$

$$\hat{c}_{\Lambda_c} = c_{\Lambda_c} + \frac{M}{2}\sum_{t=1}^{T}\hat{\eta}_c^t \tag{6.49}$$

ICA Mixture Update

- $p'(\boldsymbol{c})$

$$\eta_c^t = \tilde{\kappa}_c\tilde{\Lambda}_c^{\frac{M}{2}}\prod_{j=1}^{M}\exp\left[\frac{\langle\Lambda_c\rangle}{2}\langle\left(x_j^t - \hat{x}_j^t - y_j\right)^2|c\rangle\right] \tag{6.50}$$

$$\hat{\eta}_c^t = \frac{\eta_c^t}{\sum_{c=1}^{C}\eta_c^t} \tag{6.51}$$

where

$$\tilde{\kappa}_c = \exp\left[\Psi(\hat{\iota}_c) - \Psi(\sum_{c'}\hat{\iota}_{c'})\right]$$

$$\tilde{\Lambda}_c = \hat{b}_{\Lambda_c}\exp\left[\Psi(\hat{c}_{\Lambda_c})\right]$$

where (6.51) ensures that $\sum_c\hat{\eta}_c^t = 1$.

- $p'(\boldsymbol{\kappa})$

$$\hat{\iota}_c = \iota_0 + \sum_{t=1}^{T} \hat{\eta}_c^t \tag{6.52}$$

The relevant moments are given by

$$\begin{aligned} \langle a \rangle &= \text{mean}(a) \\ \langle a^2 \rangle &= \text{mean}(a)^2 + \text{variance}(a) \end{aligned}$$

Note that for a mixture with only one ICA component, $\hat{\eta}_c^t = 1$ for all t and equations (6.33)-(6.45) reduce to the update equations for a single ICA model.

The need for the factorisation in (5.19) over that in (5.18) is now more readily understood. The difference in performance is more accute in an ICA mixture model context. To understand this more clearly, consider the update equations for $p'(s_{c,i}^t)$ presented below

$$\hat{\mu}_i^t \propto \sum_{q_i=1}^{m_i} \hat{\gamma}_{i,q_i}^t \langle \beta_{i,q_i} \rangle \langle \mu_{i,q_i} \rangle + \hat{\eta}_c^t \langle \Lambda_c \rangle \sum_{j=1}^{M} \langle A_{ji} \rangle \left(x_j^t - \langle \hat{x}_{j,k \neq i}^t | c \rangle - \langle y_j \rangle \right) \tag{6.53}$$

$$\hat{\mu}_{i,q_i}^t \propto \langle \beta_{i,q_i} \rangle \langle \mu_{i,q_i} \rangle + \hat{\eta}_c^t \langle \Lambda_c \rangle \sum_{j=1}^{M} \langle A_{ji} \rangle \left(x_j^t - \langle \hat{x}_{j,k \neq i}^t | c \rangle - \langle y_j \rangle \right) \tag{6.54}$$

Equation (6.53) is the update under a Gaussian posterior while (6.54) is the update under a MoG posterior (posterior mean for component q_i).

As well as the noise estimate, $\langle \Lambda_c \rangle$, the data term is also weighted by the current ICA's responsibility for the current datum, $\hat{\eta}_c^t$. If ICA model c has little or no responsibility, its source parameters should remain static until it observes data deemed under its jurisdiction. This is only possible under a MoG posterior, via (6.54), as under a Gaussian posterior the source MoG components would evolve towards a common Gaussian.

6.3.4 Implementing vbMoICA

The measure F is maximised as discussed in section 4.2. All the derived posteriors require solving a set of coupled parameter update equations. In practice, this is best achieved by first initialising the posterior component responsibilities ($p'(\boldsymbol{c})$), use these to initialise each ICA component then commence learning on

each ICA component. These components are then used to calculate the new posterior responsibilities and the learning process is repeated until convergence. The above steps may be conveniently implemented in the algorithm shown here in pseudo-code form in Table 6.1.

```
initialise;
WHILE (ΔF_(tot) <  tolerance )

   FOR every ICA component
      WHILE (ΔF_(ica) < tolerance)
         update ICA observation model by cycling through
         equations (6.33)-(6.38) until convergence;
         update ICA source model by cycling through
         equations (6.39)-(6.45) until convergence;
         update ICA noise model using equations (6.46)-(6.49);
         calculate F^new_(ica);
         calculate ΔF_(ica) ≐ |F^new_(ica) − F^old_(ica)|;
      END WHILE;
   END FOR;

   update component indicator probabilities and parameters
   using equations (6.50)-(6.52);
   calculate F^new_tot;
   calculate ΔF_tot ≐ |F^new_tot − F^old_tot|;

END WHILE;
```

Table 6.1: Pseudo-code for vbMoICA updates.

Once trained, the model can be used to reconstruct hidden source signals (to within a scaling and permutation) given a dataset and the (now fixed) model parameter distributions by calculating $\langle \boldsymbol{c} \rangle$, $\langle \boldsymbol{q}_c \rangle$ and $\langle s_c \rangle$ under their respective posteriors.

Priors and Initialisation

As with the vbICA model, priors and intialisation have an effect on the final outcome. Unless specific priors are required, the priors should be chosen from the ranges discussed in section 5.3.5.

The choice of intialisation follows the same reasoning as section 5.3.5. The

overall mixture model is first initialised by running C-component k-means on the data. The class responsibilities generated are used to partition the data into C segments. These segments are used to initialise C ICA models either randomly or by using SVD on each segment using the procedure detailed in section 5.3.5.

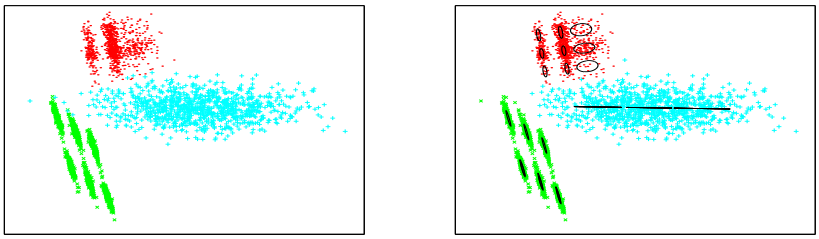
6.3.5 Results

The versatility of the vbMoICA algorithm is first demonstrated on 2- and 3-dimensional synthetic data, each grouped into 3 classes. The algorithm's ability to model intricate, highly non-Gaussian data is highlighted, as is its structure determination using ARD and the NFE F. Vague priors are set ($b = 1000$, $c = 10^{-3}$ for all Gamma distributions, scale parameter = 5 for all Dirichlets and precision = 10^{-3} for Gaussians) to encode poor prior knowledge for both the synthetic and real datasets.

Density model

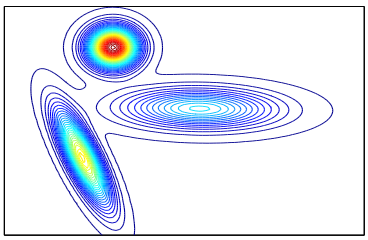
vbMoICA was tested on 2-dimensional and 3-dimensional synthetic test data drawn from three classes and with 10% added Gaussian noise. Each source MoG comprised 3 components and learning commenced using the algorithm presented in Table 6.1. Training continued until F changed by less than 0.01 percent or the number of iterations reached 200. Typically, iterations of each ICA component start high (approximately $100 - 200$ for the non-othogonal cluster presented below and < 100 for the orthogonal clusters), and end with only a handful of iterations by the end.

In the 2D case, the underlying source manifolds were 2-dimensional. Figure 6.3(a) shows the 3 clusters, with 1 Gaussian cluster, 1 non-Gaussian cluster described by orthogonal directions, and 1 highly correlated cluster made up of 6 sub-clusters. For this dataset, vbMoICA was intialised randomly. Training took 9 iterations of the vbMoICA algorithm. Figure 6.3(b) shows the partition by vbMoICA trained on 3000 points (1000 from each cluster) compared with the original data - only 12 (0.4%) data were assigned incorrectly. The 1st-

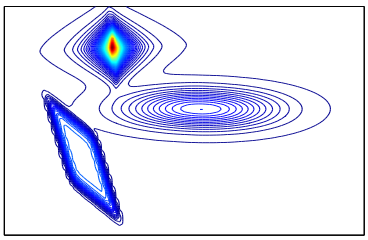


(a) Data density

(b) vbMoICA partition



(c) Gaussian mixture model



(d) gedec-ICA mixture model

Figure 6.3: 2D test data results. Contours plot $0.001 \leq p(x) \leq 1$ in 250 intervals.

deviation of the underlying MoG components are shown as ellipses. Note how the Gaussian source is deemed 1-dimensional while the 2-dimensional structure of the other classes has been clearly captured. A Gaussian source is unidentifiable from Gaussian noise in ICA. Its principal direction is captured by the single source while its 'spread' is absorbed by the (Gaussian) noise variance. The other local axes found are those in Figure 6.1. The vbMoICA data density model in Figure 6.4 shows the multi-modal structure captured within the clusters. In comparison, a MoG (Figure 6.3(c)) and the generalised exponential decorrelating ICA mixture model (gedecICA-MM) presented in [119] (Figure 6.3(d)) show no structure within the classes. As such, vbMoICA gives a more efficient representation of the true data density, with little probability mass wasted in regions of low density within each cluster.

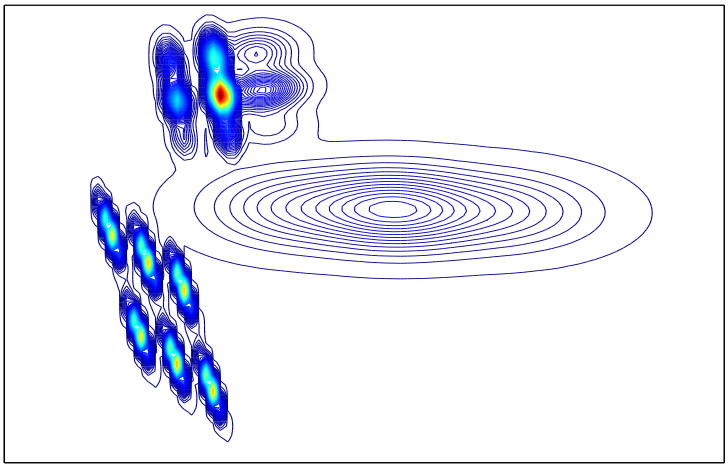
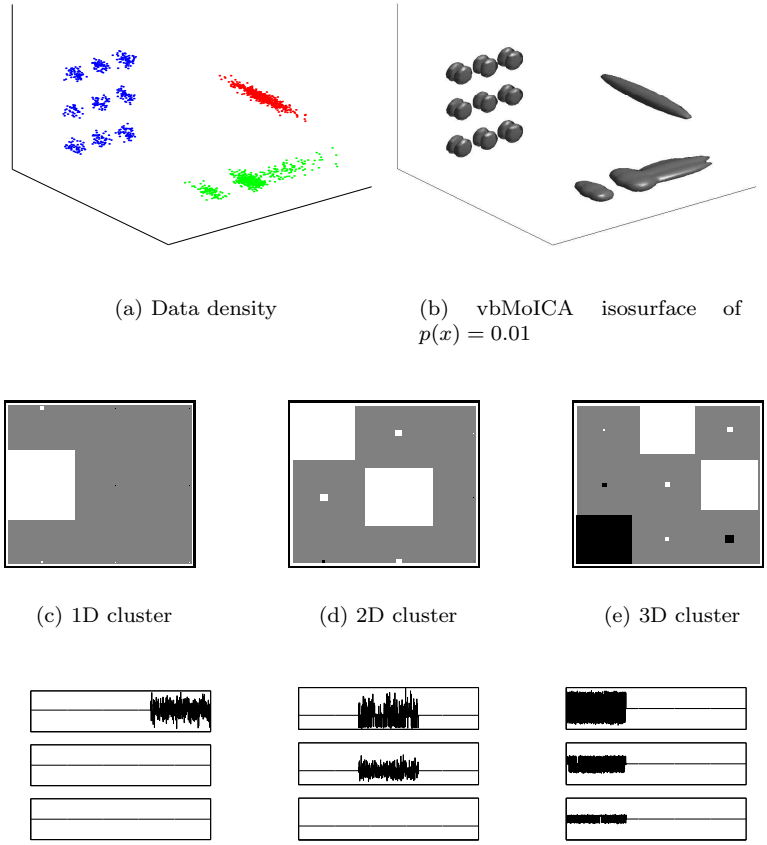


Figure 6.4: Data distribution modeled by vbMoICA.

Model order selection

The data density of the 3D test data is shown in Figure 6.5(a). The 3 clusters are intrinsically 1-, 2- and 3-dimensional. For this test, vbMoICA was initialised using SVD and trained on 1500 points, 500 drawn from each cluster. Training took 5 iterations of the vbMoICA algorithm. Figure 6.5(b) depicts the model captured by vbMoICA showing accurate representation of the clusters and cluster structures. The Hinton diagrams of the mixing matrices in Figures 6.5(c)-(e) show how vbMoICA has used ARD to correctly infer the latent dimensionalities. Unsupported columns in the matrices have been suppressed by small ARD coefficients (inverse α_i's shown in Figure 6.6(a)), effectively 'switching-off' unnecessary sources in the source reconstructions below.

Another way of inferring the latent dimensionality of each ICA component is to monitor the free-energy of the model. As implied by (6.19), this is equivalent to monitoring the contribution each ICA component makes to the overall free-energy of the model. Figure 6.6(b) is a plot of the NFE for each ICA component, F_{ICA_c}. These curves confirm the latent dimensionality inferred by ARD. Plotting the overall NFE, F, across components in Figure 6.6 correctly infers 3 clusters. There is a further peak at 5 components (and, indeed, progressively



(a) Data density

(b) vbMoICA isosurface of $p(x) = 0.01$

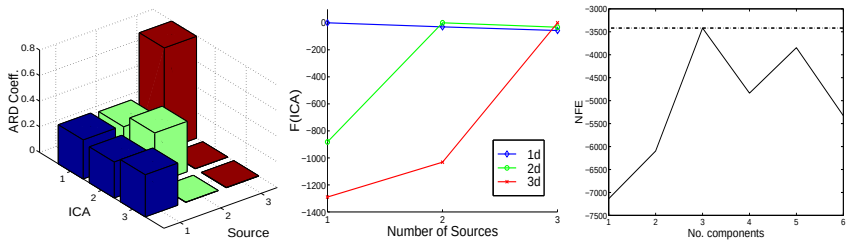
(c) 1D cluster

(d) 2D cluster

(e) 3D cluster

Figure 6.5: 3D test data results.

shallower ones at 7 and 9) as ICAs latch on to different sub-clusters in the 3D cube. It must be noted, however, that F is a bound to the *log* evidence, so these further peaks are insignificant when exponentiated to yield the evidence bound. In comparison, a Bayesian MoG infers 21 clusters, while the BIC in gedecICA-MM predicts 9. Figure 6.7 shows how well the supported sources' multi-modal densities have been modelled by the MoGs. The ability of vbMoICA to capture undulating manifolds is fundamental in correct inferring the true number of clusters.

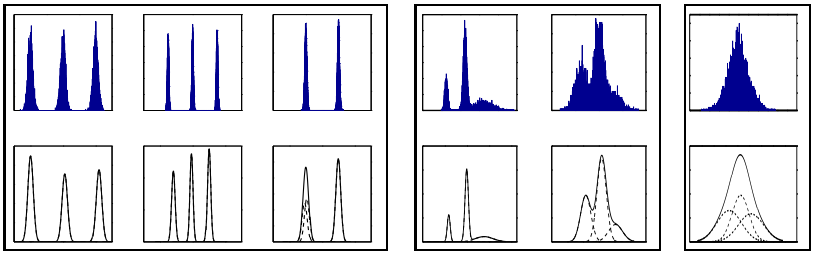


(a) ARD Coefficients

(b) F_{ICA} for each ICA component

(c) vbMoICA negative free-energy

Figure 6.6: Structure determination using ARD and negative free energy.



(a) ICA 1

(b) ICA 2

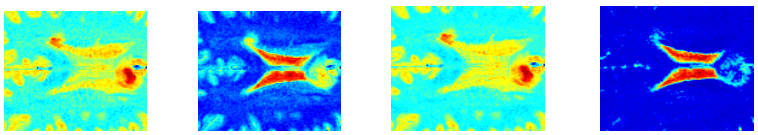
(c) ICA 3

Figure 6.7: Top: Original source distributions, Bottom: Learnt distributions.

6.4 Real data - Image Decomposition

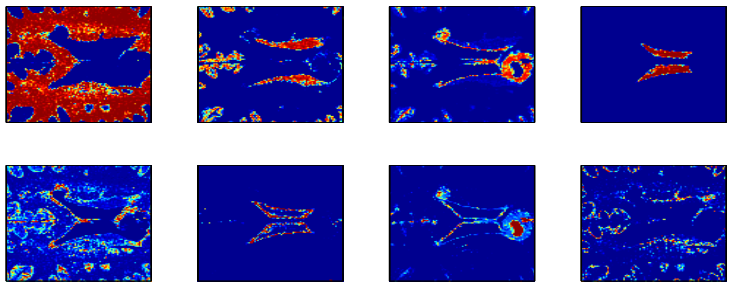
The primary aim of an ICA mixture model is to segment the data space into self-similar areas. These are self-similar in the sense that each area can be locally described by an independent coordinate frame. Crucially, data can only be drawn from one cluster at a time. Therefore, one would expect vbMoICA to be able to pick up features that suddenly appear or disappear in, say, an image. One such group of data are functional Magnetic Resonance (fMRI) images which take 'snap-shots' of the brain at regular intervals after a particular stimulus has been given.

In this section, the vbMoICA algorithm is applied to real fMRI data to decompose the images into interpretable, independent features. These are compared with the results obtained from similar subspace modelling methods. The

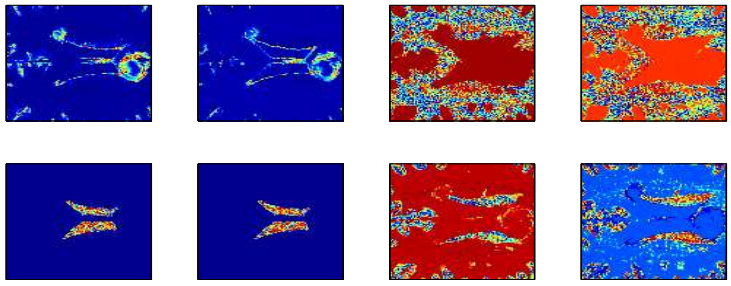


(a) Original data

(b) ICA features



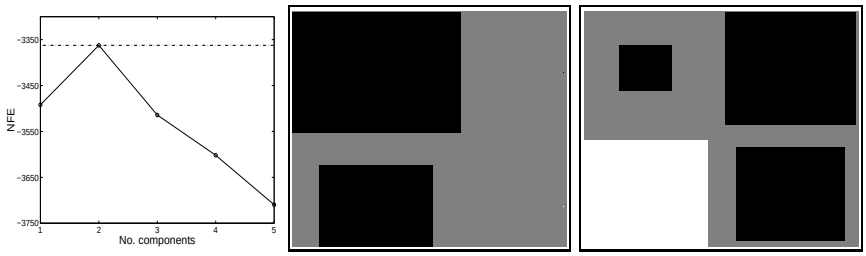
(c) vbMFA features



(d) gedecICA-MM features

Figure 6.8: fMRI images and ICA/gedecICA-MM features extracted.

data consists of fMRI images of a slice through a tumour patient's brain. Data was collected using both T2 and proton density spin sequences [121], which are used directly to form a two-dimensional feature space. Two 100x100 pixel images were vectorised, and 2000 samples were randomly drawn from the subsequent 2D data. These were analysed by the vbICA, variational Bayesian Mixture of Factor Analysers (vbMFA), gedecICA-MM and vbMoICA algorithms. Models with a range of latent dimensions were trained on the 2000 data vectors, and the most likely models were then used to unmix the complete 10000 points dataset



(a) vbMoICA negative free-energy

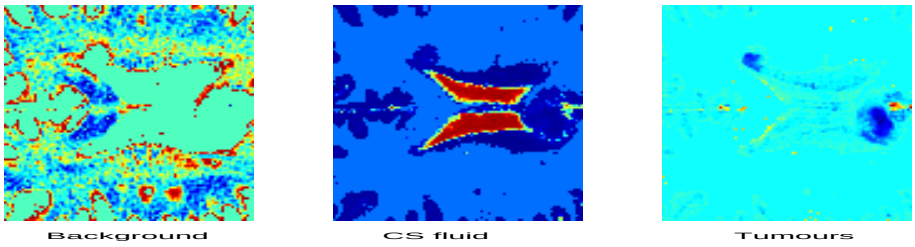
(b) 1D cluster

(c) 2D cluster

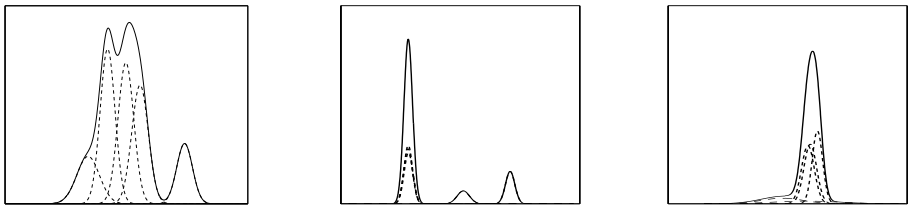
Figure 6.9: Inferred latent structure for fMRI images.

into the corresponding independent features. Figure 6.8 shows the two original images used along with the features extracted by vbICA, vbMFA and gedecICA-MM. Due to the inherent limitations of ICA, no more than two features could be extracted from the fMRI images. The vbICA algorithm favoured a 2-source ICA model, shown in Figure 6.8(b). The left-hand feature is simply a copy of one of the original images. The right-hand image has separated out a local feature, which is, in fact, cerebro-spinal fluid. The vbMFA infers an 8 component model, giving the features in 6.8(c). The Bayesian Information Criterion of the gedecICA-MM chose a 4-component model with 2 sources per ICA component, giving the eight overall features in Figure 6.8(d). Although the gedecICA-MM has managed to separate out the cerebro-spinal fluid, most features are simply scaled copies of each other and therefore the gedecICA-MM over-represents the fMRI images.

A range of models with 1-5 components were trained using vbMoICA, with each ICA component having the maximum 2 allowed sources. The NFE plot in Figure 6.9(a) shows that a 2-component model is preferred, while the Hinton plots in Figures 6.9(b)-(c) infer 1-source and 2-source ICA components. The 3 features extracted by this model are presented in Figure 6.10 along with their learnt distributons. The single source of the first ICA component - shown in Figure 6.10(a) - is global 'background' brain-tissue detail. The second ICA component represents more local features where the central part of Figure 6.10(b)



(a) vbMoICA features



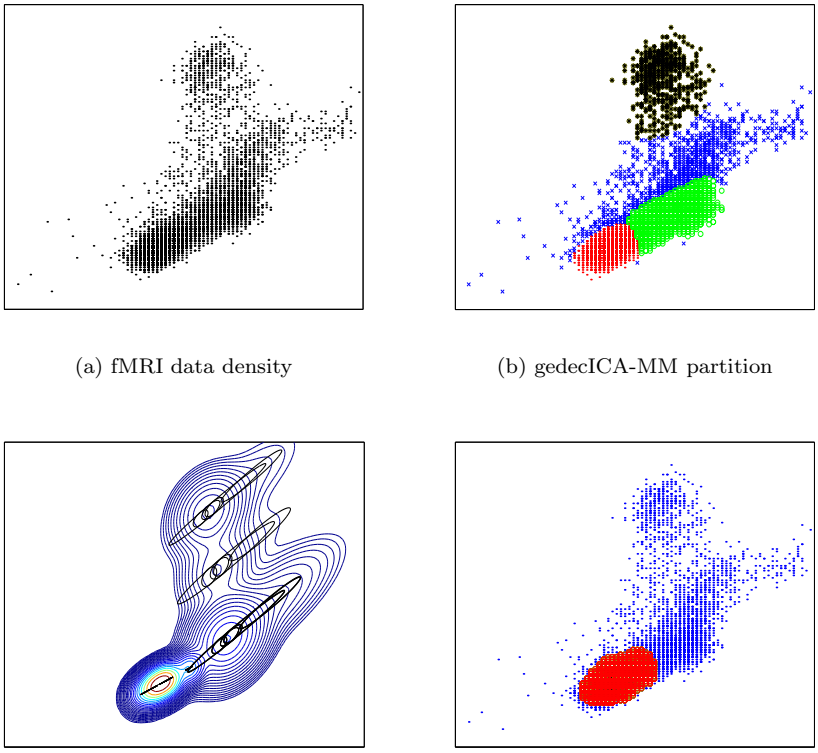
(b) ICA 1

(c) ICA 2

Figure 6.10: vbMoICA features extracted from fMRI images and respective ICA component source distributions.

is, once again, the cerebro-spinal fluid. More interestingly, however, the second source has extracted 2 dark 'blobs'. These, according to a clinician, are the tumours, which neither ICA, vbMFA or gedecICA-MM picked out. The features' respective MoGs can be interpreted as the distribution of colours in each feature. The tumours' distribution is heavily peaked around blue, with the left-hand tail capturing the yellow-green information. The other distributions similarly represent blue-red from left to right. The ability of vbMoICA to capture multi-modal feature distributions is pivotal in allowing the complex background distribution to be separated from the rest. This representation is thus much more interpretable and efficient than either simple ICA, vbMFA or the gedecICA-MM.

Figure 6.11 illustrates how the original data density has been modelled and partitioned by both gedecICA-MM and vbMoICA. The density model in Figure 6.11(c) also has the 1st deviation of the underlying MoG components superim-



(a) fMRI data density

(b) gedecICA-MM partition

(c) vbMoICA data model

(d) vbMoICA partition

Figure 6.11: vbMoICA model for fMRI images.

posed in black. Note the position of the 1D ICA (the global background image) in the bottom left segment. This leads to a partition separating this area of the data density from the rest, as depicted by the red area in Figure 6.11(d). This red area seems to be surrounded by a thin ring of blue, particularly around its lower section. These are isolated pixels that appear as dark blue specks against the light blue background in the CS fluid image (those that are not part of the dark blue brain matter). These same pixels appear as isolated yellow/orange specks against the turquoise background of the tumour image. These pixels are really part of the global background image and can be considered misclassified data vectors. The outlier data points that lie scattered to the left of the main density in Figure 6.11(d) are the small nugget of red/orange to the right of the

tumour image (and its equivalent in the CS image). Exactly what this area represents is unknown, so its classification is difficult to question. All in all, however, the overall segmentation is impressive in its interpretability, particularly as - or, perhaps, because - vbMoICA has managed to partition the data density into 2 clusters which are not linearly separable.

6.5 Discussion

In this Chapter, the vbMoICA algorithm has been comprehensively demonstrated by modelling multi-dimensional, non-Gaussian data distributions. The vbMoICA algorithm splits the distribution into self-similar areas, and uses ICA components to learn representations of these areas. The ICAs model these local cluster manifolds by forming independent directions in each underlying distribution. Automatic Relevance Determination selects the appropriate dimensionality of each manifold, and a Bayesian learning scheme allows the optimum number of ICA components to be inferred. The vbMoICA model has been demonstrated on 2-dimensional and 3-dimensional data by faithfully modelling intricate, discontinuous clusters whilst simultaneously inferring their intrinsic dimensions, something not possible under previous ICA mixture models. With vbMoICA, it is possible to compare assumptions underlying various models using the NFE, F, allowing one, for example, to ascertain the correct number of clusters. The vbMoICA model has been used to decompose real fMRI images into interpretable, self-similar features, blindly and automatically.

6.5.1 Problems

Their are two main problems when applying vbMoICA - speed and co-mean data. There are many equations, and these have to be cyclically updated to ensure convergence which can take a while if the data vectors are large in dimension and/or number. This is discussed in greater detail in Chapter 8.

The other, more fundamental problem is resolving highly overlapping distributions. Whereas a MoG can separate co-mean Gaussians provided they have

differnt variances, a vbMoICA cannot. In fact, this seems to be the case for other MoICA algorithms as well, although it is not explicitly stated in published material. Whether it is a problem in principle or with particuar formulations is not clear. This can be overcome to a certain extent, however, if there is temporal information in the data, as will be shown in the next Chapter.

6.5.2 Further Applications

Interesting data are - almost by definition - not Gaussian distributed. For data distributions that consist of self-similar non-Gaussian clusters, this model will provide a better representation over current methods, such as Gaussian mixture modelling, Mixtures of Factor Analysers and plain-vanilla ICA. The kind of data that can benefit from such an analysis range from the prosaic to the provocative.

The most obvious area is in non-stationary blind source separation, as alluded to in section 6.1. Standard ICA is now the *de jure* method for blindly separating mixtures of signals. It has shown to be very robust, effective and - in a probabilistic formulation - highly interpretable. However, ICA is essentially limited to stationary mixings of independent sources which number less than the sensors picking up the observations. ICA mixture models are an important step in overcoming these restrictions. Sudden changes in the mixing amounts will be encoded by vbMoICA as a switching between ICA components. In the cocktail party problem, for example, this would include different voices cutting in and out over a background cacophony. Provided the number of voices (plus background) active at any one time does not exceed the total number of sensors, vbMoICA can capture more sources than sensors. This was demonstrated in the fMRI example, where 3 features were extracted from 2D data. Furthermore, the independence assumption at the core of ICA holds *within* components but is not necessary *between* components. Sources which are not independent will be represented as sources within different ICA components by vbMoICA. The application of vbMoICA to enhanced Blind Source Separation does not have to stop at auditory signals. The multi-modal nature of the source distributions

lend themselves readily to image separation and other BSS problems.

Clustering in general is another area that naturally comes under vbMoICA's umbrella. Mixture modelling is the definitive formalism for finding clusters in data distributions. Interesting clusters are very often not Gaussian, so mixture models based on Gaussianity will generally not model such clusters well. The synthetic clusters in section 6.3.5 were non-Gaussian and also contained structure within themselves. This confused Gaussian-based mixture models which did not model the data accurately and failed to infer the correct number of clusters. Lee *et al.* [118] have shown that their ICA mixture model performed better at clustering the Iris data of [122] than those based on Gaussian methods. The grouping of documents into semantic clusters is currently a very active area of research (see [123] for an introduction). It is highly unlikely that such man-made corpora are intrinsically Gaussian in nature.

A related topic is non-linear ICA. In principle, the mapping from components to data need not be linear. However, solutions to the non-linear ICA probem are highly non-unique [81]. If local portions of the manifold can be assumed to be approximately linear, then an alternative approach is to model the data manifold in a piecewise manner using mixtures of ICA.

The reader may have noticed that in the use of ICA as a representative tool, the author has verged on the evangelical. This, the author strongly believes, is where ICA and its extensions will prove to be indispensable. Lee *et al.* [120] have already shown that ICA mixture models based on Laplacian source densities are over 20 percent more efficient than PCA at coding natural scenes and images of newspaper text, and more efficient than standard ICA. As ICA is only beginning to leave its BSS beginnings behind, this area is, as yet, largely unexplored.

6.5.3 Extensions

vbMoICA leads to a whole family of ICA mixture models depending on which assumptions are coded into the model. Because of the Bayesian formulation, the effectiveness of these models can be quantitatively compared. There are a

number of modifications that could be made to vbMoICA to take into account further information that one may have on the data.

One such formulation would be to impose a global set of sources. Data manifolds will again be described by separate coordinate frames, but each adapted such that the distribution of points along each axis *is the same*. The dataset will be partitioned into C classes which are *not* mutually exclusive. All observations will share common features, with the (probabilistic) class memberships dependent on how those features manifest themselves in each observation. Observations that show similar 'mixings' of features will cluster together. In principle, this is simply achieved by averaging (6.33) and (6.34) over c before presenting them to the other equations. In practice, this has been found to be difficult to implement due to the inherent ambiguities in the source reconstruction process. Recall from Chapter 2 that the sources can only be found to within a scaling and permutation. These ambiguities play havoc with the averaging with different parts of the source signal having different variances etc. Attempts have been made to overcome this by imposing normalising constraints on the mixing matrices, but to no avail. Such an extension needs more investigation.

Another modification is to utilise temporal information. Temporal information may be introduced if the mixture process prior is conditional across samples as in, for example , a Markov process, leading to a flexible Bayesian formulation of Hidden Markov ICA [124]. This is one of the models developed in Chapter 7.

Chapter 7

Dynamic Independent Component Analysis

Virtually all methods for ICA consider observed data to be independently and identically distributed and, as such, any correlations across time are deemed unimportant. Considering the wide spread and successful application of ICA to time series data, dynamics are conspicuous by their absence; considering the extra complexity involved in incorporating a dynamic element, this is understandable. Resolute blindness to temporal information may not be suitable in all cases, however. Consider the BSS problem. Speakers at a cocktail party may move around, constantly changing the mixings picked up by microphones. A static ICA algorithm would be unable to recover the sources in such an enviroment. In other observations, there may be dynamic changes in the sources themselves, affecting the source pdf statistics. In a sense, this occurs in the vectorised representation of images, such as those used as examples in Chapter 5. Natural images have localised features, each feature represented by a different part of the overall image pdf. As one moves along the image vector, a hypothetical data generator would draw from one feature 'sub-pdf' for a while, then suddenly draw from another as one moved from, say, an expanse of sky to the roof of a house, then subsequently to its brick-work etc. Such dynamics have been ignored in previous Chapters as models have been trained on data vectors randomly drawn from the observations. If, for example, ICA was carried out on

the whole, ordered dataset, one would expect a model with temporal sensitivity to out perform a static one, particularly in noisy situations when *all* information is useful. Such a model could perform filtering (e.g. noise reduction), forcasting as well as more accurate classification.

Dynamic ICA was first demonstrated by Pearlmutter and Parra's 'contextual' ICA [69] for square, noiseless mixing, where simple temporal information was incorporated into the source densities. This was improved in [125] in which Generalised Auto-Regressive (GAR) sources were used to condition source signals on previous values. A different methodology was proposed by Attias and Schreiner in [126] based on non-stationary, non-instantaneous mixing of stationary sources. This could, for example, be used for blind deconvolution. Learning non-stationary, instantaneous mixing using particle filters has been proposed by Murata *et al.* [127] and Everson and Roberts [128], the latter also making some in-roads into non-stationary sources. Penny and Roberts proposed a simple extension to Hidden Markov Models (HMMs), incorporating square, noiseless ICA observation generators with reciprical-cosh densities in [129], allowing abrupt changes in the mixing to be captured. This was extended by Penny *et al.* [124] to include ICA GAR sources. Attias extended his IFA framework [46] to include HMM sources with stationary mixing in [126]. Unlike GAR sources, HMM sources condition source signals on hidden states, which in turn are conditioned on previous hidden states, thereby capturing higher-order temporal statistics such as changes in the source statistics.

A full dynamic approach to ICA is horrendously complicated, mathematically daunting, computationally demanding and - evident from the paucity of research - seldom successful. This Chapter will take a simplified route, seeking to incorporate simple temporal information into the Independent Component Analysis process effectively by marrying the vbICA and vbMoICA models to Hidden Markov Models. HMMs are models for picking up dynamic changes of state in the underlying data generation process and are therefore useful in capturing high-order temporal information. The unobservable (hidden) underlying

process moves from state to state in a *Markov* process, whereby the occupancy of a state at time t depends only on which state the process was in at some previous time, $t - \tau$. In the case of ICA, HMM methodology will be incorporated by stipulating this Markov prior over the mixture coefficients in the ICA source models, and/or over the mixing coefficients in the mixture model presented in the preceding Chapter. The former will result in ICA with flexible, dynamic sources, and can be considered a Bayesian treatment of [126]. The latter will give a Hidden Markov Model with ICA observation generators, a flexible, non-square, noisy and Bayesian extension of [129]. In addition, these ICA generators can use dynamic HMM sources themselves, allowing simultaneous use of temporal information in the source and mixing dynamics. The proposed method is a piecewise approach to detecting dynamic movement, focussing on abrupt changes in the observation model and/or in the source model, while assuming static statistics in between. Furthermore, ICA with HMM sources can exploit temporal statistics to separate *Gaussian* signals, provided the sources have different dynamics. In this light, ICA with HMM sources can be seen as Factor analysis with non-stationary (necessarily) Gaussian sources.

The Chapter starts with a brief overview of Hidden Markov Models. The standard Forward-Backward HMM learning algorithm is derived to increase understanding. Gaussian HMMs are then recast into a Bayesian setting in section 7.3 using the variational framework. In section 7.4, these are incorporated into the vbICA model of Chapter 5, giving variational Bayesian ICA with HMM sources (vbICA-HMM as opposed to vbICA-MoG/MoP). This is then turned on its head to produce an HMM with ICA observation densities, the dynamic equivalent of Chapter 6. This is termed vbHMM-ICA. Finally, the two are combined to analyse stock prices, giving the wonderfully monikered vbHMM^{2}ICA.

7.1 Hidden Markov Models

Hidden Markov Models are an important tool for discovering structure in time-varying data. Many real-world data may be considered generated by a physical

process which can switch between a number of different states. Which state the process is in a any given time depends *only* on the state at some previous time. This is termed a Markov process. The signals[1] observed result from this process, but the process itself and its state changes are *not* observable. As the (hidden) physical process makes a path from one state to another, signals with different characteristics are emitted. The characteristics are often probabilistic relationships between the underlying state and the data generation process. Consequently, two identical state paths may produce two superficially different signals. For example, an identical word uttered by two different people won't sound the same due to differences in accent, sex, age, background noise etc., but the underlying phonetic ordering (i.e. state path) is identical. In order to extract and classify pertinant information from such signals, one must recover the underlying state dynamics which give rise to the manifest data. The difference between signal characteristics may be too subtle to pick out where these state changes occur 'by eye', so HMMs are used to model the generation process in order to try and recover the hidden states underpinning the observed sequence.

The HMM Generative Model

In essence, an HMM is a finite-state machine that switches between different probability density functions (pdfs) which represent the observation generators. The state change is a Markov process, whereby the occupancy of state d at time t depends probabilistically *only* on the state at some previous time, $t - \tau$. The model presented below is a first-order Markov process, where $\tau = 1$. Each observation is generated by first moving from state c at time $t - 1$ to state d at time t according to transition probability r_{cd}, then stochastically drawing (or picking) from the d^{th} pdf (see Figure 7.1).

The model is defined as follows. Let the model $\mathcal{M}$ have C states represented by $q = \{1, 2, \ldots, c, ..., C\}$. Let variable q^t be an C-dimensional vector that indicates which state is chosen at time t. If $\mathcal{M}$ is in state c at time t, then q^t

[1] In practice, the observed data need not be continuous - the output of the process could be a string of symbols picked from a finite dictionary.

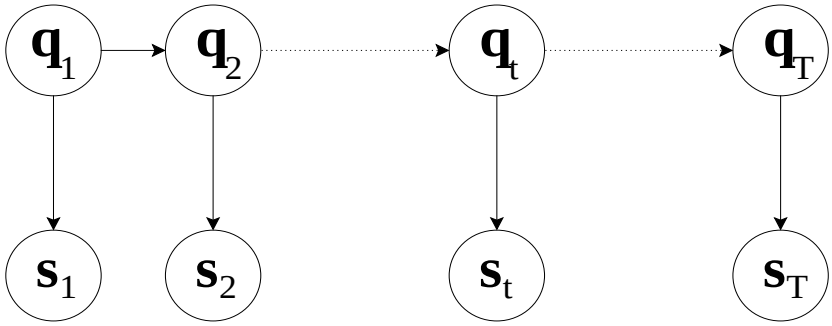


Figure 7.1: Hidden Markov Model.

has a 1 in the c^{th} entry and zeros everywhere else. Let $\boldsymbol{R}$ represent the matrix of state transition probabilities where the probability of moving from state q_c to q_d is given by r_{cd}

$$p(q^t = q_d | q^{t-1} = q_c, \boldsymbol{R}, \mathcal{M}) = r_{cd} \tag{7.1}$$

where the explicit dependence on $\mathcal{M}$ encodes the implicit assumptions underlying the model, e.g. model structure, Markov dependence etc.

The probability of $\mathcal{M}$ starting in a given state is given by the vector $\boldsymbol{\pi} = \{\pi_1, \pi_2, ..., \pi_c, ..., \pi_C\}$ where

$$p(q^1 = q_c | \boldsymbol{\pi}, \mathcal{M}) = \pi_c \tag{7.2}$$

Let the observation at time t be denoted s^t (although 1-D observations are analysed here, multi-dimensional data can also be modelled by HMMs). The probability of generating datum s^t given state $q^t = q_c$ is given by

$$p(s^t | q^t = q_c, \theta_c, \mathcal{M}) = p_c(s^t | \theta_c) \tag{7.3}$$

where θ_c represents the parameters of the c^{th} observation pdf. The complete parameter set for the C observations pdfs is $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_C\}$, and for model $\mathcal{M}$ the total parameter set is $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \boldsymbol{R}, \boldsymbol{\theta}\}$[2].

If there are T data points, then $\boldsymbol{s} = \{s^1, ..., s^T\}$ and $\boldsymbol{q} = \{q^1, ..., q^T\}$. The joint probability of state path $\boldsymbol{q}$ generating observation sequence $\boldsymbol{s}$ given pa-

[2]The parameters are, of course, model dependent. The dependency is clear from context and thus not made explicit in the notation for brevity.

rameter set $\boldsymbol{\Theta}$ is

$$\begin{aligned} p(\boldsymbol{s},\boldsymbol{q}|\boldsymbol{\Theta},\mathcal{M}) &= p(q^1|\boldsymbol{\pi},\mathcal{M})\prod_{t=2}^{T} p(q^t|q^{t-1},\boldsymbol{R},\mathcal{M})\prod_{t=1}^{T} p(s^t|q^t,\theta_c,\mathcal{M}) \\ &= p(\boldsymbol{q}|\boldsymbol{\Theta},\mathcal{M})p(\boldsymbol{s}|\boldsymbol{q},\boldsymbol{\Theta},\mathcal{M}) \end{aligned} \tag{7.4}$$

where

$$p(\boldsymbol{q}|\boldsymbol{\Theta},\mathcal{M}) = p(q^1|\boldsymbol{\pi},\mathcal{M})\prod_{t=2}^{T} p(q^t|q^{t-1},\boldsymbol{R},\mathcal{M}) \tag{7.5}$$

$$p(\boldsymbol{s}|\boldsymbol{q},\boldsymbol{\Theta},\mathcal{M}) = \prod_{t=1}^{T} p(s^t|q^t = q_c,\theta_c,\mathcal{M}) \tag{7.6}$$

The relevance of a model to a particular data sequence can be quantified using the data likelihood marginalised over all possible state paths. This is the likelihood that the data, $\boldsymbol{s}$, was generated by model $\mathcal{M}$ using parameters Θ irrespective of state path

$$p(\boldsymbol{s}|\boldsymbol{\Theta},\mathcal{M}) = \sum_{\{\boldsymbol{q}\}} p(\boldsymbol{q}|\boldsymbol{\Theta},\mathcal{M})p(\boldsymbol{s}|\boldsymbol{q},\boldsymbol{\Theta},\mathcal{M}) \tag{7.7}$$

where $\{\boldsymbol{q}\} = \{\boldsymbol{q}_1,...,\boldsymbol{q}_{C^T}\}$, the space of all state paths. Note the mixture model form of (7.7); HMMs are simply mixture models with Markov time dependencies across hidden states $\boldsymbol{q}$.

The quantity in (7.7) can be used to learn model $\mathcal{M}$ by calculating the optimal parameter values, $\hat{\boldsymbol{\Theta}}$, that maximise the data likelihood. The optimised model can then be used to recover a probability distribution over possible state paths given the data sequence observed

$$p(\boldsymbol{q}|\boldsymbol{s},\hat{\boldsymbol{\Theta}},\mathcal{M}) = \frac{p(\boldsymbol{s},\boldsymbol{q}|\hat{\boldsymbol{\Theta}},\mathcal{M})}{p(\boldsymbol{s}|\hat{\boldsymbol{\Theta}},\mathcal{M})} \tag{7.8}$$

7.2 Learning and Inference

The summation in 7.7 is over all possible state paths, a calculation that is exponential in the number of states. This is clearly problematic as model dimensionality increases, limiting the practical application of HMMs. This can be overcome, however, by making use of the HMM's Markov nature. What follows is an overview of this methodology - see [130, 131] for more details on HMMs.

7.2.1 Evaluating the likelihood

Consider the following definition

$$\alpha_c^t \doteq p(\boldsymbol{s}^{1:t}, q^t = q_c|\boldsymbol{\Theta}) \tag{7.9}$$

where $\boldsymbol{s}^{1:t} = \{s^1, ..., s^t\}$ and where the explicit dependence on model parameters and assumptions has been dropped for brevity. The *forward* variable α_c^t is the probability of $\mathcal{M}$ generating partial sequence $\boldsymbol{s}^{1:t}$ and being in state q_c at time t. Using (7.1)-(7.3), the probability of starting in state q_c and generating the first observation is

$$\alpha_c^1 = \pi_c p_c(s^1|\theta_c) \tag{7.10}$$

The probability of $\mathcal{M}$ generating $\boldsymbol{s}^{1:t}$, making a transition to q_d at time $t+1$ and generating observation s^{t+1} is

$$\alpha_d^{t+1} = \left[\sum_{c=1}^{C} \alpha_c^t r_{cd}\right] p_d(s^{t+1}|\theta_d) \tag{7.11}$$

By recursively evaluating (7.11) for $t = 1 : T$, the likelihood in (7.7) is expressed as the probability of $\mathcal{M}$ generating $\boldsymbol{s} = \boldsymbol{s}^{1:T}$ and ending in any state

$$p(\boldsymbol{s}|\boldsymbol{\Theta}, \mathcal{M}) = \sum_{c=1}^{C} \alpha_c^T \tag{7.12}$$

This forward recursion allows the likelihood in (7.7) to be calculated recursively using $\mathcal{O}(N)$ multiplications and $\mathcal{O}(N^2T)$ additions while side-stepping the need to calculate the space of all possible state paths.

7.2.2 Learning the parameters

Given the information calculated in the previous section, how are the model parameters to be adjusted so as to maximise the likelihood of $\mathcal{M}$ generating the observations?

The EM algorithm

The EM algorithm (see section 3.6.3) is an efficient way of learning parameters when there are hidden states to infer. The optimum parameters $\hat{\boldsymbol{\Theta}}$ are found by

iteratively maximising the expectation of the joint log likelihood in (7.4) under the posterior in (7.8) using the parameters from the previous iteration, $\boldsymbol{\Theta}$

$$Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}) \doteq \sum_{\{\boldsymbol{q}\}} p(\boldsymbol{q}|\boldsymbol{s}, \boldsymbol{\Theta}) \log p(\boldsymbol{s}, \boldsymbol{q}|\hat{\boldsymbol{\Theta}}) \tag{7.13}$$

$$= Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}) + Q(\boldsymbol{R}, \hat{\boldsymbol{R}}) + Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) \tag{7.14}$$

where

$$Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}) = \sum_{\{\boldsymbol{q}\}} p(\boldsymbol{q}|\boldsymbol{s}, \boldsymbol{\Theta}) \log p(q^1|\hat{\boldsymbol{\pi}}) \tag{7.15}$$

$$Q(\boldsymbol{R}, \hat{\boldsymbol{R}}) = \sum_{\{\boldsymbol{q}\}} p(\boldsymbol{q}|\boldsymbol{s}, \boldsymbol{\Theta}) \sum_{t=2}^{T} \log p(q^t|q^{t-1}, \hat{\boldsymbol{R}}) \tag{7.16}$$

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \sum_{\{\boldsymbol{q}\}} p(\boldsymbol{q}|\boldsymbol{s}, \boldsymbol{\Theta}) \sum_{t=1}^{T} \log p(s^t|q^t, \hat{\boldsymbol{\theta}}) \tag{7.17}$$

and where the explicit dependence on $\mathcal{M}$ has been dropped for brevity. As the objective function factorises over the model parameters, the optimum values can be found by maximising each objective factor separately to yield parameter update equations for each part of the model.

As discussed in section 3.6.3, the EM algorithm cycles through an Expectation step and Maximisation step until the parameter values converge. The parameters are initialised to some start values, $\boldsymbol{\Theta}$. The E-step computes the posterior $p(\boldsymbol{q}|\boldsymbol{s}, \boldsymbol{\Theta})$. The M-step finds parameter values, $\hat{\boldsymbol{\Theta}}$, that maximise the functions (7.15)-(7.17). $\boldsymbol{\Theta}$ is set to $\hat{\boldsymbol{\Theta}}$ and the EM steps are repeated. This process is repeated until convergence.

Unfortunately, the summation in (7.14) is over all possible state sequences, requiring computation that - as discussed previously - increases exponentialy with the number of states, C. Fortunately, the methodology developed in the previous section can be used to derive a more efficient procedure - the *Forward-Backward* algorithm [132].

7.2.3 The Forward-Backward algorithm

Thankfully, the methodology used in section 7.2.1 comes to the rescue, leading to the Baum-Welch or Forward-Backward (FB) algorithm for learning the

parameters. The forward-variable, α_c^t, was defined in section 7.2.1. Define the *backward* variable as the probability of generating observations $\boldsymbol{s}^{t:T}$ given that $\mathcal{M}$ is in state q_c at time t

$$\beta_c^t \doteq p(\boldsymbol{s}^{t:T}|q^t = q_c, \boldsymbol{\Theta}) \tag{7.18}$$

The probability of $\mathcal{M}$ generating the last observation while in state q_c is simply

$$\beta_c^T = p_c(s^T|\theta_c) \tag{7.19}$$

The probability of $\mathcal{M}$ generating s^t while in state q_c and subsequently generating the rest of the observation sequence is

$$\beta_c^t = p_c(s^t|\theta_c) \sum_{d=1}^{C} r_{cd}\beta_d^{t+1} \tag{7.20}$$

By recursively evaluating (7.20) for $t = T : 1$, the likelihood in (7.7) can also be expressed as the probability of $\mathcal{M}$ generating $\boldsymbol{s} = \boldsymbol{s}_{1:T}$ when starting in any state

$$p(\boldsymbol{s}|\boldsymbol{\Theta}, \mathcal{M}) = \sum_{c=1}^{C} \beta_c^1 \tag{7.21}$$

E-step

The E-step consists of calculating the posterior distribution over hidden states given the current parameter values. The forward-backward variables are used to to efficiently compute the posterior via a pair of intermediate variables. First, calculate the probability of a state change from $q^t = q_c$ to $q^{t+1} = q_d$ given the observations (and, implicitly, $\boldsymbol{\Theta}$ and $\mathcal{M}$)

$$\begin{aligned}
\xi_{cd}^t &\doteq p(q^t = q_c, q^{t+1} = q_d|\boldsymbol{s}) \\
&= \frac{p(q^t = q_c, q^{t+1} = q_d, \boldsymbol{s})}{p(\boldsymbol{s})} \\
&= \frac{p(\boldsymbol{s}^{1:t}, q^t = q_c)p(q^{t+1} = q_d|q^t = q_c)p(\boldsymbol{s}^{t:T}, q^{t+1} = q_d)}{p(\boldsymbol{s})} \\
&= \frac{\alpha_c^t r_{cd}\beta_d^{t+1}}{\sum_c \alpha_c^T}
\end{aligned} \tag{7.22}$$

Now calculate the probability of state q_c at time t given the signals

$$\gamma_c^t \doteq p(q^t = q_c|\boldsymbol{s})$$

$$
\begin{aligned}
&= \sum_{d=1}^{C} p(q^t = q_c, q^{t+1} = q_d | \boldsymbol{s}) \\
&= \sum_{d=1}^{C} \xi_{cd}^t
\end{aligned} \tag{7.23}
$$

where

$$
\gamma_c^T = \sum_{d=1}^{C} \xi_{cd}^{T-1} \tag{7.24}
$$

The variables ξ_{cd}^t and γ_c^t encapsulate all the information needed from $p(\boldsymbol{q}|\boldsymbol{s}, \boldsymbol{\Theta}, \mathcal{M})$ to take the expectations in (7.15)-(7.17).

M-step

The initial state distribution is found by maximising (7.15)

$$
Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}) = \sum_{\{\boldsymbol{q}\}} p(\boldsymbol{q}|\boldsymbol{s}, \boldsymbol{\Theta}) \log p(q^1|\hat{\boldsymbol{\pi}}) \tag{7.25}
$$

The log term only depends on the first state, so

$$
\begin{aligned}
Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}) &= \sum_{c=1}^{C} p(q^1 = q_c|\boldsymbol{s}, \boldsymbol{\Theta}) \log p(q^1|\hat{\pi}_c) \\
&= \sum_{c=1}^{C} \gamma_c^1 \log \hat{\pi}_c
\end{aligned} \tag{7.26}
$$

Maximising (7.26) leads to the update

$$
\hat{\pi}_c = \gamma_c^1 \tag{7.27}
$$

As one would expect, the probability of $\mathcal{M}$ starting in state q_c is simply given by $\gamma_1(i)$, the probability that $q^1 = q_c$ given the signals observed.

An estimate for the transition matrix is found by maximising (7.16)

$$
Q(\boldsymbol{R}, \hat{\boldsymbol{R}}) = \sum_{\{\boldsymbol{q}\}} p(\boldsymbol{q}|\boldsymbol{s}, \boldsymbol{\Theta}) \sum_{t=2}^{T} \log p(q^t|q^{t-1}, \hat{\boldsymbol{R}}) \tag{7.28}
$$

This can be rearranged as

$$
\begin{aligned}
Q(\boldsymbol{R}, \hat{\boldsymbol{R}}) &= \sum_{c=1}^{C} \sum_{d=1}^{C} \sum_{t=2}^{T} p(q^t = q_c, q^{t-1} = q_d|\boldsymbol{s}, \boldsymbol{\Theta}) \log p(q^t|q^{t-1}, \hat{\boldsymbol{R}}) \\
&= \sum_{c=1}^{C} \sum_{d=1}^{C} \left[\sum_{t=1}^{T-1} \xi_{cd}^t \right] \log \hat{r}_{cd}
\end{aligned} \tag{7.29}
$$

Maximising (7.29) leads to the update

$$\hat{r}_{cd} = \frac{\sum_t \xi_{cd}^t}{\sum_t \gamma_c^t} \tag{7.30}$$

The estimate in (7.30) for the probability of making a transition from $q^t = q_c$ to $q^{t+1} = q_d$ is the number of expected transitions from q_c to q_d over the time sequence $t = 1 : T$, divided by the total number of transitions from q_c expected under the observation sequence.

The updates for the observation model can be found by maximising

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \sum_{\{\boldsymbol{q}\}} p(\boldsymbol{q}|\boldsymbol{s}, \boldsymbol{\Theta}) \sum_{t=1}^{T} \log p(s^t|q^t, \hat{\boldsymbol{\theta}}) \tag{7.31}$$

This can be rearranged as

$$\begin{aligned} Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) &= \sum_{c=1}^{C} \sum_{t=1}^{T} p(q^t = q_c|s^t, \boldsymbol{\Theta}) \sum_{t=1}^{T} \log p(s^t|q^t, \hat{\theta}_c) \\ &= \sum_{c=1}^{C} \sum_{t=1}^{T} \gamma_c^t \log p_c(s^t|\theta_c) \end{aligned} \tag{7.32}$$

The exact form of the updates depends on the functional form for $p_c(s^t|\theta_c)$. In principle, a wide range of observation generators may be employed [129]. For example, if a Gaussian observation model is used

$$p_c(s^t|\theta_c) = \left(\frac{\beta_c}{2\pi}\right)^{\frac{1}{2}} \exp\left[-\frac{\beta_c}{2}\left(s^t - \mu_c\right)^2\right] \tag{7.33}$$

then substituting (7.33) into (7.32) and maximising with respect to $\hat{\mu}_c$ and $\hat{\sigma}_c$ gives the following intuitive updates

$$\hat{\mu}_c = \frac{\sum_t \gamma_c^t s^t}{\sum_t \gamma_c^t} \tag{7.34}$$

$$\frac{1}{\hat{\beta}_c} = \frac{\sum_t \gamma_c^t (s^t - \hat{\mu}_c)^2}{\sum_t \gamma_c^t} \tag{7.35}$$

These are the same updates as those for a maximum-likelihood MoG. Therefore - and as implied by (7.7) - learning an HMM is the same as learning a mixture model, except that the mixture coefficients are computed using the forward-backward algorithm. The newly updated parameters are now used to recompute the E-step followed by a new M-step. The cyclic process is continued until the parameters converge.

7.2.4 Inferring the state path

Once an HMM is learnt, it can be used to infer the most probable hidden state, q^{t^*}, that generated a given observation, s^t

$$\begin{aligned} q^{t^*} &= \arg\max_c [p(q^t = q_c|\boldsymbol{s})] \\ &= \arg\max_c [\gamma_c^t] \end{aligned} \tag{7.36}$$

where the explicit dependence on $\boldsymbol{\Theta}$ and $\mathcal{M}$ have been dropped for brevity. The collection $\{q^{1^*}, ..., q^{t^*}\}$ is not, however, equivalent to the most probable state sequence, $\boldsymbol{q}^*$, as it doesn't take into account any temporal information, i.e. the probability of jumping between two successive states. Two such states maybe highly likely given the observations in isolation, but highly unlikely given the observations in succession if the probability of transitioning between them is small. The path sought is

$$\boldsymbol{q}^* = \arg\max_c [p(\boldsymbol{q}|\boldsymbol{s})] \tag{7.37}$$

This requires a computation that is exponential in C. This can be overcome (again) using the inductive methodology employed for learning, leading to the *Viterbi* algorithm [133].

Viterbi algorithm

The factorisation implied by the Markov process - together with the time-invariant model probabilities - are used to compute $\boldsymbol{q}^*$ in $\mathcal{O}(TC^2)$ time. The principle underlying a first-order HMM is that the observation at time t *only* depends on the state at time t which *only* depends on the state at time $t-1$. This means the most probable state sequence leading up to time T, $\boldsymbol{q}^{1:T^*}$ is equivalent to the most probable state sequence up to $T-1$, $\boldsymbol{q}^{1:T-1^*}$, multiplied by the the most probable state transition and subsequent generation of s^T.

Define the intermediate probability $\delta_t(j)$ as

$$\delta_d^t \doteq \max_{\boldsymbol{q}^{1:t-1}} [p(\boldsymbol{q}^{1:t-1}, q^t = q_d, s^t)] \tag{7.38}$$

$$= \max_{\boldsymbol{q}^{1:t-1}} [p(\boldsymbol{q}^{1:t-1}|\boldsymbol{s}^{1:t-1}) p(q^t = q_d|q^{t-1} = q_c) p(s^t|q^t)] \tag{7.39}$$

The probability in (7.39) is maximised when $\boldsymbol{q}^{1:t-1} = \boldsymbol{q}^{1:t-1^*}$, the most probable state sequence up to $t-1$. The expression in (7.39) can therefore be expressed recursively

$$\delta_d^t = \max_c [\delta_c^{t-1} r_{cd}] p_d(s^t|\theta_d) \tag{7.40}$$

At each iteration, the most probable preceding state to reach state q_d at time t (given observations uptil time $t-1$) must be stored for later reference

$$\Psi_d^t = \arg\max_c [\delta_c^{t-1} r_{cd}] \tag{7.41}$$

The Viterbi algorithm is initialised by setting

$$\delta_d^1 = \pi_d p_d(s^t|\theta_d) \tag{7.42}$$

$$\Psi_d^1 = 0; \tag{7.43}$$

After initialisation, (7.40) is iterated for $t = 2 : T$, storing the maximising arguments in Ψ_d^t. At time T, the index of the largest δ^T is taken as the ending state

$$\boldsymbol{q}^*(T) = \arg\max_c [\delta_c^T] \tag{7.44}$$

The most likely preceding state is recalled from Ψ^T

$$\boldsymbol{q}^*(T-1) = \Psi^T [\boldsymbol{q}^*(T)] \tag{7.45}$$

This is then used to recall the most likely state preceding time $T-1$ from Ψ^{T-1} etc. until time $t = 1$,thereby recovering the most likely hidden state path, $\boldsymbol{q}^*$, that generated the observations, $\boldsymbol{s}$.

7.3 Bayesian Formalism

The HMM learning formalism derived above employs a maximum-likelihood regime. A Bayesian scheme can be used to introduce prior knowledge, control complexity, perform model selection, and integrate out the dependency on the parameters in (7.7)

$$\begin{aligned} p(\boldsymbol{s}|\mathcal{M}) &= \int_{\boldsymbol{\Theta}} p(\boldsymbol{s}|\boldsymbol{\Theta}, \mathcal{M}) p(\boldsymbol{\Theta}|\mathcal{M}) d\boldsymbol{\Theta} \\ &= \sum_{\{\boldsymbol{q}\}} \int_{\boldsymbol{\Theta}} p(\boldsymbol{q}|\boldsymbol{\Theta}, \mathcal{M}) p(\boldsymbol{s}|\boldsymbol{q}, \boldsymbol{\Theta}, \mathcal{M}) p(\boldsymbol{\Theta}|\mathcal{M}) d\boldsymbol{\Theta} \end{aligned} \tag{7.46}$$

As ever, a strict Bayesian treatment is computationally expensive and is often intractable. Therefore, a variational Bayesian approximation can be used to allow tractability and increase computational efficiency.

$$p(\boldsymbol{s}|\mathcal{M}) \geq \langle \log p(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{\Theta}|\mathcal{M}) \rangle_{p'(\boldsymbol{q},\boldsymbol{\Theta})} + \mathcal{H}[p'(\boldsymbol{q}, \boldsymbol{\Theta})] \tag{7.47}$$

If $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \boldsymbol{R}, \boldsymbol{\theta}\}$, then the following factorisation is chosen

$$p'(\boldsymbol{q}, \boldsymbol{\Theta}) = p'(\boldsymbol{q})p'(\boldsymbol{\pi})p'(\boldsymbol{R})p'(\boldsymbol{\mu})p'(\boldsymbol{\beta}) \tag{7.48}$$

where $\boldsymbol{\mu} = \{\mu_1, ..., \mu_C\}$ and $\boldsymbol{\beta} = \{\beta_1, ..., \beta_C\}$, the observation model parameters. The parameter prior distributions are a Dirichlet for the initial-state and state-transition probabilities, a Gaussian for the observation density mean and a Gamma for the observation density precision (inverse variance)

$$p(\boldsymbol{\pi}) = \mathcal{D}(\boldsymbol{\pi}; \lambda_0) \tag{7.49}$$

$$p(\boldsymbol{R}) = \prod_{c=1}^{C} \mathcal{D}(\boldsymbol{R_c}; \boldsymbol{\iota}_{c,1:C}) \tag{7.50}$$

$$p(\boldsymbol{\mu}) = \prod_{c}^{C} \mathcal{N}(\mu_c; m_0, \tau_0) \tag{7.51}$$

$$p(\boldsymbol{\beta}) = \prod_{c}^{C} \mathcal{G}(\beta_c; b_0, c_0) \tag{7.52}$$

where bold-face indicates a vector of C parameters and where $\boldsymbol{R_c}$ is the c^{th} row of $\boldsymbol{R}$. Substituting the priors and (7.48) into the negative free energy (7.47) gives the following posteriors

$$p'(\boldsymbol{\pi}) = \mathcal{D}(\boldsymbol{\pi}; \hat{\boldsymbol{\lambda}}_{1:C}) \tag{7.53}$$

$$p'(\boldsymbol{R}) = \prod_{c=1}^{C} \mathcal{D}(\boldsymbol{R_c}; \hat{\boldsymbol{\iota}}_{c,1:C}) \tag{7.54}$$

$$p'(\boldsymbol{q}) = \frac{1}{Z_{\boldsymbol{q}}} \left[\tilde{\pi}_{q^1} \prod_{t=2}^{T} \tilde{r}_{q^{t-1}q^t} \prod_{t=1}^{T} \tilde{p}_{q^t} \right] \tag{7.55}$$

$$p'(\boldsymbol{\mu}) = \prod_{c}^{C} \mathcal{N}(\mu_c; \hat{m}_c, \hat{\tau}_c) \tag{7.56}$$

$$p'(\boldsymbol{\beta}) = \prod_{c}^{C} \mathcal{G}(\beta_c; \hat{b}_c, \hat{c}_c) \tag{7.57}$$

Where the posterior parameters are given by

$$\hat{\lambda}_c = \lambda_0 + \gamma_c^1 \tag{7.58}$$

$$\hat{\iota}_{cd} = \iota_{c0} + \sum_{t=1}^{T-1} \xi_{cd}^t \tag{7.59}$$

$$\tilde{\pi}_c = \exp\left[\Psi(\hat{\lambda}_c) - \Psi(\sum_{c'} \hat{\lambda}_{c'})\right] \tag{7.60}$$

$$\tilde{r}_{cd} = \exp\left[\Psi(\hat{\iota}_{cd}) - \Psi(\sum_{d'} \hat{\iota}_{cd'})\right] \tag{7.61}$$

$$\tilde{p}_c = \tilde{\beta}_c^{\frac{1}{2}} \exp\left[-\frac{\langle\beta_c\rangle}{2}\langle\left(s^t - \mu_c\right)^2\rangle\right] \tag{7.62}$$

$$\tilde{\beta}_c = \hat{b}_c \exp\left[\Psi(\hat{c}_c)\right] \tag{7.63}$$

$$\hat{m}_c = \frac{1}{\hat{\tau}_c}\left(\tau_0 m_0 + \langle\beta_c\rangle \sum_{t=1}^{T} \gamma_c^t s^t\right) \tag{7.64}$$

$$\hat{\tau}_c = \tau_0 + \langle\beta_c\rangle \sum_{t=1}^{T} \gamma_c^t \tag{7.65}$$

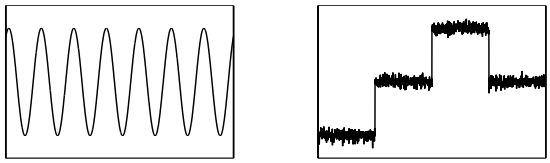
$$\hat{b}_c = \left[\frac{1}{b_0} + \frac{1}{2}\sum_{t=1}^{T} \gamma_c^t \langle\left(s^t - \mu_c\right)^2\rangle\right]^{-1} \tag{7.66}$$

$$\hat{c}_c = c_0 + \frac{1}{2}\sum_{t=1}^{T} \gamma_c^t \tag{7.67}$$

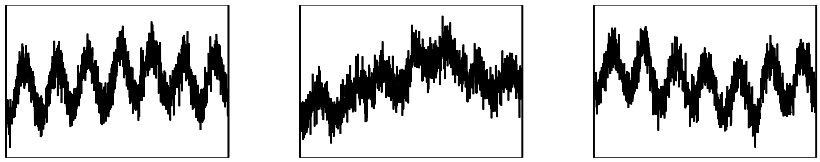
where $\langle a\rangle$ indicates expectations w.r.t. $p'(a)$ and $\Psi(.)$ is the digamma function. Normalisation of (7.55) is assured by normalising the γ_c^t's in the forward-backward algorithm. The update equations above are coupled so must be solved iteratively. The variables ξ_{cd}^t and γ_c^t are calculated from the standard forward-backward algorithm in section 7.2, with the 'tilded' estimates above, (7.60)-(7.62), substituting the maximum-likelihood estimates. These are then used to cyclically iterate through the updates above. The forward-backward pass is repeated and the process continues until convergence.

7.4 Dynamic ICA Models

As the previous section shows, HMMs are simply mixture models where the probability of (hidden) states are coupled across time. From this observation, it is clear that HMMs can be married to ICA in one of two ways: HMMs can



(a) Source signals



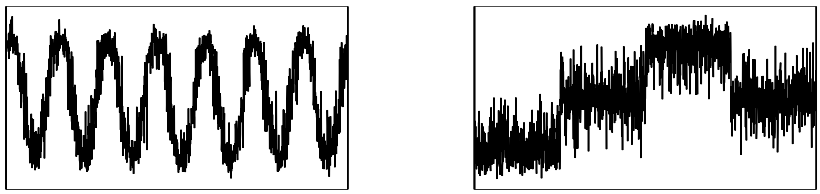
(b) Mixtures

Figure 7.2: Original signals and mixtures.

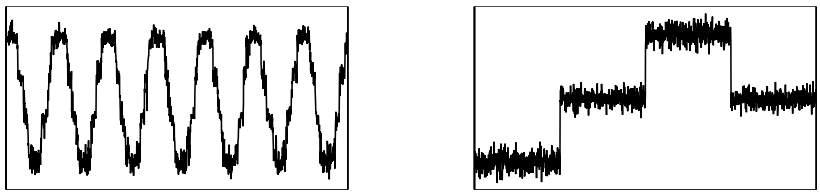
replace the temporally-blind MoGs/MoPs as ICA source models, and/or the Gaussian observation model in the previous section can be replaced with non-Gaussian, multivariate ICA observation models. The former is developed in the next section, while the latter is derived in section 7.4.2.

7.4.1 ICA with HMM Sources

ICA with HMM sources have been developed in maximum-likelihood guise by Attias [134]. Variational Bayesian ICA with HMM sources (vbICA-HMM) is simply a case of substituting (5.8) for (7.4)-(7.6) and re-deriving the appropriate equations. This gives (7.58)-(7.67) averaged over $p'(\boldsymbol{S}|\boldsymbol{q})$ (in the case of vbICA2), with q_i identified with c, additional factorisation over $i = \{1 \dots L\}$ sources, and with (7.62) replaced by $\tilde{p}_c$ for the appropriate model, i.e. vbICA1/2, MoG or MoP etc. The updates for the source model parameters are the same as those in Chapter 5, but with γ^t_{i,q_i} computed using the forward-backward algorithm. Initialisation remains the same, with learning based on the template of Table 5.1, with the addition of a forward-backward pass in the source updates.



(a) vbICA-MoG reconstructions



(b) vbICA-HMM reconstructions and state sequences

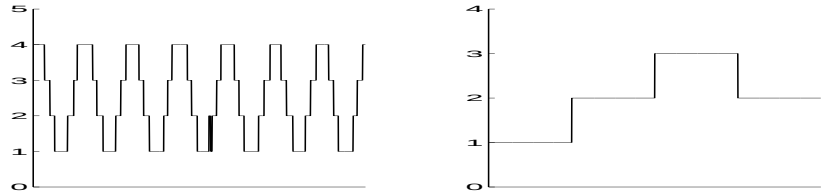


Figure 7.3: Reconstructions by vbICA-MoG and vbICA-HMM.

Results

The vbICA-HMM was compared to vbICA-MoG on two datasets. The first was a simple mixture of a sine wave and noisy step wave, shown in Figure 7.2(a). These were mixed into three signals by a randomly generated mixing matrix and 30% Gaussian noise was added. The start state probabilities and transition matrix for HMM source i are denoted $\boldsymbol{\pi}_i$ and $\boldsymbol{R}$ respectively. Their respective parameters are λ_0 and $\boldsymbol{\iota}$ respectively. λ_0 was set to 1, while ι_{cd} was set to 5 for all $c \neq d$ and to 10 for $c = d$, giving a slight bias in favour of staying in the same state. This is encoding the assumption that any small length of source signal will tend to be generated by the same state, which should improve reconstruction in uncertain (e.g. high noise, scarce data) situations.

(a) Source images

(b) Mixtures: 5% noise

(c) Mixtures: 50% noise

Figure 7.4: Original images and mixtures.

A range of vbICA-MoG and vbICA-HMM models were trained on 1600 data vectors, differing in both the number of sources and the number of Gaussians in the source models. Training ceased when the NFE converged to within 0.001 %. The NFE indicated that both vbICA flavours prefered models with two sources, with vbICA-MoG favouring three components in each MoG and vbICA-HMM choosing four and three components respectively for each source.

Figure 7.3 plots the sources reconstructed by the favoured models. Note that the vbICA-HMM source signals are much cleaner. The average MSE of the vbICA-HMM reconstruction is some 5-6 times smaller than that of vbICA-MoG, while the cross-talk is over three times less. The extra temporal information available to vbICA-HMM has allowed a more faithful reconstruction of the source signals. The source dynamics captured by vbICA-HMM are illustrated by the inferred state paths in Figure 7.3. The cyclical nature of the sign wave is encoded in the learnt state transition matrix. For example, each state has a high probability of staying within itself ($p_{dd} > 0.9$), a small probability of transitioning to the next state along ($p_{d-1,d}, p_{d,d+1} < 0.1$) and non-contiguous

(a) vbICA-MoG: 5%

(b) vbICA-MoG: 50%

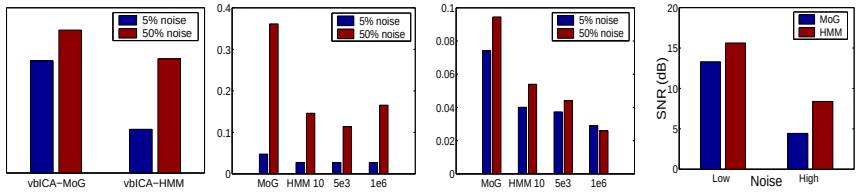
(c) vbICA-HMM: 5%

(d) vbICA-HMM: 50%

Figure 7.5: Reconstructed images.

states have virtually no probability of transitioning between each other. This give rise to the cyclical nature of its state path, with two states representing the peaks and troughs and two states representing the rise and fall between them. The vbICA-HMM model has learnt that this is the underlying pattern of a sine wave, and it is this extra 'understanding' that has facilitated the source reconstructions. Similarly, the state path of source 2 has captured the dynamic changes of state manifest in the second source signal.

To assess the performance of vbICA-HMM on more realistic datasets, the model was used to separate the image mixtures in Figure 2.10 of Chapter 2. Both vbICA-HMM and vbICA-MoG were trained on two sets of mixtures, one with 5% added Gaussian noise and the other with 50%, as shown in Figure 7.4. Each network was given 5 components per source and trained on the whole 16129-vector dataset. The HMM hyper-parameters for vbICA-HMM were set to the same values as before, with a bias towards stayng in the same state. This encodes the assumption that features are localised over a number of pixels. Figure 7.5 shows the image reconstructions for vbICA-MoG and vbICA-HMM.



(a) Free energy (b) MSE (c) Cross-talk (d) Signal-noise ratio

(e) Over-strong prior (f) Removing noise

Figure 7.6: Performance of vbICA-HMM compared with vbICA-MOG.

The benefits reaped from exploiting temporal information is clear: the vbICA-HMM images are clearer than those of vbICA-MoG. Although vbICA-MoG has been able to separate the images (unlike traditional ICA methods), the images are clearly noisier than those recovered by the HMM based source models. This is particularly evident at the higher noise level. Even though vbICA-HMM is more complex, this improved modelling is reflected in the higher NFE of vbICA-HMM over vbICA-MoG. The free energy (negative of the NFE) is shown in Figure 7.6(a). Note that the FE of noisy vbICA-HMM is comparable to that of low-noise vbICA-MoG, underlining the improvement. Shown along side are the differences in MSE and cross-talk between vbICA-MoG and vbICA-HMM for different strengths of prior over the diagonal of the transition matrix ($\iota_{dd} = 10, 5000, 10^6$). The MSE of vbICA-HMM is nearly half that of vbICA-MoG in the low noise case, and nearly a third in the high-noise case. Similarly, the cross-talk is some 50% lower. Of the three strengths, the weaker prior has the highest NFE, although the medium model has the lowest MSE and the strongly constrained model has the lowest cross-talk. Note that the MSE of

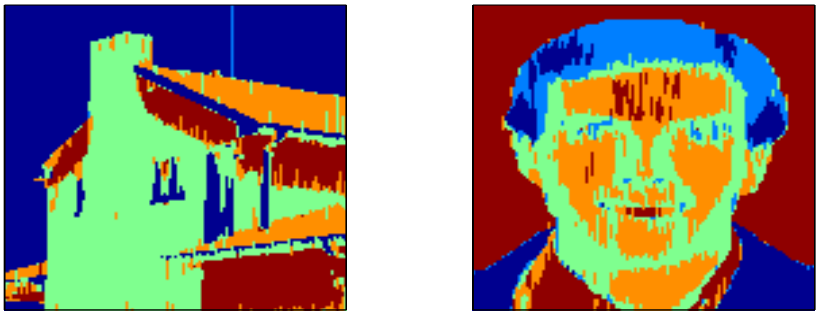


Figure 7.7: Image segmentation by vbICA-HMM.

the strong prior is worse than that of the medium prior. This is a case of the prior being *too* strong. Figure 7.6(c) shows the reconstructed images under this model. As ICA finds one-dimensional components, the images are presented to the models in a vectorised form. The overly-strong prior causes the HMM source models to stay in particular states longer than they should, giving rise to the vertical streaking. Care must be taken in setting priors for specific purposes, for example for finding localised features.

The vbICA-HMM model is more robust under noise than vbICA-MoG, which is confirmed by the increased average signal-to-noise ratio of the source signals (Figure 7.6(d)) under vbICA-HMM. This benefit can be harnessed to denoise data. Figure 7.6(d) shows an image of 'house' with 50% added Gaussian noise. This image was duplicated and fed to the vbICA-HMM model learnt on the 5% noise image mixtures to infer the source signals and noise variance. Of the two sources, one applied to 'bloke' so could be discounted. The other was a denoised version of the noisy house, shown in Figure 7.6(d). The learnt HMM source model used the temporal structure in the noisy image to infer the values of corrupted pixels, improving the SNR from 3.70dB to 7.23dB.

The inferred state path of an HMM is a classification/segmentation of the data presented (in this case, a source signal). The image segmentation carried out under the weak prior is shown in Figure 7.7. It is clear that each of the 5 HMM Gaussian generators corresponds to a brightness level, where - for example - the blue in 'House' representing the bright sky and white wood on the house,

and the orange in 'Bloke' are the highlights on his skin. The colours do not correlate in the two images as the inferred state labels have an arbitrary ordering. Nontheless, the simultaneous (blind) image separation and segmentation carried out by vbICA-HMM is very impressive.

7.4.2 HMM with ICA Generators

The observation model updates (7.62)-(7.67) are for univariate Gaussian generators. In practice, a wide range of generators can be used [129]. Hidden Markov Models with (noiseless) ICA generators were first formulated by Penny *et al.* [129] using reciprocal-cosh source densities, and learnt via maximum likelihood. This was extended by Penny *et al* in [124] by utilising dynamic sources based on Generalised Autoregressive models. This section will model multivariate, non-Gaussian data by replacing the univariate Gaussian observation model of section 7.3 with ICA generators based on the vbICA model. This leads to a dynamic extension of the vbMoICA model of Chapter 6 - a variational Bayesian Hidden Markov Model with ICA generators (vbHMM-ICA).

If the observations are M-dimensional and non-Gaussian distributed, then an appropriate observation model for HMM analysis is given by

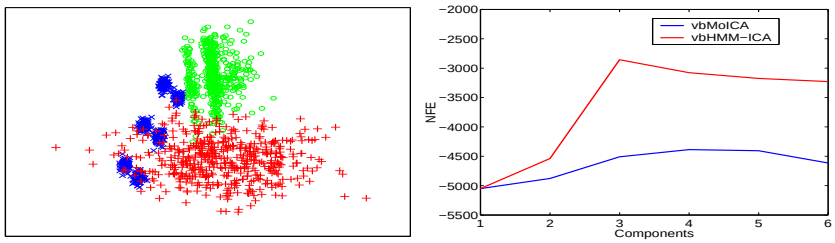
$$p_c(\boldsymbol{x}^t|\boldsymbol{\Theta}_c) \quad = \quad \left(\frac{\Lambda_c}{2\pi}\right)^{\frac{M}{2}} \exp\left[-\frac{\Lambda_c}{2}\left(\boldsymbol{x}^t - \boldsymbol{A}_c\boldsymbol{S}_c - \boldsymbol{y}_c\right)^2\right] \tag{7.68}$$

where Λ_c is the noise precision for generator $i = \{1 \dots C\}$. In this model, $\hat{\eta}_c^t$ is identified with γ_c^t, $\boldsymbol{\kappa}$ with $\boldsymbol{\pi}$ and the ICA model parameters $\Theta = \{\boldsymbol{A}, \boldsymbol{\Lambda}, \boldsymbol{\theta}\}$ replace the Gaussian parameters $\boldsymbol{\mu}$ and $\boldsymbol{\beta}$ of section 7.3.

The priors (7.49) and (7.50) remain the same, while (7.51) and (7.52) are replaced by the vbICA priors of section 5.3.1 (including ARD if required). It is then straight forward to show that (7.53)-(7.55) remain the same with (7.62) replaced with

$$\tilde{p}_c \quad = \quad \tilde{\Lambda}_c^{\frac{M}{2}} \exp\left[-\frac{\langle\Lambda_c\rangle}{2}\langle\left(\boldsymbol{x}^t - \boldsymbol{A}_c\boldsymbol{S}_c - \boldsymbol{y}_c\right)^2|c\rangle\right] \tag{7.69}$$

and (7.64)-(7.67) replaced by the relevant ICA parameter updates in section 6.3.3. Initialisation is identical to vbMoICA (see section 6.3.4), and training



(a) Data (b) NFE of models trained

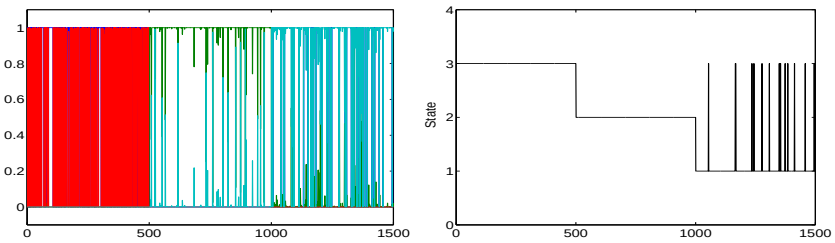
Figure 7.8: Data and NFE of models trained.

follows Table 6.1, albeit with the component indicator probabilities, $\hat{\eta}_c^t$, computed using the forward-backward algorithm.

Results

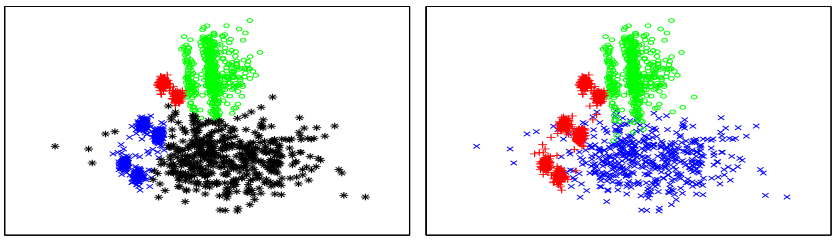
The vbHMM-ICA algorithm was used to partition three 2D non-Gaussian clusters that were partially overlapping. A plot of the data is shown in Figure 7.8(a). The data-set consisted on 1500 data vectors, 500 from each cluster. To provide some temporal information, the first 500 were from the blue cluster, the second 500 from the green and the final 500 from the red. A number of vbMoICA and vbHMM-ICA models were trained covering 1-6 components. The HMM hyper-parameters of vbHMM-ICA were set as with vbMoICA in Chapter 6. Training continued until the NFE converged to within 0.001 %.

The NFE of the learnt models is shown in Figure 7.8(b). The vbHMM-ICA networks are preferred over the vbMoICA networks for all model orders, even though their dynamic models make them more complex. A 4 component model is preferred by vbMoICA, while vbHMM-ICA correctly infers 3 components. The difficulty vbMoICA had in separating the overlapping clusters is evident in the ambiguous posterior class probabilities assigned to each point, as shown in Figure 7.9(a). If the maximum probabilities are chosen, then vbMoICA classifies 244 points out of 1500 (16.27%) incorrectly, 183 of which are allocated to a unnecessary 4^{th} cluster (Figure 7.9(c)). The state path of the preferred vbHMM-ICA model is shown in Figure 7.9(b) and represents the class labels



(a) vbMoICA class probabilities

(b) vbHMM-ICA state path



(c) vbMoICA classification

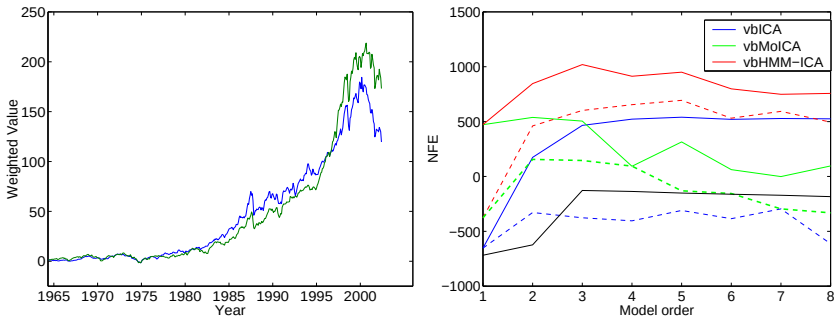
(d) vbHMM-ICA classification

Figure 7.9: Classification of non-Gaussian clusters.

assigned to each point. The associated data partition is shown in Figure 7.9(d). Of the 1500 points, only 20 (1.33%) are misclassified. If the prior over the transition matrix diagonals is increased to 1000, then this drops to just 3 points.

7.5 Real Data - Hierarchical Dynamics

The dynamic ICA algorithms developed in this Chapter were used to model the dynamics of the FTSE-100 and Dow Jones Industrial Average-30 (DJIA30) over the last 38 years. Figure 7.10(a) plots their course from June 1964 to June 2002 at monthly intervals, giving 457 data-vectors in total. These were normalised to unit variance and adjusted to intially start at the weight value of 1 for 6/1964. This data was then fed to a whole gamut of vbICA models, each trained until the NFE converged to within 0.001%. The models used were vbICA, vbMoICA and vbHMM-ICA each with either MoG or HMM sources, giving 6 flavours in total. The 2-source vbICA models were trained across 1-



(a) Share indices. Blue: FTSE100, Green: DJIA30.

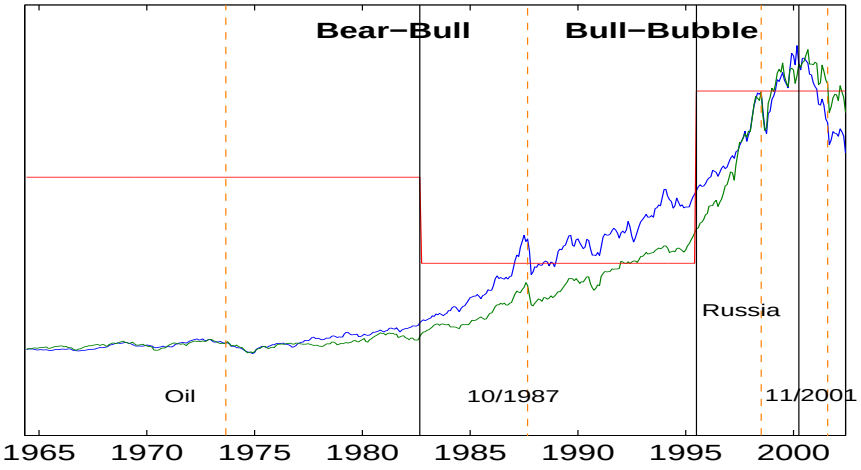
(b) Model selection. Dashed: MoG sources, Solid: HMM sources, Black: Gaussian vbHMM.

Figure 7.10: Original data and NFE of various models.

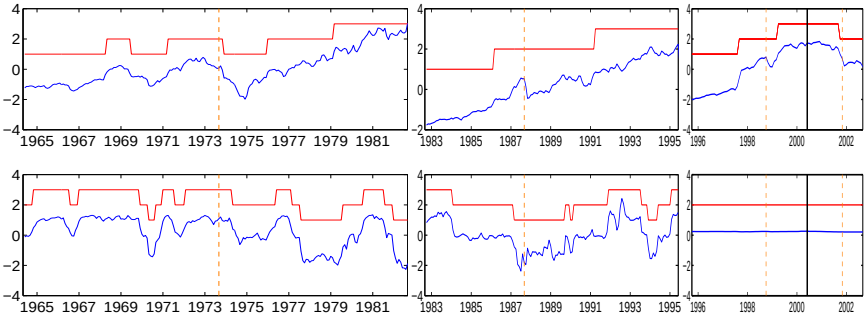
8 components per source (same number per source), and the vbMoICA and vbHMM models were trained across 1-8 component ICA generators, each 2-component ICA with 3 source states each. As a comparison, a vbHMM model with 1-8 Gaussian generators was also trained.

The negative free energy plots for each model is shown in Figure 7.10(b). Across the board, the dynamic models outperformed the stationary ones. The simplest dynamic model, the vbHMM-Gaussian network, performed worse than the dynamically-challenged vbMoICA model, indicating the data distribution is non-Gaussian. In fact, by far the most likely formalism was a vbHMM model with ICA generators, each of which had dynamic HMM source models (vbHMM^{2}ICA). This hierarchical approach scored higher than vbMoICA-HMM, vbICA-HMM and standard vbHMM-ICA, with an HMM with 3 ICA-HMM generators being the most likely.

Exactly what this model learnt is shown in Figure 7.11. To give some meaning to these results, important changes and impacting events in the markets are indicated in Figure 7.11(a). The markets can be divided into 4 regimes, the boundaries of which are marked by solid, black lines. The last sustained, deflationary period of the markets (termed a 'bear market') officially ended on August 12^{th} 1982 [135] and was followed by a sustained, inflationary periord



(a) Regimes and events. Black: Market regimes, Orange: Events, Red: State-path.



(b) ICA2 Recon.

(c) ICA1 Recon.

(d) ICA3 Recon.

Figure 7.11: Patterns in share indices.

(termed a 'bull market'); this is represented by the first boundary. Shares in the web-browser company 'Netscape' were floated on the 9^{th} of August 1995, almost 13 years to the day later. This is generally regarded as the beginning of what is popularly called the 'internet [speculative] bubble' [135]; this is the second black line. This bubble burst on April 14^{th} 2000 [135], forcing the market - yet again - into a bear market; this is the final black boundary. Also scribed in dotted orange are pertinent dates that had marked affects on the markets - the october 1973 raising of oil prices by OPEC, the stock market crash on October

19^{th} 1987 ('Black Monday'), the devaluation of the Russian rouble on the 17^{th} of August 1998, and the September 11^{th} 2001 terrorist attacks on the USA.

Plotted in red on Figure 7.11(a) is the inferred state-path of the vbHMM^{2}ICA model's 'macro-states'. The 2 changes of state correspond exactly to the dates the markets moved from bear to bull status, and from bull to bubble status. Thus, ICA-HMM generator 2 is responsible for pre-bull market dates, ICA-HMM 1 for the bull market and ICA-HMM 3 for the bubble and subsequent bear market. What vbHMM^{2}ICA has captured are fundamental changes in the markets which manifest themselves as fundamental changes in the *statistics* of the markets. The markets are chaotic, but structured and increase in the long term, so these statistics are non-Gaussian. Although the standard vbHMM-Gaussian model also favoured three states, the associated state-path did not correspond to any recognised changes in the markets' regime. Gaussian generators are not well matched to detecting changes in non-Gaussian statistics.

Plotted below Figure 7.11(a) are the ICA-HMM source reconstructions together with their state-paths of 'micro-states'. The signals for source 1 represent a 'proto-index' which is the underlying signal common to both the FTSE100 and the DJIA30. The state-paths for these signals essentially track the mean of this signal in a piecewise fashion. On two occasions, the changes of state match with specific stimuli, namely the 1973 oil crisis and the September 11^{th} attacks. The former led to a subsequent bear market, while the later greatly worsened an already weak market, deepening the bear market substantially. These mark changes in the underlying statistics. The 1987 crash and russian currency crisis do not register in the state dynamics, both of which had only short-term impacts on an overall bear/bubble market.

The meaning of the second source is less clear. Whereas the overall vbHMM model and the first source of the constituent ICA-HMM models track the changing mean, the second seems to bear little relation to the stock indices. It is tempting to dismiss these signals as non-Gaussian noise, but - for ICA's 1 and 2 at least - there does seem to be some superficial meaning. While source signal

1 tracks the common, underlying proto-index, signal two seems to track the *difference* between the two. This is further supported by entries in the mixing matrices. The appropriate entries for source-to-sensor mapping for source 2 are roughly equal in magnitude for each sensor signal, but with different signs. This would indicate that source 2 quantifies the difference between the two such that

$$\text{FTSE100} \approx A_{11}\text{proto} + \alpha\text{diff} \tag{7.70}$$

$$\text{DJIA30} \approx A_{21}\text{proto} - \alpha\text{diff} \tag{7.71}$$

where $\alpha = \frac{1}{2}(|A_{12}| + |A_{22}|)$. This does not seem to be the case for ICA 3, where the second signal has been suppressed. The reason for this is unclear.

Note that the final regime change is picked up by a change in ICA-HMM state rather than a change of state in the overall vbHMM, as the previous two were. There are only 26 data vectors after the bursting of the bubble, which is probably not enough to warrant a vbHMM state of their own - it is cheaper to code them as a state change within an ICA-HMM generator. If there were enough data vectors, it is hard to tell whether the continuing bear market's statistics are simply a mirror of the bull market's, or are unique to a bear market. In the former case, such a regime would be represented by the same state as the bull market with the ICA source signals coding for the downward trend. In the latter, there would be an extra 4^{th} state solely determined by bear market statistics. The HMMs seem to capture mainly changes in amplitude, so it is likely that a prolonged bear market would register as a negative bull market.

7.6 Discussion

This Chapter has shown how the explicit use of temporal information in data can be used to improve the decompositional and representational power of vbICA and its siblings. This information is exploited by utilising Hidden Markov Models to capture high-order dynamics in the statistics of the data. ICA with HMM sources (vbICA-HMM) is more robust at finding independent components in noisy data as demonstrated by the image separation of very noisy image mix-

tures. An added benefit is the simultaneous segmentation of these images into areas of similar grey-scale. Mixtures of ICA with Markov priors are equivalent to an HMM with ICA generators (vbHMM-ICA). This used dynamics in the data to separate overlapping multidimensional, non-Gaussian clusters much more accurately than a (static) vbMoICA model. The combination of vbHMM-ICA with vbICA-HMM was then used to find patterns in the FTSE100 and DJIA30 stock indices. The hierachical dynamics and patterns found were shown to tally with recognised changes in market regime and specific historical events.

7.6.1 Problems

The same problems that afflict vbICA and vbMoICA impinge on the greater applicability of the dynamic ICA models developed here, namely correlated data, speed, and highly overlapping clusters. In the case of vbICA-HMM, the correlated problem can be overcome by decorrelating the data as shown in section 5.7.1 - this does not affect the dynamics. Speed takes a further hit with the addition of a Forward-Backward step which takes a time proprtional in T, the length of the dataset. This can be overcome using a programming language more sophisticated than MATLAB. The problem of highly overlapping clusters in the data density is the real problem as this especially limits the use of these models in non-stationary BSS. A set of dynamic signals which are being dynamically mixed may still overlap in the data space if they are co-mean and the mean remains static. How to overcome this is, as yet, an open question.

7.6.2 Further Applications and Extensions

The most obvious application of vbICA-HMM is in the BSS of time-series data. Music signals, for example, have temporal structure in a similar way to images, so BSS would be more robust using vbICA-HMM, exploiting temporal information to obtain cleaner reconstructions. This would be particularly applicable to speech recognition. HMMs are the current cutting edge method for recognising words and syllables. Although these work exceptionally well in lab conditions, the real world is noisy. When speaking into a (stereo) microphone, the word

being spoken and the rubbish in the backround are resolutely independent. ICA with 2 sources, one HMM and the other MoG, could be used to blindly siphon off background noise, giving a cleaner signal to the HMM thereby improving performance in real situations. If this HMM is trained ahead of time then installed in the ICA network, the improvement would be even more accute.

The models could be extended in a number of ways. The dynamics of the HMM part could be made more sophisticated. The HMMs presented in this Chapter have been a 1st-order Markov process, whereby the statistics at time t depend on the hidden state at time $t-1$. This could be generalised to $t-\tau$. The independent components could be made dependent by coupling the hidden states using coupled HMMs [136]. This is an important method in fusing information extracted from disparate data, for example combining auditory speech and visual lip information to make speech recognition more reliable. The same could be done to vbHMM-ICA to model multivariate data with hidden dependencies. Non-Gaussian observation densities for could be utilised in the source HMMs to perform specific functions, for example illumination/object segmentation in images[137].

Chapter 8

Close

And thus the journey ends. This closing Chapter provides a summary, discusses outstanding problems and suggests future extensions before a final conclusion.

8.1 Summary

The thesis was split into 8 Chapters. The first introduced the fundamental concepts of pattern recognition, while this Chapter concludes the thesis. Chapters 2 - 4 explained the theory, while Chapters 5 - 7 derived the proposed ICA methods.

Independent Component Analysis is the fundamental area of research in this thesis, and was introduced in Chapter 2. The Chapter explained that the repackaging of informative signals into independent components, or sources, stemmed from the desire for an intrinsic coordinate frame in which the data was at its most informative, allowing structure within the data distribution - and, therefore, patterns in the data - to become more apparent. The notion of independence was also shown to be of importance from a neurological point of view. Two popular theories on neocortical coding were discussed, one based on the reduction of redundancy across imput sensory signals, and the other on a sparse coding schema for these signals; both were shown to naturally lead to independent output signals. Applications for ICA were briefly discussed. The ICA problem was then formulated as the search for a mathematical mapping, or - equivalently - as a problem in modelling the data generation process. Various

quantitative measures of independence were also introduced. This was followed by a brief history of ICA research in both the neurologically-motivated mapping community, and the distribution-orientated modelling community. Both of these strands of research can be understood from a probabilistic generative modelling stand-point, so this paradigm was chosen as the basis for the ICA research presented in this thesis. A basic generative model for ICA was then formally derived and demonstrated by separating mixed music signals. The Chapter concluded with a discussion of the limitations of this basic model, in particular the absence of a noise model, the unflexible source model, and the inability to infer the most appropriate number of components.

The ability to combine simple models to produce more powerful ones, compare models and model assumptions, and incorporate prior knowledge and constraints into these models is the central contribution this thesis makes to ICA research. The framework behind this is introduced in Chapter 3. This Chapter delved deeper into the world of modelling by introducing the Bayesian framework. Bayes' theorem and its use in inferring quantities of interest was introduced. After a brief detour into the graphical representations of models, the Chapter explained how Bayesian inference could be used to build and learn a parametric generative model, and how information could be subsequently elicited from it. The Chapter also explained how Bayes' theorem could be used to compare candidate models and incorporate prior knowledge and constraints into these models. Bayesian computations are intractable, however, for all but the simplest models. Various methods for approximating Bayesian inference were discussed, but all were eventually dismissed as none encompassed all the benefits that the Bayesian framework enjoys.

The existence of this thesis rests on a recent development in approximating Bayesian inference - the variational approximation. Chapter 4 derived this method from the intuitive angle of fitting approximating distributions to the true, but intractable posterior. The Chapter then explained how the variational method could be used to approximate Bayesian inference, and detailed

the variational Bayesian methodology for learning and inference in models. The Chapter concluded with a demonstration of variational Bayesian learning for a mixture of Gaussian model. The demonstration confirmed that important aspects of the Bayesian framework, such as model comparison and incorporation of prior knowledge, remained intact under this approximation.

Chapters 5-7 described the new ICA methods developed. Chapter 5 derived the core variational Bayesian ICA algorithms. After a brief overview of the current state of the art in ICA research, the proposed variational Bayesian ICA model was introduced. This used the MoG presented at the end of Chapter 4 as its source model, allowing complex multi-modal independent components, and incorporated an explicit noise model. Two sets of learning algorithms were derived based on different assumptions about the model's parameter posterior probability density. The algorithms were tested on test data, and both were shown to recover a range of source distributions with multi-modal densities. The Bayesian methodology's ability to compare models was highlighted by correctly inferring the number of sources underlying various observations. Both algorithms were shown to comprehensively outperform popular ICA algorithms at separating noisy mixtures of images. The two different variational algorithms were then compared. While the first was shown to be quicker, the second was found to be more robust and more readily extensible. The ability to encode prior knowledge and constraints was explored by incorporating a method for 'killing' unnecessary sources (Automatic Relevance Determination), and enforcing positivity constraints for non-negative data. The vbICA model was then used to remove artifactes from ECG signals. The Chapter concluded with a discussion of the merits and limitations of vbICA.

ICA assumes that the whole data distribution is described by one coordinate frame. This may not always be the case, particularly if the data distribution exhibits localised, self-similar manifolds, or clusters. In such a case, a mixture of ICAs can be used, with each ICA responsible for finding a local coordinate frame that best describes the cluster under its jurisdiction. Chapter 6 bolted

together multiple vbICA models to produce a variational Bayesian Mixture of Independent Component Analysers. This was tested on 2D and 3D test data consisting of complex, multi-modal, non-Gaussian clusters. The vbMoICA algorithm was found to produce more faithful representations than previous methods, automatically determine the local dimensionality of each cluster correctly, and infer the correct number of clusters in the data distribution. The vbMoICA algorithm was then used to decompose functional Magnetic Resonance Imaging brain scans into localised and interpretable features. Possible applications were discussed, together with some possible future extensions.

Although ICA is used widely in analysing time series, the vast majority of ICA algorithms ignore temporal information. The vbICA and vbMoICA models are no different. This was rectified in Chapter 7 by incorporating Hidden Markov Models into vbICA and vbMoICA. This led to vbICA with dynamic HMM sources (vbICA-HMM), and vbHMM with ICA components (vbHMM-ICA) respectively. The vbICA-HMM model was shown to separate out noisy mixtures of dynamic source signals better than the standard vbICA-MoG model. In particular, vbICA-HMM unmixed noisy mixtures of images much better than vbICA-MoG and simultaneously segmented them. Similarly, vbHMM-ICA was shown to separate overlapping, non-Gaussian clusters better than vbMoICA, if the data was temporally correlated. An HMM with ICA components, each with dynamic sources (vbHMM^2-ICA) was used to find hierarchical dynamics and interpretable patterns in share indices.

8.2 Problems to Overcome

Although the versatility and robustness of the vbICA-based algorithms have been comprehensively demonstrated, there are practical issues to address.

8.2.1 Initialisation

As discussed in section 6.3.4, the choice of initialisation can have an effect on the final solution. SVD intialisation always starts at the same place for a given

dataset so always ends at the same solution. In practice, even different random intialisations generally lead to the same solution as well. However, if the data density is highly correlated (or particular manifolds in the case of the vbMoICA and its HMM derivatives), the ICA model will stay close to the PCA solution if initialised using SVD. Different random initialisations may lead to different solutions, or may converge onto the PCA/FA solution.

Decorrelation

One possible solution is to decorrelate or 'whiten' the data density, then commence learning on these whitened data. This strategy is employed by a number of ICA algorithms (for example, FastICA) as it can speed up convergence under appropriate conditions (i.e. if the algorithm is derived assuming whitened inputs). In the case of ICA mixture models, this is not possible as one cannot identify the different clusters before learning. One strategy is to whiten each cluster identified by the k-means/GMM initialisation, but this is rather an *ad hoc* method and will only work if the clusters are well separated.

Correlated posterior

Another option is to drop the factorial approximation in (6.13) as it is no longer a good approximation for highly correlated data. Dropping this will solve much of the problem, albeit with a corresponding reduction in speed. As speed is another major issue to tackle, this may not be possible. More investigation is certainly needed into intialisation strategies to overcome the correlation problem.

ICA initialisation of vbICA

One possible improvement in initialisation is to perform ICA using some quick established method which is better at dealing with correlated data, then use its results to intialise vbICA. This feels like cheating, however.

8.2.2 Speed

Speed is the other problem. Bayesian learning is inherently slower than non-Bayesian methods due to the extra parameters that have to be estimated. In fact, Bayesian inference can be shown to be NP-hard [96]. The variational approximation brings tractability to the Bayesian formulation, and is speedier than the non-deterministic sampling regime, but it still remains understandably slower than its non-Bayesian counterparts. Where traditional ICA algorithms take seconds, minutes and hours, vbICA can take minutes, hours and even days depending on the dimensionality and size of the dataset. This severely limits its range of application, particularly to high-dimensional data.

The central problem lies in the need to cyclically update the coupled mean-field equations. This constant cycling over equations, sub-routines - and models in the case of vbMoICA etc. - is cumbersome. Alternative optimisation strategies need to be researched to see if this constant cycling can be short-cut. For example, Everson and Roberts [35] note that an unmixing matrix must also be a decorrelating one, so constraining learning of the unmixing matrix to the manifold of decorrelating matrices will greatly speed up convergence. Interesting work by Valpola and Pajunen [138] on speeding up Bayesian ICA may also provide a basis for a faster vbICA algorithm. The optimisation of algorithms is a notoriously Sisyphean task, however, and may require a whole further Doctorate worth of work.

8.2.3 Active model selection

Model-order selection for the number of ICA sources, source MoG components, ICA components and HMM generators can become prohibitive if a large number are required. Some progress may be made in this area if, for example, some sort of birth-death [139] split-merge [140] criteria could be enforced allowing active generating, culling, splitting and merging of components during the learning process.

8.3 Future Directions

The models developed in this thesis can be extended in a number of ways.

8.3.1 Further source models

The source models installed so far are Mixture of Gaussians, Mixture of Positives and Hidden Markov Model versions of these. Other source models worth consideration are

- Autoregressive sources to incorporate first-order dynamics - s_i^t conditioned on s_i^{t-1} rather than q_i^t.
- Binary sources to model binary data. Useful for binary coding, compression and latent variable model error correcting.
- Wavelet sources for modelling temporally dependent data.

These are just three possible source models. The extensible nature of probabilistic modelling allows any probabilistic model to be bolted on to any other, allowing a vast array of possible source models.

8.3.2 Fully dynamic models

The dynamic ICA models explored in Chapter 7 use HMMs to learn dynamics. Although these are capable of capturing higher-order temporal information, they can only deal with varying, non-instantaneous mixing and varying sources in a piecewise manner. To overcome this, a Markov process could be derived where, for example, the mixing matrix at time t was conditioned on the mixing matrix at time $t-1$ rather than through an intermediary hidden variable. A similar system could be used for the source density statistics. See [128] for more details.

8.3.3 Nonlinear ICA

The mixing so far has been considered linear. This assumption may not be adequate for some datasets. In such cases, nonlinear mixing maybe more appropriate. Nonlinear mappings, however, lead to many indeterminancies which

make ICA intractable. These can be overcome if the nonlinear mapping is limited in some way, as discussed in [81]. These caveats may allow vbICA to be extended to nonlinear-vbICA.

8.3.4 Blind Deconvolution

The mixing so far has been considered instantaneous. In real world situations, this is rarely the case. A more genral model would incorporate delayed mixing which could then be used for deconvolution as well as unmixing. More information can be found in [141].

8.3.5 Overcomplete sources

ICA will only work if there are equal or fewer components than dimensions in the data. Clearly, the brain can separate many signals with just the use of two ears. To achieve this computationally, more information needs to be extracted, such as frequency information.

8.4 Conclusion

This thesis has sought to extend ICA and make it more powerful, flexible and more widely applicable. The Bayesian formulation has allowed prior knowledge and constraints to be incorporated in a principled manner, such as Automatic Relevance Determination and non-negative constraints. Bayesian inference has been used to avoid overfitting and increase performance for noisy data. Bayesian model selection has allowed the most appropriate number of independent components to be inferred, and allowed the quantified comparison of different models competing to explain the same data. Mixtures of ICAs have allowed different parts of the data distribution to be decomposed into independent components most appropriate for local conditions. Dynamics have been incorporated by using HMM sources in ICA models as well as ICA generators in HMM models. The flexibility, power and robustness of variational Bayesian Independent Component Analysis has been shown to exceed that of traditional ICA methods.

Bibliography

[1] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Walton Street, Oxford, 1995.

[2] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill Education, 2001.

[3] R. Cox, "Probability, frequency and reasonable expectation," *American Journal of Physics*, vol. 14, pp. 1–13, 1946.

[4] C. Shannon, "The mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[5] K. Pearson, "On lines and plains of closest fit to systems of points in space," *Philosophy Magazine*, 1901.

[6] I.T. Jolliffe, *Principal Component Analysis*, Springer, 1986.

[7] M.E. Tipping and C.M. Bishop, "Probabilistic principal component analysis," Tech. Rep., Aston University, 1997.

[8] B.S. Everitt, Ed., *An Introduction to Latent Variable Modelling*, Chapman and Hall, 1984.

[9] P.J. Huber, "Projection pursuit," *The Annals of Statistics*, vol. 13, no. 2, pp. 435–475, 1985.

[10] J.H. Friedman, "Exploratory projection pursuit," *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 249–266, March 1987.

[11] M.C. Jones and R. Sibson, "What is projection pursuit?," *Journal of the Royal Statistical Society, Series A*, vol. 150, pp. 1–36, 1987.

[12] M. Abramowitz and C.A. Stegun, Eds., *Handbook of Mathematical Functions and Formulas, Graphs and Mathematical Tables*, Dover, New York, 9 edition, 1972.

[13] M. Girolami and C. Fyfe, "Negentropy and kurtosis as projection pursuit indices provide generalised ICA algorithms," in *Advances in Neural Information Processing Systems*, A. Cichoki and A. Back, Eds., 1996.

[14] P. Comon, "Independent Component Analysis: A new concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.

[15] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for Independent Component Analysis," *Neural Computation*, vol. 9, no. 7, pp. 1483–1492, 1997.

[16] A. Hyvärinen, "Fast ICA by a fixed-point algorithm that maximises non-Gaussianity," in *Independent Component Analysis: Principles and Practice*, S.J. Roberts and R. Everson, Eds., pp. 71–94. Cambridge University Press, 2001.

[17] H.B. Barlow, "Possible principles underlying the transformations of sensory messages," in *Sensory Communication*, W.A. Rosenblith, Ed., pp. 217–234. MIT Press, 1961.

[18] H.B. Barlow, "Unsupervised learning," *Neural Computation*, vol. 1, pp. 295–311, 1989.

[19] H.B. Barlow, "What is the computational goal of the neocortex?," in *Large-scale Neuronal Theories of the Brain*, C. Koch and J.L. Davis, Eds., pp. 1–22. MIT Press, 1994.

[20] J.J. Attick and A.N. Redlich, "Towards a theory of early visual processing," *Neural Computation*, vol. 2, pp. 308–320, 1990.

[21] R. Linsker, "An application of the principle of maximum information transfer to linear systems," in *Advances in Neural Information Systems*, D.S. Touretzky, Ed. 1989, vol. 1, Morhan Kaufmann.

[22] R. Linsker, "Local synaptic learning rules suffice to maximise mutual information in a linear network," *Neural Computation*, vol. 4, pp. 691–702, 1992.

[23] G. Deco and D. Obradovic, "Linear redundancy reduction learning," *Neural Networks*, vol. 8, no. 5, pp. 751–755, 1995.

[24] John Wilkinson, "The engineer's guide to compression," Online booklet, Snell and Wilcox,.

[25] J.J. Attick, "Could information theory provide an ecological theory of sensory perception?," *Network*, vol. 3, pp. 213–251, 1992.

[26] H.B. Barlow, T.P. Kaushal, and G.J. Mitchison, "Finding minimum entropy codes," *Neural Computation*, vol. 1, pp. 412–423, 1989.

[27] D.J. Field, "What is the goal of sensory coding?," *Neural Computation*, vol. 6, pp. 559–601, 1994.

[28] J. Herault and C. Jutten, "Space or time adaptive signal processing by neural models," in *Proceedings AIP Conference: Neural Networks for Computing*, J.S. Denker, Ed. American Institute for Physics, 1986, vol. 151, pp. 206–211.

[29] S. Makeig, A.J. Bell, T-P. Jung, and T.J. Sejnowski, "Independent Component Analysis for electroencephalographic data," in *Advances in Neural Information Processing Systems*, D. Touretzky, M. Mozer, and M. Hasselmo, Eds., 1996, vol. 8, pp. 145–151.

[30] R. Vigario, J. Sarela, V. Jousmaki, and E. Oja, "Independent component approach to the analysis of EEG and MEG recordings," *IEEE Transactions on Biomedical Engineering*, 2000.

[31] A.D. Back and A.S. Weigend, "A first application of Independent Component Analysis to extracting structure from stock returns," *International Journal of Neural Systems*, vol. 8, no. 4, pp. 473–484, 1998.

[32] E. Chin, A.S. Weigend, and H. Zimmerman, "Computing portfolio risk using gaussian mixtures and independent component analysis," in *Proceedings of the 1999 IEEE/IAFE/INFORMS Conference on Computational Intelligence for Financial Engineering (CIFEr'99)*. IAFE, 1999.

[33] K. Torkolla, "Blind separation of audio signals: Are we there yet?," in *Proceedings of the First International Conference on Independent Component Analysis and Blind Source Separation (ICA'99)*, 1999, pp. 239–244.

[34] M.S. Bartlett, H.M. Lades, and T.J. Sejnowski, "Independent components representations for face recognition," in *Proceedings of the SPIE Symposium on Electronic Imaging: Science and Technology: Conference on Human Vision and Electronic Imaging III*, 1998, pp. 528–539.

[35] R Everson and S.J. Roberts, "Independent Component Analysis: A flexible nonlinearity and decorrelating manifold approach," *Neural Computation*, vol. 1, no. 8, 1999.

[36] A.J. Bell and T.J. Sejnowski, "The "independent components" of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–3338, 1997.

[37] B.A. Olshausen and D.J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[38] B.A. Olshausen and D.J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[39] J.H. van Hateren and A. van der Schaaf, "Independent component filters of natural images compared with simple cells in primary visual cortex," *Proceedings of the Royal Society of London, series B*, vol. 265, pp. 359–366, 1998.

[40] B.L. Isbell and P. Viola, "Restructuring sparse high dimensional data for effective retrieval," in *Advances in Neural Information Processing Systems*, M.S. Kearns, S.A. Solla, and D.A. Cohn, Eds., 1999, vol. 11, pp. 480–486.

[41] T. Kolenda, L.K. Hansen, and S. Sigurdsson, "Independent Component Analysis in text," in *Advances in Independent Component Analysis*, M. Girolami, Ed., Perspectives in Neural computing. Springer, 2000.

[42] T. Ristaniemi and J. Joutsensalo, "On the performance of blind source separation in CDMA downlink," in *Proceedings of ICA '99*, 1999, pp. 437–442.

[43] Y. Deville, J. Damour, and N. Charkani, "Improved multi-tag radio-frequency identification systems based on new source separation neural networks," in *Proceedings of the First International Conference on Independent Component Analysis and Blind Source Separation (ICA'99)*, 1999, pp. 449–454.

[44] L. Parra, C. Spence, P. Sajda, A. Ziehe, and K-R. Muller, "Unmixing hyperspectral data," in *Advances in Neural Information Processing Systems*, 2000, vol. 12.

[45] A.J. Bell and T.J. Sejnowski, "An information maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[46] H. Attias, "Independent Factor Analysis," *Neural Computation*, vol. 11, pp. 803–851, 1999.

[47] R.M. Everson and S.J. Roberts, "Measuring mutual information," Tech. Rep., University of Exeter, 2000.

[48] M. Basseville, "Distance measures for signal processing and pattern recognition," *Signal Processing*, vol. 18, no. 4, pp. 349–369, December 1989.

[49] J-P. Nadal and N. Parga, "Non-linear neurons in the low noise limit: A factorial code maximises information transfer," *Network*, vol. 5, pp. 565–581, 1994.

[50] T-W. Lee and T. Sejnowski, "Independent Component Analysis for sub-Gaussian and super-Gaussian mixtures," in *4th Hoint Symposium on Neural Computation*. Institute for Neural Computation, 1997, vol. 7, pp. 132–140.

[51] O. Kellenberg, *Foundations of Modern Probability*, Springer-Verlag, New York, 1997.

[52] A. Hyvärinen, "New approximations of differential entropy for Independent Component Analysis and projection pursuit," in *Advances in Neural Information Processing Systems*. 1998, vol. 10, pp. 273–279, MIT Press.

[53] S.M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, 1993.

[54] D.O. Hebb, *The Organization of Behaviour*, Wiley, New York, 1949.

[55] C. Jutten and J. Herault, "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, pp. 1–10, 1991.

[56] P. Comon, C. Jutten, and J. Herault, "Blind separation of sources, Part II: Problem statement," *Signal Processing*, vol. 24, pp. 11–20, 1991.

[57] M.G. Kendal and A. Stuart, *The Advanced Theory of Statistics*, Charles Griffin, 1969.

[58] S-I. Amari, A. Cichocki, and H. Yang, "A new learning algorithm for blind source separation," in *Advances in Neural Information Processing Systems*, D. Touretzky, M. Mozer, and M. Hasselmo, Eds., 1996, vol. 8, pp. 757–763.

[59] Laheld B. Cardoso, J-F., "Equivariant adaptive source separation," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 362–370, 1996.

[60] J-F. Cardoso, "High-order contrasts for Independent Component Analysis," *Neural Computation*, vol. 11, pp. 157–192, 1999.

[61] J. Karhunen and J. Joutensalo, "Representation and separation of signals using nonlinear PCA type learning," *Neural Networks*, vol. 7, no. 1, pp. 113–127, 1994.

[62] E Oja, "The nonlinear PCA learning rule in Independent Component Analysis," *Neurocomputing*, vol. 17, pp. 25–45, 1997.

[63] Z. Roth and Y. Baram, "Multidimensional density shaping by sigmoids," *IEEE Transactions on Neural Networks*, vol. 7, no. 5, pp. 1291–1298, 1996.

[64] M.A. Girolami, "An alternative perspective on adaptive Independent Component Analysis algorithms," *Neural Computation*, vol. 10, no. 8, pp. 2103–2114, 1998.

[65] T.W. Lee, M. Girolami, and T.J. Sejnowski, "Independent Component Analysis using an extendend infomax algorithm for mixed sub-Gaussian and super-Gaussian sources," *Neural Computation*, vol. 11, no. 2, pp. 409–433, 1999.

[66] M. Gaeta and J-L. Lacoume, "Source separation without prior knowledge: The maximum likelihood approach," in *Proceedings of Eusipo*, 1990, pp. 621–624.

[67] D-T. Pham, P. Garrat, and C. Jutten, "Separation of a mixture of independent sources through a maximum likelihood approach," in *Proceedings of EUSIPO*, 1992, pp. 771–774.

[68] D.J.C. Mackay, "Maximum likelihood and covariant algorithms for Independent Component Analysis," Tech. Rep., University of Cambridge, 1996.

[69] B. Pearlmutter and L. Parra, "A context-sensitive generalization of ICA," in *ICONIP '96*, 1996, pp. 151–157.

[70] J-F. Cardoso, "Infomax and maximum likelihood for blind source separation," *IEEE Letters on Signal Processing*, vol. 4, pp. 112–114, 1997.

[71] A.P. Dempster, N.M. Laird, and Rubin D.B., "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, pp. 1–38, 1977.

[72] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul, "An introduction to variational methods for graphical models," in *Learning in Graphical Models*, M.I. Jordan, Ed. The MIT Press, Cambridge, Massachusetts,USA, 1999.

[73] D.J.C. MacKay, *Bayesian Methods for Adaptive Models*, Ph.D. thesis, California Institute of Technology, 1991.

[74] S.J. Roberts, "Independent component analysis: Source assessment and separation, a Bayesian approach," *IEE Proceedings on Vision, Image, and Signal Processing*, vol. 145, no. 3, pp. 149–154, 1998.

[75] K.H. Knuth, "Bayesian source separation and localisation," in *Proceedings of SPIE'98*, A. Mohammad-Djafari, Ed., 1998, vol. 3459, pp. 147–158.

[76] H Lappalainen, "Ensemble learning for Independent Component Analysis," in *Proceedings of the First International Workshop on Independent Component Analyis*, 1999, pp. 7–12.

[77] N.D. Lawrence and C.M. Bishop, "Variational Bayesian Independent Component Analysis," Tech. Rep., Computer Laboratory, University of Cambridge, 2000.

[78] J.W. Miskin and D.J.C. MacKay, "Application of ensemble learning to infra-red imaging," in *Proceedings of ICA2000*, 2000.

[79] T.W. Lee, M. Girolami, A.J. Bell, and T.J. Sejnowski, "A unifying information-theoretic framework for Independent Component Analysis," *International Journal of Mathematical and Computer Modelling*, 1998.

[80] S.J. Roberts and R. Everson, Eds., *Independent Component Analysis: Principles and Practice*, Cambridge University Press, 2001.

[81] J. Karhunen, "Nonlinear Independent Component Analysis," in *Independent Component Analysis: Principles and Practice*, S.J. Roberts and R. Everson, Eds., chapter 4, pp. 113–134. Cambridge University Press, 2001.

[82] M. Lewicki and B. Olshausen, "Inferring sparse, overcomplete image codes using an efficient coding frameowork," in *Advances in Neural Information Processing Systems*, vol. 10.

[83] A. Hyvärinen, R. Cristescu, and E. Oja, "A fast algorithm for estimating overcomplete ICA bases for image windows," in *Proceedings of the International Joint Conference on Neural Networks*, Washington D.C., 1999.

[84] S. Roweis and Z. Ghahramani, "A unifying review of linear Gaussian models," *Neural Computation*, vol. 2, no. 2, pp. 305–345, 1999.

[85] J-F. Cardos, "On the stability of source separation algorithms," in *Neural Networks for Signal Processing VIII*, A. Constantinides, King S-Y., M. Niranjan, and E. Wilson, Eds., 1998, pp. 13–22.

[86] T. Bayes, "An essay towards solving a problem in the doctrine of chances," *Philosophical Transactions of the Royal Society*, vol. 53, pp. 370–418, 1763, Available from JSTOR, http://www.jstor.ac.uk.

[87] S.M. Stigler, "Laplace's 1774 memoir on inverse probability," *Statistical Science*, vol. 1, no. 3, pp. 359–378, 1986.

[88] B. Ripley, *Spatial Statistics*, Wiley, New York, U.S.A., 1981.

[89] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian relation of images," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[90] J. Besag, J. York, and A. Mollie, "Bayesian image restoration with two applications in spatial statistics," *Ann. Inst. Statist. Math.*, vol. 43, no. 1, pp. 1–59, 1991.

[91] J. Hertz, A. Krogh, and R. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1991.

[92] W.L. Buntine, "Operations of learning with graphical models," *Journal of Artificial Intelligence Research*, vol. 2, pp. 159–225, 1994.

[93] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Mateo, CA, 2 edition, 1988.

[94] M.I. Jordan, Ed., *Learning in Graphical Models*, The MIT Press, Cambridge, Massachusetts,USA, 1999.

[95] E.T. Jaynes, *Probability Theory: The Logic of Science*, http://bayes.wustl.edu/etj/prob.html, 1995.

[96] G.F. Cooper, "Computational complexity of probabilistic inference using Bayesian belief networks," *Artificial Intelligence*, vol. 42, no. 3, pp. 393–405, 1990.

[97] J. Rissanen, "MDL modeling - an introduction," in *From Statistical Physics to Statistical Inference and Back*, P. Grassberger and J-P. Nadal, Eds., pp. 95–104. Kluwer Academic, 1994.

[98] W.D. Penny and S.J. Roberts, "Variational Bayes for 1-dimensional mixture models," Tech. Rep. PARG-00-2, Engineering Science, University of Oxford, March 2000.

[99] A.M. Walker, "On the asymptotic behaviour of posterior distributions," *Journal of the Royal Statistical Society B*, vol. 31, pp. 80–88, 1967.

[100] D.J.C. Mackay, "An introduction to Monte Carlo methods," in *Proceedings of the 1996 Erice Summer School*, http://www.cs.toronto.edu/ mackay/abstracts/erice.html, 1996.

[101] N. Metropolis, A.W. Rosenbluth, M.N. Roenbluth, A.H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[102] G. Schwarz, "Estimating the dimensionality of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[103] D.J.C. MacKay, "Information theory, probability and neural networks draft 1.2.4," http://wol.ra.phy.cam.ac.uk/mackay/, 1997.

[104] G Stephenson, *Mathematical Methods for Science Students*, Longman Scientific and Technical, 1992.

[105] H.B. Callen, *Thermodynamics and an Introduction to Thermostatistics*, John Wiley and Sons Inc., New York, 2 edition, 1960 (revised 1985).

[106] C. Peterson and J.R. Anderson, "A mean field theory learning algorithm for neural networks," *Complex Systems*, vol. 1, pp. 995–1019, 1987.

[107] G.E. Hinton and D. van Camp, "Keeping neural networks simple by minimising the description length of the weights," in *Proceedings of COLT-93*, 1993.

[108] D.J.C. MacKay, "Developments in probabilistic modelling with neural networks - ensemble learning," in *Proceedings of the third Annual Symposium on Neural Networks*, Nijmegen, The Netherlands, 1995, pp. 191–198, Springer.

[109] H. Attias, "Learning parameters and structure of latent variable models by variational Bayes," in *Electronic Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-1999)*, http://www2.sis.pitt.edu/~dsl/UAI/uai.html, 1999, Association for Uncertainty in Artificial Intelligence (AUAI).

[110] T.S. Jaakkola and M.I. Jordan, "Bayesian parameter estimation via variational methods," *Statistics and Computing*, vol. 10, pp. 25–37, 2000.

[111] Z. Ghahramani and M. Beal, "Variational inference for Bayesian mixtures of factor analysers," in *Advances in Neural Information Processing Systems*, 2000, vol. 12, pp. 449–455.

[112] J.W. Miskin, *Ensemble Learning for Independent Component Analysis*, Ph.D. thesis, University of Cambridge, 2000.

[113] W.D. Penny, "Variational Bayes for d-dimensional Gaussian mixture models," Tech. Rep., Wellcome Department of Cognitive Neurology, University College London, 2001.

[114] D.J.C. MacKay, "Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks," *Network: Computation in Neural Systems*, vol. 6, pp. 469–505, 1995.

[115] G.E. Hinton, "Connectionist learning procedures," *Artificial Intelligence*, vol. 40, pp. 185–234, 1989.

[116] T. Lee and M.S. Lewicki, "Image processing methods using ICA mixture models," in *Independent Component Analysis: Principles and Practice*, chapter 9, pp. 234–253. Cambridge Universiy Press, 2001.

[117] M.E. Tipping and C.M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.

[118] T. Lee, M.S. Lewicki, and T.J. Sejnowski, "ICA mixture models for unsupervised classification of non-Gaussian classes and automatic context switching in blind signal separation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, October 2000.

[119] W.D. Penny and S.J. Roberts, "Mixtures of Independent Component Analysers," in *Artificial Neural Networks - ICANN2001*. International Conference on Artificial Neural Networks, 2001, pp. 527–534.

[120] T.W. Lee, M.S. Lewicki, and T.J. Sejnowski, "ICA mixture models for unsupervised classification and automatic context switching," in *International Workshop on Independent Component Analysis*, 1999, pp. 209–214.

[121] S. Webb, Ed., *The Physics of Medical Imaging*, The Institute of Physics, 1996.

[122] R. Fisher, "The use of multiple measurements in taxonomic problem," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[123] T.K. Landaur, P.W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, pp. 295–284, 1998.

[124] W. Penny, S. Roberts, and R. Everson, "Hidden Markov Independent Components Analysis," in *Advances in Independent Component Analysis*, Mark Girolami, Ed. Kluwer Academic Publishers, 2000.

[125] B. Pearlmutter and L. Parra, "Maximum likelihood blind source separation: A context-sensitive generalization of ICA," in *Advances in Neural Information Processing Systems*, M.C Mozer, M.I Jordan, and T. Petsche, Eds. 1997, vol. 9, pp. 613–619, MIT press.

[126] H. Attias and C.E. Schreiner, "Blind source separation and deconvolution: the dynamic component analysis algorithm," *Neural Computation*, vol. 10, pp. 1373–1412, 1998.

[127] N. Murata, S. Ikeda, and A. Ziehe, "Adaptive on-line learning in changing environments," in *Advances in Neural Information Processing Systems*, M.C Mozer, M.I Jordan, and T. Petsche, Eds. 1997, vol. 9, pp. 599–605, MIT press.

[128] R Everson and S.J. Roberts, "Non-stationary Independent Component Analysis," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN'99)*. IEE, 1999.

[129] W.D. Penny and S.J. Roberts, "Hidden Markov Models with extended observation densities," Tech. Rep., Imperial College of Science Technology and Medicine, 1998.

[130] L. R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 257-286, 1989.

[131] Z. Ghahramani, "Learning dynamic Bayesian networks," Tech. Rep., Dept. of Computer Science, Univ. of Toronto, Toronto, 1997.

[132] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximisation technique occurring in the statistical analysis of Markov chains," *The Annals of Mathematical Statistics*, vol. 41, pp. 164–171, 1970.

[133] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE transactions on Information Theory*, vol. IT-13, pp. 260–269, 1967.

[134] H. Attias, "ICA and graphical models," in *Independent Component Analysis: Principles and Practice*, S Roberts and R Everson, Eds., chapter 3, pp. 95–112. Cambridge university press, 2001.

[135] John Cassidy, *Dot.Con*, Penguin, 2002.

[136] I. Rezek, M. Gibbs, and S.J. Roberts, "Maximum a posteriori estimation of coupled Hidden Markov Models," *Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, 2001, To appear.

[137] I. Stainvas and D. Lowe, "Illumination/segmentation model," in *Advances in Neural Information Processing Systems*, 2002, To appear.

[138] H. Valpola and P. Pajunen, "Fast algorithms for Bayesian Independent Component Analysis," in *Proceedings of ICA2000*, Helsinki, Finland, 2000, pp. 233–237.

[139] S.J. Roberts, C. Holmes, and D. Denison, "Minimum entropy data partitioning using reversible jump Markov chain Monte Carlo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 909–915, 2001.

[140] N. Ueda, R. Nakano, Z. Ghahramani, and G. Hinton, "SMEM algorithm for mixture models," *Advances in Neural Information Processing Systems*, vol. 11, 1999.

[141] L.C. Parra and C.D. Spence, "Separation of non-stationary sources," in *Independent Component Analysis: Principles and Practice*, S.J. Roberts and R. Everson, Eds., chapter 5, pp. 135–179. Cambridge University Press, 2001.

Appendix A

Probability Distributions

A.1 Gaussian

The Gaussian distribution is defined as

$$p(x|m,b) = \left(\frac{b}{2\pi}\right)^{\frac{1}{2}} \exp\left[-\frac{b}{2}(x-m)^2\right] \tag{A.1}$$

Expectations

Using the definition in (1.15)

$$\langle x \rangle = m \tag{A.2}$$

$$\langle x^2 \rangle = m^2 + \frac{1}{b} \tag{A.3}$$

Entropy

Using the definition in (1.14)

$$\mathcal{H}(x) = \frac{1}{2}\log\frac{2\pi e}{b} \tag{A.4}$$

KL-Divergence

Using the definition in (1.17) together with the following definitions

$$p(x|\theta_p) = \frac{b_p}{2\pi}^{\frac{1}{2}} \exp\left[-\frac{b_p}{2}(x-m_p)^2\right] \tag{A.5}$$

$$q(x|\theta_q) = \frac{b_q}{2\pi}^{\frac{1}{2}} \exp\left[-\frac{b_q}{2}(x-m_q)^2\right] \tag{A.6}$$

The KL-divegence between two normal pdfs over the same variable is

$$KL\left[q||p\right] = \frac{1}{2}\left[\left(\frac{b_p}{b_q} - 1\right) - \log\frac{b_p}{b_q} + b_p(m_q - m_p)^2\right] \tag{A.7}$$

A.2 Gamma

The Gamma distribution is defined as

$$p(x|b,c) = \frac{x^{c-1}}{\Gamma(c)b^c}\exp\left(-\frac{x}{b}\right) \tag{A.8}$$

for $x \geq 0$ and 0 for $x < 0$.

Expectations

Using the definitions in (1.15) and (1.20),

$$\langle x\rangle = bc \tag{A.9}$$

$$\langle x^2\rangle = bc(1+c) \tag{A.10}$$

$$\langle \log x\rangle = \Psi(c) + \log b \tag{A.11}$$

Entropy

Using the definition in (1.14)

$$\mathcal{H}(x) = \log b + c + \log\Gamma(c) - (c-1)\Psi(c) \tag{A.12}$$

KL-Divergence

Using the definition in (1.17) together with the following definitions

$$p(x|\theta_p) = \frac{x^{c_p-1}}{\Gamma(c_p)b_p^{c_p}}\exp\left(-\frac{x}{b_p}\right) \tag{A.13}$$

$$q(x|\theta_q) = \frac{x^{c_q-1}}{\Gamma(c_q)b_q^{c_q}}\exp\left(-\frac{x}{b_q}\right) \tag{A.14}$$

The KL-divegence between two gamma pdfs over the same variable is

$$KL\left[q||p\right] = c_q\left(\frac{b_q}{b_p} - 1\right) + (c_q - c_p)\left[\Psi(c_q) + \log b_q\right] - \log\left[\frac{\Gamma(c_q)b_q^{c_q}}{\Gamma(c_p)b_p^{c_p}}\right] \tag{A.15}$$

A.3 Dirichlet

The Dirichlet distribution is defined as

$$p(\boldsymbol{x}|\boldsymbol{c}) = \Gamma\left(\sum_{k'=1}^{d} c_{k'}\right) \prod_{k=1}^{d} \frac{x_k^{c_k-1}}{\Gamma(c_k)} \tag{A.16}$$

where $\boldsymbol{x} = \{x_1, x_2, \ldots, x_d\}$ and $\boldsymbol{c} = \{c_1, c_2, \ldots, c_d\}$.

Expectations

Using the definition in (1.15)

$$\langle x_k \rangle = \frac{c_k}{\sum_{k'} c_{k'}} \tag{A.17}$$

$$\langle \log x_k \rangle = \Psi(c_k) - \Psi\left(\sum_{k'} c_{k'}\right) \tag{A.18}$$

Entropy

Using the definition in (1.14)

$$\mathcal{H}(x) = \sum_{k=1}^{d} \log\Gamma(c_k) - \log\Gamma(\bar{c}) - \sum_{k=1}^{d}(c_k - 1)\left[\Psi(c_k) - \Psi(\bar{c})\right] \tag{A.19}$$

where $\bar{c} = \sum_k c_k$.

KL-Divergence

Using the definition in (1.17) together with the following definitions

$$p(x|\theta_p) = \Gamma\left(\sum_{k'=1}^{d} c_{p,k'}\right) \prod_{k=1}^{d} \frac{x_k^{c_p,k-1}}{\Gamma(c_{p,k})} \tag{A.20}$$

$$q(x|\theta_q) = \Gamma\left(\sum_{k'=1}^{d} c_{q,k'}\right) \prod_{k=1}^{d} \frac{x_k^{c_q,k-1}}{\Gamma(c_{q,k})} \tag{A.21}$$

The KL-divegence between two Dirichlet pdfs over the same variable is

$$KL\left[q||p\right] = \log\frac{\Gamma(\bar{c}_q)}{\Gamma(\bar{c}_p)} - \sum_{k=1}^{d}\log\frac{\Gamma(c_{q,k})}{\Gamma(c_{p,k})} + \sum_{k=1}^{d}(c_{q,k} - c_{p,k})\left[\Psi(c_{q,k}) - \Psi(\bar{c}_q)\right] \tag{A.22}$$

A.4 Exponential

The Exponential distribution is defined as

$$p(x|b) = \frac{1}{b}\exp\left(-\frac{x}{b}\right) \tag{A.23}$$

Expectations

Using the definition in (1.15)

$$\langle x \rangle = b \tag{A.24}$$

$$\langle x^2 \rangle = 2b^2 \tag{A.25}$$

Entropy

Using the definition in (1.14)

$$\mathcal{H}(x) = 1 + \log b \tag{A.26}$$

KL-Divergence

Using the definition in (1.17) together with the following definitions

$$p(x|\theta_p) = \frac{1}{b_p} \exp\left(-\frac{x}{b_p}\right) \tag{A.27}$$

$$q(x|\theta_q) = \frac{1}{b_q} \exp\left(-\frac{x}{b_q}\right) \tag{A.28}$$

The KL-divegence between two exponential pdfs over the same variable is

$$KL\left[q||p\right] = \left(\frac{b_q}{b_p} - 1\right) - \log \frac{b_q}{b_p} \tag{A.29}$$

A.5 Truncated Gaussian

The truncated Gaussian distribution is defined as

$$p(x|m, b) = \frac{1}{\text{erfc}\left(-m\frac{b}{2}\right)} \left(\frac{2b}{\pi}\right)^{\frac{1}{2}} \exp\left[-\frac{b}{2}\left(x - m\right)^2\right] \tag{A.30}$$

for $x \geq 0$ and 0 for $x < 0$. The term $\text{erfc}(a) = 1 - \text{erf}(a)$. The definition of $\text{erf}(a)$ is given by (1.21).

Expectations

Using the definition in (1.15)

$$\langle x \rangle = m + \sqrt{\frac{2}{\pi b}} \frac{1}{\text{erfcx}\left(-m\sqrt{\frac{b}{2}}\right)} \tag{A.31}$$

$$\langle x^2 \rangle = m^2 + \frac{1}{b} + m\sqrt{\frac{2}{\pi b}} \frac{1}{\text{erfcx}\left(-m\sqrt{\frac{b}{2}}\right)} \tag{A.32}$$

where $\text{erfcx}(a) = e^{a^2}\text{erfc}(a)$.

Entropy

Using the definition in (1.14)

$$\mathcal{H}(x) = \frac{1}{2}\log\frac{\pi e}{2b} + \log\text{erfc}\left(-m\frac{b}{2}\right) - m\sqrt{\frac{b}{2\pi}}\frac{1}{\text{erfcx}\left(-m\sqrt{\frac{b}{2}}\right)} \tag{A.33}$$

KL-Divergence

Using the definition in (1.17) together with the following definitions

$$p(x|\theta_p) = \frac{1}{\text{erfc}\left(-m_p\frac{b_p}{2}\right)}\left(\frac{2b_p}{\pi}\right)^{\frac{1}{2}}\exp\left[-\frac{b_p}{2}\left(x-m_p\right)^2\right] \tag{A.34}$$

$$q(x|\theta_q) = \frac{1}{\text{erfc}\left(-m_q\frac{b_q}{2}\right)}\left(\frac{2b_q}{\pi}\right)^{\frac{1}{2}}\exp\left[-\frac{b_q}{2}\left(x-m_q\right)^2\right] \tag{A.35}$$

The KL-divegence between two truncated Gaussian pdfs over the same variable is

$$\begin{aligned} KL\left[q||p\right] &= \frac{1}{2}\left[\left(\frac{b_p}{b_q}-1\right) - \log\frac{b_p}{b_q} + b_p(m_q-m_p)^2\right] \\ &+ \left[\left(b_p+b_q\right)m_q - 2b_p m_p\right]\sqrt{\frac{1}{2\pi b_q}}\frac{1}{\text{erfcx}\left(-m_q\sqrt{\frac{b_q}{2}}\right)} \\ &+ \log\frac{\text{erfc}\left(-m_p\frac{b_p}{2}\right)}{\text{erfc}\left(-m_q\frac{b_q}{2}\right)} \end{aligned} \tag{A.36}$$

Appendix B

Derivations for vbICA1

B.1 Source Model

$p'(\boldsymbol{q})$

The posterior over the indicator variables $\boldsymbol{q}$ is given by

$$\log p'(\boldsymbol{q}) \propto \langle \log p(\boldsymbol{S}|\boldsymbol{q},\boldsymbol{\mu},\boldsymbol{\beta})\rangle_{p'(\boldsymbol{S})p'(\boldsymbol{\mu})p'(\boldsymbol{\beta})} + \langle \log p(\boldsymbol{q}|\boldsymbol{\pi})\rangle_{p'(\boldsymbol{\pi})} \tag{B.1}$$

Substituting (5.10) and (5.9) into (B.1) gives

$$\begin{aligned}\log p'(\boldsymbol{q}) \quad &\propto \quad \sum_{t=1}^{T}\sum_{i=1}^{L} \frac{1}{2}\langle \log \beta_{i,q_i}\rangle - \frac{\langle \beta_{i,q_i}\rangle}{2}\left(\langle s_i^{t\,2}\rangle - 2\langle s_i^t\rangle\langle \mu_{i,q_i}\rangle + \langle \mu_{i,q_i}^2\rangle\right) \\ &+ \quad \sum_{t=1}^{T}\sum_{i=1}^{L}\langle \log \pi_{i,q_i}\rangle \end{aligned} \tag{B.2}$$

Exponentiating yields the posterior over $\boldsymbol{q}$

$$p'(\boldsymbol{q}) = \prod_{t=1}^{T}\prod_{i=1}^{L}\hat{\gamma}_{i,q_i}^t \tag{B.3}$$

where

$$\gamma_{i,q_i}^t \quad = \quad \tilde{\pi}_{i,q_i}\tilde{\beta}_{i,q_i}^{\frac{1}{2}} \exp\left[-\frac{\langle \beta_{i,q_i}\rangle}{2}\langle \left(s_i^t - \mu_{i,q_i}\right)^2\rangle\right] \tag{B.4}$$

$$\hat{\gamma}_{i,q_i}^t \quad = \quad \frac{\gamma_{i,q_i}^t}{\sum_{q_i'}\gamma_{i,q_i'}^t} \tag{B.5}$$

Equation (B.5) ensures that $\sum_{q_i}\hat{\gamma}_{i,q_i}^t = 1$. The tilded variables are exponentiated versions of $\langle \log \beta_{i,q_i}\rangle$ and $\langle \log \pi_{i,q_i}\rangle$ (under their respective posteriors) and are given by

$$\tilde{\pi}_{i,q_i} \quad = \quad \exp\left[\Psi(\hat{\lambda}_{i,q_i}) - \Psi(\sum_{q_i'}\hat{\lambda}_{i,q_i'})\right] \tag{B.6}$$

$$\tilde{\beta}_{i,q_i} = \hat{b}_{i,q_i} \exp\left[\Psi(\hat{c}_{i,q_i})\right] \tag{B.7}$$

where $\Psi(.)$ is the Digamma function

$p'(\boldsymbol{\mu})$

The posterior over the component means $\boldsymbol{\mu}$ is given by

$$\log p'(\boldsymbol{\mu}) \propto \langle \log p(\boldsymbol{S}|\boldsymbol{q}, \boldsymbol{\mu}, \boldsymbol{\beta}) \rangle_{p'(\boldsymbol{S})p'(\boldsymbol{q})p'(\boldsymbol{\beta})} + \log p(\boldsymbol{\mu}) \tag{B.8}$$

Substituting (5.10) and (5.13) into (B.8) gives

$$\begin{aligned} \log p'(\boldsymbol{\mu}) &\propto \sum_{i=1}^{L} \sum_{q_i=1}^{m_i} \sum_{t=1}^{T} \hat{\gamma}_{i,q_i}^t \left[\frac{1}{2} \langle \log \beta_{i,q_i} \rangle - \frac{\langle \beta_{i,q_i} \rangle}{2} \left(\langle s_i^{t\,2} \rangle - 2 \langle s_i^t \rangle \mu_{i,q_i} + \mu_{i,q_i}^2 \right) \right] \\ &+ \sum_{q_i=1}^{m_i} \frac{1}{2} \log \tau_{i0} - \frac{\tau_{i0}}{2} \left(\mu_{i,q_i}^2 - 2\mu_{i,q_i} m_{i0} + m_{i0}^2 \right) \end{aligned} \tag{B.9}$$

where $\langle a \rangle$ is the expectation w.r.t. $p'(a)$. Collecting together terms in μ_{i,q_i} gives

$$\begin{aligned} \log p'(\boldsymbol{\mu}) &\propto \sum_{i=1}^{L} \sum_{q_i=1}^{m_i} -\frac{1}{2} \left[\left(\tau_{i0} + \langle \beta_{i,q_i} \rangle \sum_{t=1}^{T} \hat{\gamma}_{i,q_i}^t \right) \mu_{i,q_i}^2 - 2 \left(\tau_{i0} m_{i0} + \langle \beta_{i,q_i} \rangle \sum_{t=1}^{T} \hat{\gamma}_{i,q_i}^t \langle s_i^t \rangle \right) \mu_{i,q_i} \right] \\ &\propto \sum_{i=1}^{L} \sum_{q_i=1}^{m_i} -\frac{1}{2} \left(\tau_{i0} + \langle \beta_{i,q_i} \rangle \sum_{t=1}^{T} \hat{\gamma}_{i,q_i}^t \right) \left[\mu_{i,q_i}^2 - 2 \frac{\tau_{i0} m_{i0} + \langle \beta_{i,q_i} \rangle \sum_t \hat{\gamma}_{i,q_i}^t \langle s_i^t \rangle}{\tau_{i0} + \langle \beta_{i,q_i} \rangle \sum_t \hat{\gamma}_{i,q_i}^t} \mu_{i,q_i} \right] \end{aligned} \tag{B.10}$$

As $\log p'(\boldsymbol{\mu})$ is a sum of $\sum_i m_i$ quadratics in μ_{i,q_i}, (B.10) implies $p'(\boldsymbol{\mu})$ is a product of $L \times \prod_i m_i$ Gaussian densities

$$p'(\boldsymbol{\mu}) = \prod_{i=1}^{L} \prod_{q_i=1}^{m_i} \mathcal{N}(\mu_{i,q_i}; \hat{m}_{i,q_i}, \hat{\tau}_{i,q_i}) \tag{B.11}$$

where

$$\hat{m}_{i,q_i} = \frac{1}{\hat{\tau}_{i,q_i}} \left(\tau_{i0} m_{i0} + \langle \beta_{i,q_i} \rangle \sum_{t=1}^{T} \hat{\gamma}_{i,q_i}^t \langle s_i^t \rangle \right) \tag{B.12}$$

$$\hat{\tau}_{i,q_i} = \tau_{i0} + \langle \beta_{i,q_i} \rangle \sum_{t=1}^{T} \hat{\gamma}_{i,q_i}^t \tag{B.13}$$

$p'(\boldsymbol{\beta})$

The posterior over the component precisions $\boldsymbol{\beta}$ is given by

$$\log p'(\boldsymbol{\beta}) \propto \langle \log p(\boldsymbol{S}|\boldsymbol{q}, \boldsymbol{\mu}, \boldsymbol{\beta}) \rangle_{p'(\boldsymbol{S})p'(\boldsymbol{q})p'(\boldsymbol{\mu})} + \log p(\boldsymbol{\beta}) \tag{B.14}$$

Substituting (5.10) and (5.14) into (B.14) gives

$$\begin{aligned}\log p'(\boldsymbol{\beta}) &\propto \sum_{i=1}^{L}\sum_{q_i=1}^{m_i}\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^{t}\left[\frac{1}{2}\log\beta_{i,q_i}-\frac{\beta_{i,q_i}}{2}\left(\langle s_i^{t\,2}\rangle-2\langle s_i^t\rangle\langle\mu_{i,q_i}\rangle+\langle\mu_{i,q_i}^2\rangle\right)\right]\\ &+ \sum_{i=1}^{L}\sum_{q_i=1}^{m_i}(c_{i0}-1)\log\beta_{i,q_i}-\frac{\beta_{i,q_i}}{b_{i0}} \qquad \text{(B.15)}\end{aligned}$$

Collecting together terms in β_{i,q_i} and using (4.27) gives

$$\begin{aligned}\log p'(\boldsymbol{\beta}) &\propto \sum_{i=1}^{L}\sum_{q_i=1}^{m_i}\left[\left(c_{i0}+\frac{1}{2}\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^{t}\right)-1\right]\log\beta_{i,q_i}\\ &\quad-\left[\frac{1}{b_{i0}}+\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^{t}\langle\left(s_i^t-\mu_{i,q_i}\right)^2\rangle\right]\beta_{i,q_i} \qquad \text{(B.16)}\end{aligned}$$

The functional form of (B.16) implies $p'(\boldsymbol{\beta})$ is a product of $L\times\prod_i m_i$ Gamma distributions

$$p'(\boldsymbol{\beta})=\prod_{i=1}^{L}\prod_{q_i=1}^{m_i}\mathcal{G}(\beta_{i,q_i};\hat{b}_{i,q_i},\hat{c}_{i,q_i}) \qquad \text{(B.17)}$$

where

$$\hat{b}_{i,q_i} = \left[\frac{1}{b_{i0}}+\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^{t}\langle\left(s_i^t-\mu_{i,q_i}\right)^2\rangle\right]^{-1} \qquad \text{(B.18)}$$

$$\hat{c}_{i,q_i} = c_{i0}+\frac{1}{2}\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^{t} \qquad \text{(B.19)}$$

$p'(\boldsymbol{\pi})$

The posterior over the indicator priors $\boldsymbol{\pi}$ is given by

$$\log p'(\boldsymbol{\pi})\propto\langle\log p(\boldsymbol{q}|\boldsymbol{\pi})\rangle_{p'(\boldsymbol{q})}+\log p(\boldsymbol{\pi}) \qquad \text{(B.20)}$$

Substituting (5.10) and (5.12) into (B.20) gives

$$\begin{aligned}\log p'(\boldsymbol{\pi}) &\propto \sum_{i=1}^{L}\sum_{q_i=1}^{m_i}\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^{t}\langle\log\pi_{i,q_i}\rangle+\sum_{i=1}^{L}\sum_{q_i=1}^{m_i}(\lambda_{i0}-1)\log\pi_{i,q_i}\\ &\propto \sum_{i=1}^{L}\sum_{q_i=1}^{m_i}\left[\left(\lambda_{i0}+\sum_{t=1}^{T}\hat{\gamma}_{i,q_i}^{t}\right)-1\right]\log\pi_{i,q_i} \qquad \text{(B.21)}\end{aligned}$$

The functional form of (B.21) implies a product of L non-symmetric Dirichlets for $p'(\boldsymbol{\pi})$

$$p'(\boldsymbol{\pi})=\prod_{i=1}^{L}\Gamma(\sum_{q_i'}\hat{\lambda}_{q_i'})\prod_{q_i=1}^{m}\frac{\pi_{i,q_i}^{\hat{\lambda}_{i,q_i}-1}}{\Gamma(\hat{\lambda}_{i,q_i})} \qquad \text{(B.22)}$$

where

$$\hat{\lambda}_{i,q_i} = \lambda_{i0} + \sum_{t=1}^{T} \hat{\gamma}_{i,q_i}^t \tag{B.23}$$

B.2 Network Model

$p'(\boldsymbol{S})$

The posterior over the source signals $\boldsymbol{S}$ is given by

$$\log p'(\boldsymbol{S}) \propto \langle \log p(\boldsymbol{X}|\boldsymbol{S},\boldsymbol{A},\boldsymbol{\Lambda}) \rangle_{p'(\boldsymbol{A})p'(\boldsymbol{\Lambda})} + \langle \log p(\boldsymbol{S}|\boldsymbol{q},\boldsymbol{\mu},\boldsymbol{\beta}) \rangle_{p'(\boldsymbol{q})p'(\boldsymbol{\mu})p'(\boldsymbol{\beta})} \tag{B.24}$$

Substituting (5.24) and (5.10) into (B.24) and keeping only terms in s_i^t gives

$$\begin{aligned} \log p'(\boldsymbol{S}) \quad \propto \quad & \sum_{t=1}^{T}\sum_{i=1}^{L}\sum_{j=1}^{M} -\frac{\langle \Lambda_j \rangle}{2} \left[\langle A_{ji}^2 \rangle s_i^{t\,2} - 2 s_i^t \langle A_{ji} \rangle \left(x_j^t - \sum_{k \neq i}^{L} \langle A_{jk} \rangle \langle s_k^t \rangle \right) \right] \\ + \quad & \sum_{t=1}^{T}\sum_{i=1}^{L}\sum_{q_i=1}^{m_i} p'(\boldsymbol{q}) \left[-\frac{\langle \beta_{i,q_i} \rangle}{2} \left(s_i^{t\,2} - 2 s_i^t \langle \mu_{i,q_i} \rangle \right) \right] \end{aligned} \tag{B.25}$$

Collecting together terms in s_i^t

$$\begin{aligned} \log p'(\boldsymbol{S}) \quad \propto \quad & -\frac{1}{2} \sum_{t=1}^{T}\sum_{i=1}^{L} \left(\sum_{q_i=1}^{m_i} \hat{\gamma}_{i,q_i}^t \langle \beta_{i,q_i} \rangle + \sum_{j=1}^{M} \langle \Lambda_j \rangle \langle A_{ji}^2 \rangle \right) s_i^{t\,2} \\ + \quad & 2 \left[\sum_{q_i=1}^{m_i} \hat{\gamma}_{i,q_i}^t \langle \beta_{i,q_i} \rangle \langle \mu_{i,q_i} \rangle + \sum_{j=1}^{M} \langle \Lambda_j \rangle \langle A_{ji} \rangle \left(x_j^t - \sum_{k \neq i}^{L} \langle A_{jk} \rangle \langle s_k^t \rangle \right) \right] s_i^t \end{aligned}$$

The quadratic form for $\log p'(\boldsymbol{S})$ implies

$$p'(\boldsymbol{S}) = \prod_{t=1}^{T} \prod_{i=1}^{L} \mathcal{N}(s_i^t; \hat{\mu}_i^t, \hat{\beta}_i^t) \tag{B.26}$$

where

$$\hat{\mu}_i^t = \frac{1}{\hat{\beta}_i^t} \left[\sum_{q_i=1}^{m_i} \hat{\gamma}_{i,q_i}^t \langle \beta_{i,q_i} \rangle \langle \mu_{i,q_i} \rangle + \sum_{j=1}^{M} \langle \Lambda_j \rangle \langle A_{ji} \rangle \left(x_j^t - \sum_{k \neq i}^{L} \langle A_{jk} \rangle \langle s_k^t \rangle \right) \right] \tag{B.27}$$

$$\hat{\beta}_i^t = \sum_{q_i=1}^{m_i} \hat{\gamma}_{i,q_i}^t \langle \beta_{i,q_i} \rangle + \sum_{j=1}^{M} \langle \Lambda_j \rangle \langle A_{ji}^2 \rangle \tag{B.28}$$

$p'(\boldsymbol{A})$

The posterior over the mixing matrix $\boldsymbol{A}$ is given by

$$\log p'(\boldsymbol{A}) \propto \langle \log p(\boldsymbol{X}|\boldsymbol{S},\boldsymbol{A},\boldsymbol{\Lambda}) \rangle_{p'(\boldsymbol{S})p'(\boldsymbol{\Lambda})} + \log p(\boldsymbol{A}) \tag{B.29}$$

Substituting (5.24) and (5.16) into (B.29) and keeping only terms in A_{ji} gives

$$\begin{aligned}\log p'(\boldsymbol{A}) &\propto \sum_{i=1}^{L}\sum_{j=1}^{M}\sum_{t=1}^{T} -\frac{\langle\Lambda_j\rangle}{2}\left[\langle s_i^{t\,2}\rangle A_{ji}^2 - 2A_{ji}\langle s_i^t\rangle\left(x_j^t - \sum_{k\neq i}^{L}\langle A_{jk}\rangle\langle s_k^t\rangle\right)\right] \\ &+ \sum_{i=1}^{L}\sum_{j=1}^{M} -\frac{\alpha_{ji}}{2}A_{ji}^2 \qquad \text{(B.30)}\end{aligned}$$

Collecting together terms in A_{ij}

$$\begin{aligned}\log p'(\boldsymbol{A}) &\propto -\frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{M}\left(\alpha_{ji} + \langle\Lambda_j\rangle\sum_{t=1}^{T}\langle s_i^{t\,2}\rangle\right)A_{ji}^2 \qquad \text{(B.31)} \\ &- 2\left[\langle\Lambda_j\rangle\sum_{t=1}^{T}\langle s_i^t\rangle\left(x_j^t - \sum_{k\neq i}^{L}\langle A_{jk}\rangle\langle s_k^t\rangle\right)\right]A_{ji}\end{aligned}$$

The quadratic form for $\log p'(\boldsymbol{A})$ implies

$$p'(\boldsymbol{A}) = \prod_{i=1}^{L}\prod_{j=1}^{M}\mathcal{N}(A_{ji}; \hat{m}_{A_{ji}}, \hat{\alpha}_{ji}) \qquad \text{(B.32)}$$

where

$$\hat{m}_{A_{ji}} = \frac{\langle\Lambda_j\rangle}{\hat{\alpha}_{ji}}\sum_{t=1}^{T}\langle s_i^t\rangle\left(x_j^t - \sum_{k\neq i}^{L}\langle A_{jk}\rangle\langle s_k^t\rangle\right) \qquad \text{(B.33)}$$

$$\hat{\alpha}_{ji} = \alpha_{ji} + \langle\Lambda_j\rangle\sum_{t=1}^{T}\langle s_i^{t\,2}\rangle \qquad \text{(B.34)}$$

$p'(\boldsymbol{\Lambda})$

The posterior over the noise precision $\boldsymbol{\Lambda}$ is given by

$$\log p'(\boldsymbol{\Lambda}) \propto \langle\log p(\boldsymbol{X}|\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{\Lambda})\rangle_{p'(\boldsymbol{S})p'(\boldsymbol{A})} + \log p(\boldsymbol{\Lambda}) \qquad \text{(B.35)}$$

Substituting (5.24) and (5.15) into (B.35) and keeping only terms in Λ_j gives

$$\begin{aligned}\log p'(\boldsymbol{\Lambda}) &\propto \sum_{j=1}^{M}\sum_{t=1}^{T}\frac{1}{2}\log\Lambda_j - \frac{\Lambda_j}{2}\left(x_j^{t\,2} - 2x_j^t\sum_{i=1}^{L}\langle A_{ji}\rangle\langle s_i^t\rangle + \langle\sum_{i=1}^{L}A_{ji}s_i^t\sum_{i=1}^{L}A_{ji}s_i^t\rangle\right) \\ &+ \sum_{j=1}^{M}(c_{\Lambda_j} - 1)\log\Lambda_j - \frac{\Lambda_j}{b_{\Lambda_j}} \qquad \text{(B.36)}\end{aligned}$$

Collecting together terms in Λ_j

$$\begin{aligned}\log p'(\boldsymbol{\Lambda}) &\propto \sum_{j=1}^{M}\left[\left(c_{\Lambda_j} + \frac{T}{2}\right) - 1\right]\log\Lambda_j \\ &- \left[\frac{1}{b_{\Lambda_j}} + \frac{1}{2}\sum_{t=1}^{T}\left(x_j^{t\,2} - 2x_j^t\sum_{i=1}^{L}\langle A_{ji}\rangle\langle s_i^t\rangle + \langle\sum_{i=1}^{L}A_{ji}s_i^t\sum_{i=1}^{L}A_{ji}s_i^t\rangle\right)\right]\Lambda_j \qquad \text{(B.37)}\end{aligned}$$

The functional form of $\log p'(\mathbf{\Lambda})$ implies the posterior over $\mathbf{\Lambda}$ is a product of M Gamma distributions

$$p'(\mathbf{\Lambda}) = \prod_{j=1}^{M} \mathcal{G}(\Lambda_v; \hat{b}_{\Lambda_j}, \hat{c}_{\Lambda_j}) \tag{B.38}$$

where

$$\hat{b}_{\Lambda_j} = \left[\frac{1}{b_{\Lambda_j}} + \frac{1}{2} \sum_{t=1}^{T} \left({x_j^t}^2 - 2x_j^t \sum_{i=1}^{L} \langle A_{ji} \rangle \langle s_i^t \rangle + \langle \sum_{i=1}^{L} A_{ji} s_i^t \sum_{i'=1}^{L} A_{ji'} s_{i'}^t \rangle \right) \right]^{-1} \tag{B.39}$$

$$\hat{c}_{\Lambda_j} = c_{\Lambda_j} + \frac{T}{2} \tag{B.40}$$

Equation (B.39) can be rewritten as

$$\hat{b}_{\Lambda_j} = \left[\frac{1}{b_{\Lambda_j}} + \frac{1}{2} \sum_{t=1}^{T} \langle \left(x_j^t - \hat{x}_j^t \right)^2 \rangle \right]^{-1} \tag{B.41}$$

where

$$\hat{x}_j^t = \sum_{i=1}^{L} A_{ji} s_i^t \tag{B.42}$$

The expectations in (B.41) are w.r.t. $p'(\boldsymbol{S})$ and $p'(\boldsymbol{A})$ and are given by

$$\langle \hat{x}_j^t \rangle = \sum_{i=1}^{L} \langle A_{ji} \rangle \langle s_i^t \rangle \tag{B.43}$$

$$\langle \hat{x}_i^t \hat{x}_j^t \rangle = \langle \hat{x}_i^t \rangle \langle \hat{x}_j^t \rangle \tag{B.44}$$

$$\langle \hat{x}_j^t \hat{x}_j^t \rangle = \langle \hat{x}_j^t \rangle^2 + \sum_{i=1}^{L} \langle A_{ji}^2 \rangle \langle {s_i^t}^2 \rangle - \langle A_{ji} \rangle^2 \langle s_i^t \rangle^2 \tag{B.45}$$

B.3 Free Energy

The negative free energy for vbICA1 simplifies in a similar way to the MoG example in section 4.4.4, so - for brevity - will not be explicitly derived. The expression for NFE for vbICA1 is

$$\begin{aligned} F_{\text{vbICA1}} &= \sum_{i=1}^{L} \left[\sum_{q_i=1}^{m_i} \log \frac{\Gamma\left(\hat{\lambda}_{i,q_i}\right)}{\Gamma\left(\lambda_{i0}\right)} - \log \frac{\Gamma\left(\sum_{q_i'} \hat{\lambda}_{i,q_i'}\right)}{\Gamma\left(m_i \lambda_{i0}\right)} \right] \\ &+ \sum_{i=1}^{L} \sum_{q_i=1}^{m_i} \log \frac{\Gamma\left(\hat{c}_{i,q_i}\right) \hat{b}_{i,q}^{\hat{c}_{i,q_i}}}{\Gamma\left(c_{i0}\right) b_{i0}^{c_{i0}}} + \sum_{t=1}^{T} \sum_{i=1}^{L} \sum_{q_i=1}^{m_i} \hat{\gamma}_{i,q_i}^t \log \hat{\gamma}_{i,q_i}^t \\ &- \frac{1}{2} \sum_{i=1}^{L} \sum_{q_i=1}^{m_i} \left[\frac{\tau_{i0}}{\hat{\tau}_{i,q_i}} - \log \frac{\tau_{i0}}{\hat{\tau}_{i,q_i}} + \tau_{i0} \left(\hat{m}_{i,q_i} - m_{i0} \right)^2 \right] + \frac{L}{2} \end{aligned}$$

$$
\begin{aligned}
&- \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{L}\log\hat{\beta}_i^t + \frac{TL}{2} \\
&- \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{M}\left[\frac{\alpha_{ji}}{\hat{\alpha}_{ji}} - \log\frac{\alpha_{ji}}{\hat{\alpha}_{ji}} + \alpha_{ji}\hat{m}_{A_{ji}}^2\right] + \frac{LM}{2} \\
&+ \sum_{j=1}^{M}\log\frac{\Gamma\left(\hat{c}_{\Lambda_j}\right)\hat{b}_{\Lambda_j}^{\hat{c}_{\Lambda_j}}}{\Gamma\left(c_{\Lambda_j}\right)b_{\Lambda_j}^{c_{\Lambda_j}}} - \frac{TM}{2}\log 2\pi
\end{aligned}
\tag{B.46}
$$

This can be written as

$$
F_{\text{vbICA1}} = \sum_{i=1}^{L} F_{\text{MoG}_i} + F_{\text{src}} + F_{\text{mix}} + F_{\text{nse}} \tag{B.47}
$$

where

$$
\begin{aligned}
F_{\text{MoG}_i} &= \left[\sum_{q_i=1}^{m_i}\log\frac{\Gamma\left(\hat{\lambda}_{i,q_i}\right)}{\Gamma\left(\lambda_{i0}\right)} - \log\frac{\Gamma\left(\sum_{q_i'}\hat{\lambda}_{i,q_i'}\right)}{\Gamma\left(m_i\lambda_{i0}\right)}\right] \\
&+ \sum_{q_i=1}^{m_i}\log\frac{\Gamma\left(\hat{c}_{i,q_i}\right)\hat{b}_{i,q}^{\hat{c}_{i,q_i}}}{\Gamma\left(c_{i0}\right)b_{i0}^{c_{i0}}} + \sum_{t=1}^{T}\sum_{q_i=1}^{m_i}\hat{\gamma}_{i,q_i}^t\log\hat{\gamma}_{i,q_i}^t \\
&- \frac{1}{2}\sum_{q_i=1}^{m_i}\left[\frac{\tau_{i0}}{\hat{\tau}_{i,q_i}} - \log\frac{\tau_{i0}}{\hat{\tau}_{i,q_i}} + \tau_{i0}\left(\hat{m}_{i,q_i} - m_{i0}\right)^2\right] + \frac{1}{2}
\end{aligned}
\tag{B.48}
$$

$$
F_{\text{src}} = \frac{TL}{2} - \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{L}\log\hat{\beta}_i^t \tag{B.49}
$$

$$
F_{\text{mix}} = \frac{LM}{2} - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{M}\left[\frac{\alpha_{ji}}{\hat{\alpha}_{ji}} - \log\frac{\alpha_{ji}}{\hat{\alpha}_{ji}} + \alpha_{ji}\hat{m}_{A_{ji}}^2\right] \tag{B.50}
$$

$$
F_{\text{nse}} = \sum_{j=1}^{M}\log\frac{\Gamma\left(\hat{c}_{\Lambda_j}\right)\hat{b}_{\Lambda_j}^{\hat{c}_{\Lambda_j}}}{\Gamma\left(c_{\Lambda_j}\right)b_{\Lambda_j}^{c_{\Lambda_j}}} - \frac{TM}{2}\log 2\pi \tag{B.51}
$$

Appendix C

Derivations for vbICA2

C.1 Network Model

$p'(\boldsymbol{S}|\boldsymbol{q})$

To get a clearer understanding of how to optimise $p'(\boldsymbol{S}|\boldsymbol{q})$, first of all write down the portion of the free energy dependent on $\boldsymbol{S}$

$$\begin{aligned} F_{\boldsymbol{S}} &= \langle \log p(\boldsymbol{X}|\boldsymbol{S}.\boldsymbol{A}, \boldsymbol{\Lambda}) \rangle_{p'(\boldsymbol{A})p'(\boldsymbol{\Lambda})p'(\boldsymbol{S}|\boldsymbol{q})p'(\boldsymbol{q})} \\ &+ \langle \log p(\boldsymbol{S}, \boldsymbol{q}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\beta}) \rangle_{p'(\boldsymbol{S}|\boldsymbol{q})p'(\boldsymbol{q})p'(\boldsymbol{\pi})p'(\boldsymbol{\mu})p'(\boldsymbol{\beta})} \\ &+ \mathcal{H}\left[p'(\boldsymbol{S}, \boldsymbol{q})\right] \end{aligned} \tag{C.1}$$

The entropy in (C.1) can be rewritten in terms of expectations

$$\mathcal{H}\left[p'(\boldsymbol{S}, \boldsymbol{q})\right] = \langle (\mathcal{H}\left[p'(\boldsymbol{S}|\boldsymbol{q})\right] - \log p'(\boldsymbol{q})) \rangle_{p'(\boldsymbol{q})} \tag{C.2}$$

Substituting (C.2) into (C.1), and optimising using (4.10) gives

$$\begin{aligned} \langle \log p'(\boldsymbol{S}|\boldsymbol{q}) \rangle_{p'(\boldsymbol{q})} &\propto \langle \log p(\boldsymbol{X}|\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{\Lambda}) \rangle_{p'(\boldsymbol{A})p'(\boldsymbol{\Lambda})p'(\boldsymbol{q})} \\ &+ \langle p(\boldsymbol{S}|\boldsymbol{q}, \boldsymbol{\mu}, \boldsymbol{\beta}) \rangle_{p'(\boldsymbol{\mu})p'(\boldsymbol{\beta})p'(\boldsymbol{q})} \\ &- \langle \log p'(\boldsymbol{q}) \rangle_{p'(\boldsymbol{q})} \end{aligned} \tag{C.3}$$

Cancelling through the expectation w.r.t. $p'(\boldsymbol{q})$ Substituting (5.10) and (5.9) into (C.3) and keeping only terms in s_i^t gives

$$\log p'(\boldsymbol{S}|\boldsymbol{q}) \propto \sum_{t=1}^{T} \sum_{i=1}^{L} \sum_{j=1}^{M} -\frac{\langle \Lambda_j \rangle}{2} \left[\langle A_{ji}^2 \rangle s_i^{t\,2} - 2 s_i^t \langle A_{ji} \rangle \left(x_j^t - \sum_{k \neq i}^{L} \langle A_{jk} \rangle \langle s_k^t \rangle \right) \right]$$

$$+ \quad \sum_{t=1}^{T}\sum_{i=1}^{L}\left[-\frac{\langle\beta_{i,q_i}\rangle}{2}\left(s_i^{t\,2} - 2s_i^t\langle\mu_{i,q_i}\rangle\right)\right] \tag{C.4}$$

Collecting together terms in s_i^t

$$\begin{aligned}\log p'(\boldsymbol{S}|\boldsymbol{q}) \quad &\propto \quad -\frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{L}\left(\langle\beta_{i,q_i}\rangle + \sum_{j=1}^{M}\langle\Lambda_j\rangle\langle A_{ji}^2\rangle\right)s_i^{t\,2} \\ &+ \quad 2\left[\langle\beta_{i,q_i}\rangle\langle\mu_{i,q_i}\rangle + \sum_{j=1}^{M}\langle\Lambda_j\rangle\langle A_{ji}\rangle\left(x_j^t - \sum_{k\neq i}^{L}\langle A_{jk}\rangle\langle s_k^t\rangle\right)\right]s_i^t\end{aligned}$$

The quadratic form for $\log p'(\boldsymbol{S}|\boldsymbol{q})$ implies

$$p'(\boldsymbol{S}|\boldsymbol{q}) = \prod_{t=1}^{T}\prod_{i=1}^{L}\mathcal{N}(s_i^t; \hat{\mu}_{i,q_i}^t, \hat{\beta}_{i,q_i}^t) \tag{C.5}$$

where

$$\hat{\mu}_{i,q_i}^t \quad = \quad \frac{1}{\hat{\beta}_{i,q_i}^t}\left[\langle\beta_{i,q_i}\rangle\langle\mu_{i,q_i}\rangle + \sum_{j=1}^{M}\langle\Lambda_j\rangle\langle A_{ji}\rangle\left(x_j^t - \langle\hat{x}_{j,k\neq i}^t\rangle\right)\right] \tag{C.6}$$

$$\hat{\beta}_{i,q_i}^t \quad = \quad \langle\beta_{i,q_i}\rangle + \sum_{j=1}^{M}\langle\Lambda_j\rangle\langle A_{ji}^2\rangle \tag{C.7}$$

where

$$\langle\hat{x}_{j,k\neq i}^t\rangle \quad = \quad \sum_{k\neq i}^{L}\langle A_{jk}\rangle\langle s_k^t\rangle$$

In practice, equation (C.6) has to be iterated for every i a number of times until $\hat{\mu}_{i,q_i}^t$ converges as it depends on every other $k \neq i$.

The optimisation procedure for $p'(\boldsymbol{A})$ and $p'(\boldsymbol{\Lambda})$ is identical to that in Appendix B.

C.2 Source Model

$p'(\boldsymbol{q})$

To get a clearer understanding of how to optimise $p'(\boldsymbol{q})$, first of all, write down the portion of the free energy dependent on $\boldsymbol{q}$

$$\begin{aligned}F_{\boldsymbol{q}} \quad &= \quad \langle\log p(\boldsymbol{X}|\boldsymbol{S}.\boldsymbol{A}, \boldsymbol{\Lambda})\rangle_{p'(\boldsymbol{A})p'(\boldsymbol{\Lambda})p'(\boldsymbol{S}|\boldsymbol{q})p'(\boldsymbol{q})} \\ &+ \quad \langle\log p(\boldsymbol{S}, \boldsymbol{q}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\beta})\rangle_{p'(\boldsymbol{S}|\boldsymbol{q})p'(\boldsymbol{q})p'(\boldsymbol{\pi})p'(\boldsymbol{\mu})p'(\boldsymbol{\beta})} \\ &+ \quad \mathcal{H}\left[p'(\boldsymbol{S}, \boldsymbol{q})\right]\end{aligned} \tag{C.8}$$

The entropy in (C.8) can be rewritten in terms of expectations using (C.2) Substituting (5.10), (5.9) and (C.2) into (C.8), and optimising using (4.10) gives

$$\begin{aligned}\log p'(\boldsymbol{q}) \quad \propto \quad & \sum_{t=1}^{T}\sum_{i=1}^{L}\sum_{j=1}^{M}\frac{\langle\Lambda_j\rangle}{2}\left(2x_j\langle A_{ji}\rangle\langle s_i^t|q_i\rangle - 2\langle A_{ji}\rangle\langle s_i^t|q_i\rangle\sum_{k\neq i}^{L}\langle A_{jk}\rangle\langle s_k^t\rangle - \langle A_{ji}^2\rangle\langle {s_i^t}^2|q_i\rangle\right) \\ + \quad & \sum_{t=1}^{T}\sum_{i=1}^{L}\frac{1}{2}\left[\langle\log\beta_{i,q_i}\rangle - \langle\beta_{i,q_i}\rangle\left(\langle {s_i^t}^2|q_i\rangle - 2\langle s_i^t|q_i\rangle\langle\mu_{i,q_i}\rangle + \langle\mu_{i,q_i}^2\rangle\right)\right] \\ + \quad & \sum_{t=1}^{T}\sum_{i=1}^{L}\langle\log\pi_{i,q_i}\rangle - \sum_{t=1}^{T}\sum_{i=1}^{L}\frac{1}{2}\log\hat{\beta}_{i,q_i}^t \end{aligned} \tag{C.9}$$

After (much) rearranging

$$\begin{aligned}\log p'(\boldsymbol{q}) \quad \propto \quad & \sum_{t=1}^{T}\sum_{i=1}^{L}\langle\log\pi_{i,q_i}\rangle + \frac{1}{2}\langle\log\beta_{i,q_i}\rangle - \frac{1}{2}\log\hat{\beta}_{i,q_i}^t \\ + \quad & \sum_{t=1}^{T}\sum_{i=1}^{L}\left[\langle\beta_{i,q_i}\rangle\langle\mu_{i,q_i}\rangle + \sum_{j=1}^{M}\langle\Lambda_j\rangle\langle A_{ji}\rangle\left(x_j - \langle\hat{x}_{j,k\neq i}\rangle\right)\right]\langle s_i^t|q_i\rangle \\ - \quad & \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{L}\left[\langle\beta_{i,q_i}\rangle + \sum_{j=1}^{M}\langle\Lambda_j\rangle\langle A_{ji}^2\rangle\right]\langle {s_i^t}^2|q_i\rangle \\ - \quad & \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{L}\langle\beta_{i,q_i}\rangle\langle\mu_{i,q_i}^2\rangle \end{aligned} \tag{C.10}$$

Noting the forms of (C.6) and (C.7) and utilising the associated expectations for $\langle s_i^t|q_i\rangle$ and $\langle {s_i^t}^2|q_i\rangle$ gives

$$\begin{aligned}\log p'(\boldsymbol{q}) \quad \propto \quad & \sum_{t=1}^{T}\sum_{i=1}^{L}\langle\log\pi_{i,q_i}\rangle + \frac{1}{2}\langle\log\beta_{i,q_i}\rangle - \frac{1}{2}\log\hat{\beta}_{i,q_i}^t \\ + \quad & \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{L}\hat{\beta}_{i,q_i}^t\hat{\mu}_{i,q_i}^{t^2} - \langle\beta_{i,q_i}\rangle\langle\mu_{i,q_i}^2\rangle \end{aligned} \tag{C.11}$$

Exponentiating yields the posterior over $\boldsymbol{q}$

$$p'(\boldsymbol{q}) = \prod_{t=1}^{T}\prod_{i=1}^{L}\hat{\gamma}_{i,q_i}^t \tag{C.12}$$

where

$$\gamma_{i,q_i}^t = \tilde{\pi}_{i,q_i}\left(\frac{\tilde{\beta}_{i,qi}}{\hat{\beta}_{i,qi}^t}\right)^{\frac{1}{2}}\exp\left[\frac{1}{2}\left(\hat{\beta}_{i,qi}^t\hat{\mu}_{i,q_i}^{t^2} - \langle\beta_{i,qi}\rangle\langle\mu_{i,q_i}^2\rangle\right)\right] \tag{C.13}$$

$$\hat{\gamma}_{i,q_i}^t = \frac{\gamma_{i,q_i}^t}{\sum_{q_i'}\gamma_{i,q_i'}^t} \tag{C.14}$$

and where $\tilde{\pi}_{i,q_i}$ and $\tilde{\beta}_{i,qi}$ are given by (B.6) and (B.7) respectively.

The optimisation procedure for $p'(\boldsymbol{\theta})$ is similar to that in Appendix B, with expectation w.r.t. $p'(\boldsymbol{S}|\boldsymbol{q})$ instead of $p'(\boldsymbol{S})$.

C.3 Energy

The negative free energy for vbICA2 is the same as (B.47), but with a different expression for F_{src}

$$F_{\text{src}} = \frac{TL}{2} - \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{L}\sum_{q_i=1}^{m_i}\hat{\gamma}_{i,q_i}^{t}\log\hat{\beta}_{i,q_i}^{t} \tag{C.15}$$

Appendix D

Derivations for Prior Constraints

D.1 ARD

Update Equations

The posterior over the mixing matrix $\boldsymbol{A}$ is given by

$$\log p'(\boldsymbol{A}) \propto \langle \log p(\boldsymbol{X}|\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{\Lambda}) \rangle_{p'(\boldsymbol{S})p'(\boldsymbol{\Lambda})} + \langle \log p(\boldsymbol{A}|\boldsymbol{\alpha}) \rangle_{p'(\boldsymbol{\alpha})} \tag{D.1}$$

This is the same as section B.2 with the additional marginalisation over $p'(\boldsymbol{\alpha})$. Therefore, the posterior over $\boldsymbol{A}$ is simply given by (5.50) and (5.49) with further expectations around functions of α_{ji}.

The posterior over the ARD coefficients $\boldsymbol{\alpha}$ is given by

$$\log p'(\boldsymbol{\alpha}) \propto \langle \log p(\boldsymbol{A}|\boldsymbol{\alpha}) \rangle_{p'(\boldsymbol{A})} + \log p(\boldsymbol{\alpha}) \tag{D.2}$$

Substituting (5.16) and (5.76) into (D.2) and keeping only terms in α_i gives

$$\log p'(\boldsymbol{\alpha}) \propto \sum_{i=1}^{L} \sum_{j=1}^{M} \frac{1}{2} \log \alpha_i - \frac{\alpha_i}{2} \langle A_{ji}^2 \rangle + \sum_{i=1}^{L} (c_{\alpha_i} - 1) \log \alpha_i - \frac{\alpha_i}{b_{\alpha_i}} \tag{D.3}$$

Collecting together terms in α_i

$$\log p'(\boldsymbol{\alpha}) \propto \sum_{i=1}^{L} \left[\left(c_{\alpha_i} + \frac{M}{2} \right) - 1 \right] \log \alpha_i - \left(\frac{1}{b_{\alpha_i}} + \frac{1}{2} \sum_{j=1}^{M} \langle A_{ji}^2 \rangle \right) \alpha_i \tag{D.4}$$

The functional form of $\log p'(\boldsymbol{\alpha})$ implies the posterior over $\boldsymbol{\alpha}$ is a product of L Gamma distributions

$$p'(\boldsymbol{\alpha}) = \prod_{i=1}^{L} \mathcal{G}(\alpha_i; \hat{b}_{\alpha_i}, \hat{c}_{\alpha_i}) \tag{D.5}$$

where

$$\hat{b}_{\alpha_i} = \left(\frac{1}{b_{\alpha_i}} + \frac{1}{2} \sum_{j=1}^{M} \langle A_{ji}^2 \rangle \right)^{-1} \tag{D.6}$$

$$\hat{c}_{\alpha_i} = c_{\alpha_i} + \frac{M}{2} \tag{D.7}$$

Energy

The negative free energy with ARD is similar to that of vbICA1 [equation (B.47)] and vbICA2 [substitution (C.15)] but with a different expression for F_{mix}

$$F_{\text{mix}} = \sum_{i=1}^{L} \log \frac{\Gamma\left(\hat{c}_{\alpha_i}\right) \hat{b}_{\alpha_i}^{\hat{c}_{\alpha_i}}}{\Gamma\left(c_{\alpha_i}\right) b_{\alpha_i}^{c_{\alpha_i}}} - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{M} \log \hat{\alpha}_{ji} + \frac{ML}{2} \tag{D.8}$$

D.2 Positivity

The derivation of the non-negative source posteriors follow a similar vein to the previous 2 appendices. The NFE is modified by the following

$$\begin{aligned} F_{\text{src}} &= \frac{TL}{2} - \frac{1}{2} \sum_{t=1}^{T} \sum_{i=1}^{L} \log \hat{\beta}_i^t \\ &+ \sum_{t=1}^{T} \sum_{i=1}^{L} \log \operatorname{erfc}\left(-\hat{\mu}_i^t \frac{\hat{\beta}_i^t}{2} \right) - \hat{\mu}_i^t \sqrt{\frac{\hat{\beta}_i^t}{2\pi}} \frac{1}{\operatorname{erfcx}\left(\hat{\mu}_i^t \sqrt{\frac{\hat{\beta}_i^t}{2}} \right)} \end{aligned} \tag{D.9}$$

$$\begin{aligned} F_{\text{mix}} &= \frac{ML}{2} - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{M} \log \hat{\alpha}_{ji} + \sum_{i=1}^{L} \log \frac{\Gamma\left(\hat{c}_{\alpha_i}\right) \hat{b}_{\alpha_i}^{\hat{c}_{\alpha_i}}}{\Gamma\left(c_{\alpha_i}\right) b_{\alpha_i}^{c_{\alpha_i}}} \\ &+ \sum_{i=1}^{L} \sum_{j=1}^{M} \log \operatorname{erfc}\left(-\hat{m}_{A_{ji}} \frac{\hat{\alpha}_{ji}}{2} \right) - \hat{m}_{A_{ji}} \sqrt{\frac{\hat{\alpha}_{ji}}{2\pi}} \frac{1}{\operatorname{erfcx}\left(\hat{m}_{A_{ji}} \sqrt{\frac{\hat{\alpha}_{ji}}{2}} \right)} \end{aligned} \tag{D.10}$$

These are true for both tGaussian and exponential sources, although $\sum_{i=1}^{L} \log \frac{\Gamma\left(\hat{c}_{\alpha_i}\right) \hat{b}_{\alpha_i}^{\hat{c}_{\alpha_i}}}{\Gamma\left(c_{\alpha_i}\right) b_{\alpha_i}^{c_{\alpha_i}}} = 0$ for exponential sources as ARD is not used.

Appendix E

Derivations for vbMoICA

E.1 Mixture Variables

$p'(\boldsymbol{\kappa})$

The posterior over the indicator priors $\boldsymbol{\kappa}$ is given by

$$\log p'(\boldsymbol{\kappa}) \propto \langle \log p(\boldsymbol{c}|\boldsymbol{\kappa}) \rangle_{p'(\boldsymbol{C})} + \log p(\boldsymbol{\kappa}) \tag{E.1}$$

Substituting (6.10) and (6.11) into (E.1) gives

$$\begin{aligned} \log p'(\boldsymbol{\kappa}) &\propto \sum_{c=1}^{C} \langle \log \kappa_c \rangle_{p'(\boldsymbol{C})} + \sum_{c=1}^{C} (\iota_0 - 1) \log \kappa_c \\ &\propto \sum_{c=1}^{m} \left[\left(\iota_0 + \sum_{t=1}^{T} \hat{\eta}_c^t \right) - 1 \right] \log \kappa_c \end{aligned} \tag{E.2}$$

The functional form of (E.2) implies a non-symmetric Dirichlet for $p'(\boldsymbol{\kappa})$

$$p'(\boldsymbol{\kappa}) = \Gamma(\sum_{c'} \hat{\iota}_{c'}) \prod_{c=1}^{C} \frac{\kappa_c^{\hat{\iota}_c - 1}}{\Gamma(\hat{\iota}_c)} \tag{E.3}$$

where

$$\hat{\iota}_c = \iota_0 + \sum_{t=1}^{T} \hat{\eta}_c^t \tag{E.4}$$

$p'(\boldsymbol{c})$

The posterior over the indicator variable c is given by

$$\log p'(\boldsymbol{c}) \propto \langle \log p(\boldsymbol{X}|\boldsymbol{\Theta}_c, \boldsymbol{c}) \rangle_{p'(\boldsymbol{\Theta}_c)} + \langle \log p(\boldsymbol{c}|\boldsymbol{\kappa}) \rangle_{p'(\boldsymbol{\kappa})} \tag{E.5}$$

Substituting the appropriate expressions into (E.5) gives

$$\begin{aligned}\log p'(\boldsymbol{c}) &\propto \sum_{t=1}^{T} \frac{1}{2}\langle \log \boldsymbol{\Lambda}_c \rangle - \frac{\langle \boldsymbol{\Lambda}_c \rangle}{2} \langle \left(x^t - \hat{x}^t - y_c\right)^2 | c \rangle \\ &+ \sum_{t=1}^{T} \langle \log \kappa_c \rangle \end{aligned} \tag{E.6}$$

Exponentiating (E.6) yields the posterior over $\boldsymbol{c}$

$$p'(\boldsymbol{c}) = \prod_{t=1}^{T} \hat{\eta}_c^t \tag{E.7}$$

where

$$\eta_c^t = \tilde{\kappa}_c \tilde{\boldsymbol{\Lambda}}_c^{\frac{M}{2}} \exp\left[-\frac{\langle \boldsymbol{\Lambda}_c \rangle}{2} \langle \left(x^t - \hat{x}^t - y_c\right)^2 | c \rangle \right] \tag{E.8}$$

$$\hat{\eta}_c^t = \frac{\eta_c^t}{\sum_{c'} \eta_{c'}^t} \tag{E.9}$$

E.2 Energy

The NFE of vbMoICA is given by (6.19)

$$F_{\text{tot}} = F_{\text{mixture}} + \sum_{c=1}^{C} F_{ICA_c} \tag{E.10}$$

where

$$F_{\text{mixture}} = \left[\sum_{c=1}^{C_i} \log \frac{\Gamma\left(\hat{\iota}_c\right)}{\Gamma\left(\iota_0\right)} - \log \frac{\Gamma\left(\sum_{c'} \hat{\iota}_c\right)}{\Gamma\left(C\iota_0\right)} \right] - \sum_{t=1}^{T} \sum_{c=1}^{C} \hat{\eta}_c^t \log \hat{\eta}_c^t \tag{E.11}$$

and where F_{ICA_c} is the NFE for vbICA2 (including ARD) with

$$F_{\text{nse}_c} = \log \frac{\Gamma\left(\hat{c}_{\Lambda_c}\right) \hat{b}_{\Lambda_c}^{\hat{c}_{\Lambda_c}}}{\Gamma\left(c_{\Lambda_c}\right) b_{\Lambda_c}^{c_{\Lambda_c}}} - \frac{M}{2} \log 2\pi \sum_{t=1}^{T} \hat{\eta}_c^t \tag{E.12}$$

and with a contribution from the bias terms, $\{\boldsymbol{y}_c\}$

$$F_{\text{bias}_c} = -\frac{1}{2} \sum_{j=1}^{M} \left(\frac{\tau_{y_j}}{\hat{\tau}_{y_j}} - 1 \right) - \log \frac{\tau_{y_j}}{\hat{\tau}_{y_j}} + \tau_{y_j} \hat{m}_{y_j}^2 \tag{E.13}$$

Appendix F

Derivations for vbHMM

F.1 HMM Variables

$p'(\boldsymbol{q})$

The posterior over the indicator variables $\boldsymbol{q}$ is given by

$$\log p'(\boldsymbol{q}) \propto \langle \log p(\boldsymbol{s}|\boldsymbol{q}, \boldsymbol{\theta q}) \rangle_{p'(\boldsymbol{\theta q})} + \langle \log p(\boldsymbol{q}|\boldsymbol{\pi}, \boldsymbol{R}) \rangle_{p'(\boldsymbol{\pi})p'(\boldsymbol{R})} \tag{F.1}$$

Substituting (7.5), (7.6) and (7.3) into (F.1) gives

$$\begin{aligned} \log p'(\boldsymbol{q}) &\propto \langle \log P(q^1|\boldsymbol{\pi}) \rangle_{p'(\boldsymbol{\pi})} + \sum_{t=2}^{T} \langle \log p(q^t|q^{t-1}, \boldsymbol{R}) \rangle_{p'(\boldsymbol{R})} \\ &+ \sum_{t=1}^{T} \langle \log p_{q^t}(s^t|\theta_{q^t}) \rangle_{p'(\theta_{q^t})} \\ &\propto \langle \log \boldsymbol{\pi} \rangle + \sum_{t=2}^{T} \langle \log r_{q^{t-1}q^t} \rangle + \sum_{t=1}^{T} \langle \log p_{q^t}(s^t|\theta_{q^t}) \rangle \end{aligned} \tag{F.2}$$

Exponentiating yields the posterior over $\boldsymbol{q}$

$$p'(\boldsymbol{q}) = \frac{1}{Z_{\boldsymbol{q}}} \left[\tilde{\pi}_{q^1} \prod_{t=2}^{T} \tilde{r}_{q^{t-1}q^t} \prod_{t=1}^{T} \tilde{p}_{q^t} \right] \tag{F.3}$$

where

$$\begin{aligned} Z_{\boldsymbol{q}} &= \sum_{\{\boldsymbol{q}\}} \tilde{\pi}_{q^1} \prod_{t=2}^{T} \tilde{r}_{q^{t-1}q^t} \prod_{t=1}^{T} \tilde{p}_{q^t} \\ \tilde{\pi}_{q^t} &= \exp \left[\Psi(\hat{\lambda}_{q^t}) - \Psi(\sum_{q^{t'}} \hat{\lambda}_{q^{t'}}) \right] \\ \tilde{r}_{cd} &= \exp \left[\Psi(\hat{\iota}_{cd}) - \Psi(\sum_{d'} \hat{\iota}_{cd'}) \right] \\ \tilde{p}_{q^t} &= \exp \left[\langle \log p_{q^t}(s^t|\theta_{q^t}) \rangle \right] \end{aligned} \tag{F.4}$$

$p'(\boldsymbol{R})$

The posterior over the transition matrix $\boldsymbol{R}$ is given by

$$\log p'(\boldsymbol{R}) \propto \langle \log p(\boldsymbol{q}|\boldsymbol{\pi}, \boldsymbol{R}) \rangle_{p'(\boldsymbol{\pi})p'(\boldsymbol{q}} + \log p(R) \tag{F.5}$$

Substituting (7.5), (7.6) and (7.1) into (F.1) gives

$$\begin{aligned} \log p'(\boldsymbol{R}) &\propto \sum_{t=2}^{T} \langle \log p(q^t|q^{t-1}, \boldsymbol{R}) \rangle_{p'(\boldsymbol{q})} + \log p(R) \\ &\propto \sum_{c=1}^{C}\sum_{d=1}^{C} \left(\sum_{t=1}^{T-1} \xi_{cd}^t + \iota_{c0} \right) \log r_{cd} \end{aligned} \tag{F.6}$$

The functional form of (F.6) implies a non-symmetric Dirichlet for $p'(\boldsymbol{R})$

$$p'(\boldsymbol{R}) = \prod_{c=1}^{C} \Gamma(\sum_{cd'} \hat{\iota}_{cd'}) \prod_{c=1}^{C} \frac{r_{cd}^{\hat{\iota}_{cd}-1}}{\Gamma(\hat{\iota}_{cd})} \tag{F.7}$$

where

$$\hat{\iota}_{cd} = \iota_{c0} + \sum_{t=1}^{T-1} \xi_{cd}^t \tag{F.8}$$

$p'(\boldsymbol{\pi})$

The posterior over the initial state vector $\boldsymbol{\pi}$ is given by

$$\log p'(\boldsymbol{\pi}) \propto \langle \log p(\boldsymbol{q}|\boldsymbol{\pi}, \boldsymbol{R}) \rangle_{p'(\boldsymbol{R})p'(\boldsymbol{q}} + \log p(\boldsymbol{\pi}) \tag{F.9}$$

Substituting (7.5), (7.6) and (7.2) into (F.1) gives

$$\begin{aligned} \log p'(\boldsymbol{\pi}) &\propto \sum_{c=1}^{C} \langle \log p(q^1 = q_c|\boldsymbol{\pi}) \rangle_{p'(\boldsymbol{q})} + \log p(\pi) \\ &\propto \sum_{c=1}^{C} \left(\gamma_c^1 + \lambda_0 \right) \log \pi_c \end{aligned} \tag{F.10}$$

The functional form of (F.10) implies a non-symmetric Dirichlet for $p'(\boldsymbol{\pi})$

$$p'(\boldsymbol{\pi}) = \Gamma(\sum_{c'} \hat{\lambda}_{c'}) \prod_{c=1}^{C} \frac{\pi_c^{\hat{\lambda}_c - 1}}{\Gamma(\hat{\lambda}_c)} \tag{F.11}$$

where

$$\hat{\lambda}_c = \lambda_0 + \gamma_c^1 \tag{F.12}$$

Averaging the above expressions over $p'(\boldsymbol{S})$ or $p'(\boldsymbol{S}|\boldsymbol{q})$ gives the updates for vbICA-HMM. Substituting (7.68) into (F.4) yields (7.69) for vbHMM-ICA.

F.2 Energy

The NFE of vbHMM is given by

$$F_{\text{tot}} = F_{\text{HMM}_{mix}} + \sum_{c=1}^{C} F_{\text{obs}_c} \tag{F.13}$$

The HMM part is given by

$$\begin{aligned} F_{\text{HMM}_{mix}} &= \sum_{c=1}^{C}\left[\sum_{d=1}^{C}\log\frac{\Gamma\left(\hat{\iota}_{cd}\right)}{\Gamma\left(\iota_{c0}\right)} - \log\frac{\Gamma\left(\sum_{d'}\hat{\iota}_{cd'}\right)}{\Gamma\left(C\iota_{c0}\right)}\right] \\ &+ \left[\sum_{c=1}^{C}\log\frac{\Gamma\left(\hat{\lambda}_c\right)}{\Gamma\left(\lambda_0\right)} - \log\frac{\Gamma\left(\sum_{c'}\hat{\lambda}_{c'}\right)}{\Gamma\left(C\lambda_0\right)}\right] \\ &- \left(\sum_{c=1}^{C}\gamma_c^1\log\gamma_c^1 + \sum_{t=1}^{T-1}\sum_{c=1}^{C}\sum_{d=1}^{C}\xi_{cd}^t\log\frac{\xi_{cd}^t}{\sum_{d'}\xi_{cd}^t}\right) \end{aligned} \tag{F.14}$$

The observation part F_{obs_c} depends on whether Gaussian or ICA generators are used. If ICA generators, then $F_{\text{obs}_c} = F_{\text{ICA}_c}$, detailed in section E.2. If Gaussian observation generators, then

$$F_{\text{obs}_c} = \sum_{c=1}^{C}\log\frac{\Gamma\left(\hat{c}_c\right)\hat{b}_c^{\hat{c}_c}}{\Gamma\left(c_0\right)b_0^{c_0}} \tag{F.15}$$

$$- \frac{1}{2}\sum_{c=1}^{C}\left[\frac{\tau_0}{\hat{\tau}_c} - \log\frac{\tau_0}{\hat{\tau}_c} + \tau_0\left(\hat{m}_c - m_0\right)^2\right] + \frac{1}{2} \tag{F.16}$$

Substituting (F.16) into (F.13) gives an expressions for the energy of HMM source i, F_{HMM_i}. This replaces F_{MoG_i} in (B.48) when ICA generators with HMM sources are used.