Infinite Independent Subspace Analysis

Krikamol Muandet
Author

Yee Whye Teh
Supervisor

This report is submitted as part requirement for the MSc Degree in Machine Learning at University College London. It is substantially the result of my own work except where explicitly indicated in the text.

September 2010

To my parents and my brother who have always encouraged and supported me and for whom I am very grateful.

Abstract

The infinite Independent Subspace Analysis (iISA) is presented in this thesis. This model is based on the nested Indian Buffet Process (nIBP), a stochastic process which assigns probability distributions to trees of infinite depth and branching factors. In Bayesian nonparametrics, the theoretical results of nesting strategies, particularly nIBP, are still lacking, so some interesting properties of nIBP are illustrated through various examples to gain insight further into the nIBP and related models such as nested Chinese restaurant process (nCRP). Using the nIBP, the iISA model eliminates the restrictions of classical ISA algorithms on the number of groups and groups sizes by allowing them to be inferred from the data. Moreover, the specialised inference algorithm based on the Metropolis-Hasting method is proposed to handle the increased complexity of the model. The experimental results have not only demonstrated the performance of the iISA model, but also led to a conceptual understanding that can be used to improve the model. Although the application of iISA model on the natural images was not successful, it provides some insights that can be used for further development.

Acknowledgements

I have been privileged to work with Yee Whye Teh, who has taught me so much in the past. Without his guidance this thesis would not have been possible.

Table of Contents

1 Introduction **2**

2 Indian Buffet Process **8**

2.1 Background 9

2.1.1 An infinite feature model 10

2.1.2 Indian buffet metaphor 12

2.1.3 A two-parameter extension 12

2.1.4 Completely random measure 13

2.1.5 The Bernoulli process 14

2.2 Hierarchical beta processes 14

2.3 Nested beta processes 15

3 Infinite Independent Subspace Analysis **22**

3.1 Independent component analysis 22

3.2 Independent subspace analysis 25

3.3 Infinite independent component analysis 26

3.4 Independent subspace analysis using nested IBP 27

4 Inference **32**

4.1 Likelihood 32

4.2 The hidden source $\mathbf{x}_t$ 33

4.3 The source dependency $\mathbf{v}_t$ 33

4.4 The mixing matrix $\mathbf{A}$ 34

4.5 The noise variance 35

4.6 The mixing matrix variance 35

4.7 Active sources . . . 35

4.8 Active subspaces . . . 36

4.9 Update the number of sources . . . 41

4.10 Update the number of subspaces . . . 44

4.11 IBP parameter α_j . . . 48

4.12 The relevant parameters . . . 48

5 Experiments and Analysis 50

5.1 Experimental setting . . . 50

5.2 Results . . . 53

5.2.1 Synthetic data . . . 53

5.2.2 Natural images . . . 62

5.3 Analysis . . . 64

6 Conclusions and Future Works 65

References 72

Appendices 74

A iISA Algorithm 74

A.1 Active subspaces . . . 75

A.2 Update number of sources . . . 76

A.3 Update number of subspaces . . . 77

A.4 MCMC sampler for iISA . . . 78

B Conditional Probabilities 79

B.1 The hidden source $\mathbf{x}_t$ . . . 79

B.2 The source dependency $\mathbf{v}_t$ . . . 80

B.3 The mixing matrix $\mathbf{A}$ . . . 81

B.4 The noise variance σ_ϵ^2 . . . 82

B.5 The variance of mixing matrix elements . . . 82

List of Figures

2.1 The binary matrix $\mathbf{Z}$ and its left-order form 11

2.2 The graph and binary matrices generated by the nIBP 18

2.3 The distributions over paths and sub-trees 19

2.4 The trees sampled from nIBP with different values of $\{\alpha_{ij}, \beta_{ij}\}$ 20

2.5 The different distributions over α and β 21

3.1 The graphical model of iICA model 27

3.2 The neural network view of the iISA 29

3.3 The graphical model for the finite version of iISA model 30

4.1 The relation of binary matrices in the two-level nested IBP 35

4.2 Two sets of columns of binary matrix $\mathbf{Z}_j$ 41

4.3 Proposing a new subspace 44

5.1 The true and inferred synthetic matrix $\mathbf{Z}$ with $K = 7$ hidden sources . . 53

5.2 The trace plots of iICA model on synthetic data 54

5.3 The true matrices (left) and inferred versions (right) of $\mathbf{U}$, $\mathbf{Z}_1$, and $\mathbf{Z}_2$. 54

5.4 The trace plots of the inference algorithm for iISA model 55

5.5 The comparisons of FastICA, ISA, iICA, and iISA 56

5.6 The subspace sizes of synthetic data sets 57

5.7 The trace plots of iISA on four synthetic data 59

5.8 the generalised Amari errors of different iISA variants 61

5.9 The examples of 160 images patches extracted from the natural images . 62

5.10 The 160 filters obtained from the ICA algorithm 63

5.11 The 160 filters obtained from the ISA algorithm 63

List of Tables

4.1 The parameters for MCMC sampler of the iISA model 49

5.1 The time per iteration in seconds for each algorithm 58

Chapter 1

Introduction

Since the invention of the computer there have been attempts to develop computers as general programmable machines, that can be adapted to solve problems requiring a considerable amount of computation, such as finance and engineering. Although at first it might seem difficult to believe that machines can, for example, play chess or analyse the human genome, the introduction of new computers that are faster, more flexible, and more efficient makes the aforementioned applications become more feasible. Nowadays, computers have become a crucial factor in the achievement of many practical applications, not only in scientific discovery, but also in various industries.

Due to the emergence of many practical applications, most problems become more complicated and require not only higher computational power but also an efficient algorithms to solve problems effectively. Machine learning has emerged as one of the scientific disciplines that aims to gain more flexibility in problem-solving by allowing the computer to automatically learn to recognise complex patterns or relationship in data so that optimal solutions can be found without hard-coding. Due to its promising performance, machine learning has been adopted in many practical uses, ranging from the problems as simple as calculators to problems as complicated as brain image analysis and the human genome project.

Although various machine learning techniques have proved successful in practice, the performance of these techniques depends largely on the choice of the models given the available data. Flexibility of these learning techniques can be gained by adjusting the values of the associated parameters. Consider the clustering problem in which the data are arranged into different clusters and we attempt to identify which data belongs to which cluster. Most clustering algorithms, for example, K-means, allows the specification of the number of clusters K before applying the algorithm to the existing data. Choosing the appropriate values of parameters in the learning model is generally regarded as model selection, which often relies on the available data. Although allowing the model to be adjusted according to the data, this scenario has some drawbacks.

Firstly, finding the right model for the available data set using model selection can be difficult and considerably computational demanding for many learning algorithms. Cross validation, for instance, is usually used to perform model selection. In cross validation, the data are separated into multiple subsets, each of which will be used to evaluate the performance of the model trained using the rest of the data. Since this involves multiple runs of training algorithms and usually requires large amounts of data, cross validation is not desirable when complex models are needed and few data are available.

Secondly, even with the simple model such as K-means, it is usually difficult to come up with the suitable value of parameters, for example, the number of clusters by just looking at the data, especially a complex and high-dimensional data. Although various preprocessing techniques can be employed to have a preliminary insight of the data, these techniques are limited in that they can only capture some parts of the data and ignore many details that might be useful. As a result, the final choice of the model cannot be based entirely on these techniques.

Another drawback of such a parametric form of the model mentioned earlier is that they do not allow the size of the model to grow as we observe more data. That is, once we choose a specific model for the available data set, it stays fixed throughout the learning process. In a clustering problem, for instance, once the number of clusters K is specified, it will not change even if more data is gathered. The model is not able to cope with the increasing number of data. In other words, the complexity of the model is fixed and is not able to grow with the data size. This is not the case in many situations, e.g., we would expect an increase in the number of clusters if more data are observed.

From the probabilistic perspective, more expressive probabilistic representations to overcome the aforementioned limitations of the parametric models are needed. The conventional probabilistic modelling such as graphical models, which defines the probability distribution over the simple fixed-dimensional random variables, is no longer sufficient. The flexible probabilistic representations that allow us to work with more general objects such as lists, trees, and partitions need to be considered. Furthermore, for the effective exploitation of this flexibility, these novel representations should also allow easy-to-derive efficient learning algorithms.

In fact, the framework of the Bayesian nonparametric model provides this kind of flexibility. In this framework, the prior and posterior are no longer restricted to the probability distributions, but are general stochastic processes. The stochastic processes can be defined as an indexed collection of random variables, where the index set is allowed to be infinite. Moving from simple probability distributions to general stochastic processes provides the possibility to manipulate the objects in infinite-dimensional space. Consider the case when the index set is the continuous random vectors $x \in \mathbb{R}^d$. If we further assume that the output of a random function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ for any finite subset of input points is normally distributed, this is a well-known Gaussian process (GP) (Rasmussen and Williams, 2005). Note that the stochastic process is defined as a collection of random variables $\{f(x)\}$ for each possible value of x. In other words, we can directly define a prior over a space of functions using the GP. Together with the well-defined likelihood, the posterior distribution over random functions can be easily obtained via a standard Bayesian methods. This example demonstrates the possibility to update the prior stochastic process into the posterior stochastic process in inference just as we do with the parametric distributions used in classical Bayesian analysis.

An important basis for the prospect of Bayesian nonparametric models is the concept of exchangeability introduced by Bruno de Finetti. De Finetti's theorem (Aldous, 1985) states that an infinite sequence of random variables $(X_1, X_2, ...)$ is said to be infinitely exchangeable if for any finite cardinal number N and permutation π, we have $p(x_1, x_2, ..., x_N) = p(x_{\pi(1)}, x_{\pi(2)}, ..., x_{\pi(N)})$, where p is a probability mass function. It is equivalent to say that any finite subsets of those random variables is invariant to permutation. Formally, the general De Finetti's theorem states $(X_1, X_2, ...)$ are infinitely exchangeable if and only if there exist a parametric model $p(x_i|\boldsymbol{\theta})$ for some parameter $\boldsymbol{\theta}$ and a probability distribution for $\boldsymbol{\theta}$ with a density $p(\boldsymbol{\theta})$ such that the joint density of any subsets of those random variables can be written as

$$p(x_1, x_2, ..., x_N) = \int \prod_{i=1}^{N} p(x_i|\boldsymbol{\theta}) p(\boldsymbol{\theta}) d(\boldsymbol{\theta}) \quad . \tag{1.1}$$

It follows from Equation (1.1) that the joint distribution $p(x_1, x_2, ..., x_N)$ is invariant to permutation as the order of random variables does not affect the value of the joint distribution. A more intriguing interpretation of this theorem states that if observations are exchangeable, there must exist an underlying parameter and a corresponding prior probability distribution, such that these observations are indeed a random sample from some models specified by this parameter. In this context, there is no restriction that $\boldsymbol{\theta}$ should be a finite-dimensional object. Thus, by considering the parameter $\boldsymbol{\theta}$ as an infinite-dimensional object and p as a stochastic process, it clearly follows that the inference over infinite-dimensional objects is amenable to the Bayesian approach.

This work focuses on one of the nonparametric Bayesian models known as the Indian Buffet Process (IBP), the goal of which is initially to provide a flexible probabilistic modelling for sparse latent feature models (Griffiths and Ghahramani, 2005a). Unlike the probabilistic mixture model, each data point can be associated with multiple binary latent variables. In IBP, there can potentially be an infinite number of latent features. This representation is appropriate when there are multiple overlapping clusters or each data point may simultaneously belong to several clusters. In the same way as GPs induce the distribution over functions, IBP provides the distribution over the binary matrices where rows are data points and columns are latent features. The number of columns may potentially be unbounded, which allows more flexibility in inferring the latent features from the data. Thus the IBP has been used successfully to extend some parametric models such as factor analysis (Paisley and Carin, 2009) and independent component analysis (Knowles and Ghahramani, 2007). Moreover, the success of IBP has also been demonstrated in diverse fields including binary matrix factorisation (Meeds et al., 2006), cognitive science (Austerweil and Griffiths, 2008; Navarro and Griffiths, 2008), and bioinformatics (Chu et al., 2006).

There have been attempts to develop more sophisticated nonparametric Bayesian models on top of IBPs so that the complex structure underlying the data can be captured more effectively. In Doshi-Velez and Ghahramani (2009), for instance, the authors propose a generalisation of IBP that is able to model the correlated latent features. The phylogenetic IBP (Miller et al., 2008) provides the non-exchangeable prior for infinite latent feature model in which the prior knowledge about the relationship of objects using a tree structure can be incorporated. A three-parameter extension of IBP (Teh and Görür, 2009) exhibiting power-law behavior can be used to model word occurrences in document corpora. Additionally, the hierarchical modelling (Teh and Jordan, 2010) is a major extension of nonparametric Bayesian models to coupling a set of parameters through a shared underlying parameters, a basis for many applications (Courville et al., 2009; Rai and Daumé III, 2008; Teh et al., 2006; Thibaux and Jordan, 2007). In accordance with the general Bayesian nonparametric framework, the introduction of stick-breaking representation (Teh et al., 2007) for the IBP, as well as the beta process (Thibaux and Jordan, 2007) as the de Finetti mixing distribution underlying the IBP establishes the connections to other Bayesian nonparametric models.

Nested modelling, unlike hierarchical modelling, allows more complex models to be built from simpler components. In this thesis, the nested Indian buffet process (nIBP) is explored, which is closely related to the nested Chinese restaurant process (nCRP) (Blei et al., 2010). Both of them specify a generative probabilistic model for tree structure. Unlike the nCRP, in which each object corresponds to a single path down the tree, the nIBP allows each object to possess multiple paths constituting a subtree. Since few works focus on nested models, particularly nIBP, one of the goals of this thesis is to study the nesting strategy in the nonparametric Bayesian framework. various characteristics of nIBP and explicit comparisons to nCRP are discussed. By understanding its theoretical properties, we expect to draw more attentions to nIBP, leading to the development of potential applications using this model.

As an example, we also develop the infinite Independent Subspace Analysis (iISA) to illustrate through a particular application how nIBP model can be applied in practice and explore the strengths and weaknesses of nIBP. Many variants of Independent Subspace Analysis (ISA) have been developed in the context of blind source separation problems. Unlike Independent Component Analysis (ICA), which assumes that all hidden sources are mutually independent, the ISA mitigates this constraint by allowing some hidden sources to be dependent. The independence assumption is thus imposed on the groups of dependent sources (subspaces) instead of individual sources. However, in traditional ISA models, the number of subspaces and the subspace dimensions must be specified. Additionally, it is not clear how to generalise the ISA models to instances when subspace dimensions are not equal. By using the nIBP, the iISA model instead allows the number of subspaces and their dimensions to be unbounded. The optimal setting can then be inferred from the data.

The remaining of this thesis is laid out as follows:

Chapter 2 *Indian buffet process*

An overview of the IBP, including a discussion of its relevant extensions.

Chapter 3 *Infinite independent subspace analysis*

A discussion of blind source separation problems that can be solved using independent component analysis and independent subspace analysis. This includes a review of the infinite ICA model and an introduction to the nonparametric version of ISA model called the iISA based on the nIBP, which is the main contribution of this thesis.

Chapter 4 *Inference*

The derivation of the conditional probabilities of variables in the infinite ISA model used for inference, including a proposal for the use of the Markov chain Monte Carlo (MCMC) method for this model.

Chapter 5 *Experiments and analysis*

The experimental results on both synthetic and real-world data, including the problems found in the experiments, and the advantages and disadvantages of the model.

Chapter 6 *Conclusions and future works*

Conclusions of the thesis and a discussion of the possibilities to improve the proposed model, including further applications of the nIBP.

Chapter 2

Indian Buffet Process

The Indian buffet process was first proposed in Griffiths and Ghahramani (2005a) for infinite latent feature models, which is then extended to a two-parameter family in Ghahramani et al. (2007). Then in Thibaux and Jordan (2007), the beta process is shown to be the underlying de Finetti mixing distribution of the IBP. Teh et al. (2007) derives a stick-breaking construction and develops an efficient slice sampling inference algorithm for the IBP. The IBP has been successfully applied in various applications, some of which include inferring hidden causes (Wood et al., 2006), choice models (Görür et al., 2006), modelling dyadic data (Meeds et al., 2006), modelling the sparsity structure in the latent variables (Knowles and Ghahramani, 2007), overlapping clustering (Heller and Ghahramani, 2007), similarity judgement (Navarro and Griffiths, 2008), matrix factorisation (Wood and Griffiths, 2006), and link prediction in relational data (Miller et al., 2009).

The IBP relies on the assumptions of independence between latent features and exchangeability of samples, which limits its uses in many practical domains. There have been many attempts to extend the capability of IBP in order to overcome this limitation. Doshi-Velez and Ghahramani (2009) generalise the IBP for correlated latent features by imposing the hierarchical structures. A similar idea is also proposed by Courville et al. (2009). In this work, the authors introduce the unbounded layers of latent factors where the correlations between latent factors in the layer below are induced via the noisy-or mechanism. In these hierarchical models, correlations reflect a higher layer of structure. A recent work also extends the IBP for correlated observations that are non-exchangeable. The phylogenetic IBP (Miller et al., 2008) models the similarities among observations by a tree, resulting in a non-exchangeable prior. In this model, the sharing of features between two samples is driven by not only the popularity of features, but also the similarity of those samples.

Following several theoretical studies and practical uses of a hierarchical nonparametric Bayesian model, a nested model is one extension that has not been extensively studied. The nested model was first studied in the context of Dirichlet process. Abel et al. (2008) proposed the nested Dirichlet process, first used in multicenter studies. With the nested Dirichlet process, data in each center and the centers themselves can be simultaneously clustered by borrowing information across centers. A similar idea is adopted in multi-task learning for the infinite Hidden Markov Model (iHMM) (Ni et al., 2007). Imposing a nested Dirichlet process prior on the base distributions enables the simultaneous performance of both task-level clustering and data-level clustering. Another interesting extension of nonparametric Bayesian models is the nCRP (Blei et al., 2003, 2010), which is proposed in the context of topics modelling. The nCRP is used as a prior on trees with infinite depth and infinite branching factors that represent the topic hierarchies of document collections. As the amount of literature indicates, there has been much works, both in theory and practical applications, particularly compared to those in the hierarchical nonparametric Bayesian models. It is therefore very beneficial to further investigate the theoretical properties and potential practical applications of the nIBP.

2.1 Background

The Indian buffet process (IBP) provides a powerful approach to defining the factorial representation in probabilistic modelling. As opposed to the multinomial representation used in classical mixture models, the factorial representation provides each data point a set of binary latent variables. A movie, for instance, may be described by a set of genres, such as action and comedy. Since a single movie can be categorised into multiple genres, each movie is associated with the binary-valued vector whose elements indicate which genres the movie belongs to, that is, the element is 1 if the movie is in the corresponding genre and 0 otherwise. If the maximum number of active elements is 1, this will correspond to the multinomial representation where each movie is assumed to be in only one genre. Therefore, in addition to a natural interpretation of featural description of the objects, factorial representation also provides a way to define the similarity between objects by looking at the number of features that two objects have in common. Additionally, the factorial representation is more powerful than multinomial representation, as the total number of clusters induced by the factorial representation is 2^K, where K is the number of binary latent variables.

2.1.1 An infinite feature model

In this section, a parametric model with a finite number of features K is defined. The nonparametric model with infinite number of features is then derived by taking $K \rightarrow \infty$. Assume that there are K features and the possession of feature k by object i is indicated by a binary variable z_{ik}. Each object possesses feature k with a probability of π_k and whose features are generated independently. The binary variable z_{ik} is generated according to the following generative process:

$$\pi_k \sim \mathrm{Beta}(\frac{\alpha}{K}, 1) \tag{2.1}$$

$$z_{ik}|\pi_k \sim \mathrm{Bernoulli}(\pi_k) \tag{2.2}$$

Given N data points, this process will generate a binary $N \times K$ matrix $\mathbf{Z}$ whose rows and columns correspond to the data points and features, respectively. Since features are generated independently, the probability of matrix $\mathbf{Z}$ given the vector $\boldsymbol{\pi}$ is

$$\begin{aligned} P(\mathbf{Z}|\boldsymbol{\pi}) &= \prod_{k=1}^{K}\prod_{i=1}^{N} P(z_{ik}|\pi_k) && (2.3) \\ &= \prod_{k=1}^{K} \pi_k^{m_k}(1-\pi_k)^{N-m_k} && (2.4) \end{aligned}$$

where $m_k = \sum_{i=1}^{N} z_{ik}$ is the number of objects that have feature k. The marginal probability of a binary matrix $\mathbf{Z}$ can be obtained from Equation (2.4) by integrating over all values for $\boldsymbol{\pi}$.

$$\begin{aligned} P(\mathbf{Z}) &= \prod_{k=1}^{K} \int \left(\prod_{i=1}^{N} P(z_{ik}|\pi_k) \right) P(\pi_k) d\pi_k && (2.5) \\ &= \prod_{k=1}^{K} \frac{\mathrm{B}(m_k + \frac{\alpha}{K}, N - m_k + 1)}{\mathrm{B}(\frac{\alpha}{K}, 1)} && (2.6) \\ &= \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(\frac{\alpha}{K} + N + 1)} && (2.7) \end{aligned}$$

The distribution $P(\mathbf{Z})$ is clearly exchangeable as it depends on only the count m_k.

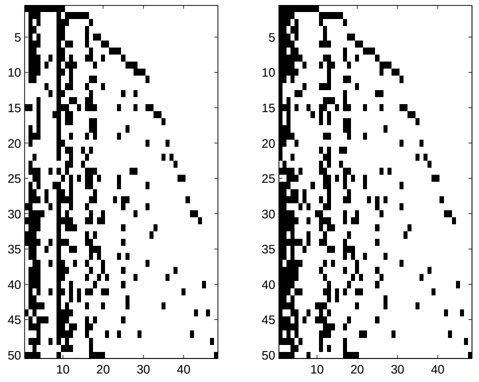


Figure 2.1 – The binary matrix **Z** and its left-order form

Instead of defining the distribution over binary matrices, it is more natural to define the distribution on the equivalence classes of binary matrices which have the same left-order form. The left-order form of matrix **Z** is denoted by $lof(\mathbf{Z})$ and can be obtained by treating columns of the binary matrix **Z** as binary numbers with the first row as the most significant bit and ordering them from left to right by their magnitudes. That is, all binary matrices in the same equivalence class correspond to the same left-order form. Figure 2.1 shows the binary matrix and its left-order form. The notion of the equivalent classes defined by the left-order form is beneficial to inference since the columns of the binary matrix **Z** and other related objects can be rearranged without affecting the result.

To facilitate the derivation of infinite cases, the history of feature k is defined using the decimal values expressed by the columns of **Z**. At object i features can have one of 2^{i-1} histories. Denote by K_h the number of features possessing the history h, with K_0 being the number of features for which $m_k = 0$ and $K_+ = \sum_{h=1}^{2^N-1} K_h$ being the number of features for which $m_k > 0$. Following (Griffiths and Ghahramani, 2005a), the probability distribution over the equivalence classes of binary matrices, [**Z**], is

$$\begin{aligned} P([\mathbf{Z}]) &= \sum_{\mathbf{Z}\in[\mathbf{Z}]} P(\mathbf{Z}) \\ &= \frac{K!}{\prod_{h=0}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{(\alpha/K)(m_k + (\alpha/K))\Gamma(N - m_k + 1)}{\Gamma(N + 1 + (\alpha/K))} \end{aligned} \tag{2.8}$$

By taking the infinite limit $K \to \infty$, the distribution defined in Equation (2.8) becomes

$$\lim_{K\to\infty} P([\mathbf{Z}]) = \frac{\alpha^{K_+}}{\prod_{h=1}^{2^N-1} K_h!} \exp\big\{ -\alpha H_N \big\} \prod_{k=1}^{K_+} \frac{(N - m_k)!(m_k - 1)!}{N!} \tag{2.9}$$

where H_N is the Nth harmonic number, $H_N = \sum_{j=1}^{N} \frac{1}{j}$. This distribution is again exchangeable as it depends on data only through the count m_k. The details of the derivation of this distribution can be found in (Griffiths and Ghahramani, 2005b).

2.1.2 Indian buffet metaphor

The process that generates the binary matrix **Z** from IBP can alternatively be understood using the Indian buffet metaphor. In the Indian buffet restaurant, it is assumed that there is an infinite number of dishes. N customers enter the restaurant one after another. The first customer chooses $\mathrm{Poisson}(\alpha)$ number of dishes. Given the history of customers for each dish, the probability that the subsequent customers will choose a particular dish is proportional to the number of customers that have previously chosen that dish. Thus the i-th customer chooses dish k with a probability of $\frac{m_k}{i}$, where m_k is the number of previous customers who sample dish k. In addition to the dishes that have already been sampled, the customer also tries a $\mathrm{Poisson}(\frac{\alpha}{i})$ number of new dishes.

If each row and column of the binary matrix **Z** is the customer and the dish, respectively, the above generative process will produce the sparse binary matrix according to IBP. Nevertheless, neither the rows nor the columns of matrix **Z** are exchangeable. The inference can be performed by treating the current data point as the last customer in the buffet.

2.1.3 A two-parameter extension

According to Equation (2.2), the distribution of the number of features is characterised by α. Specifically, the distribution on the total number of features and the number of features per object are coupled through α. To remove this undesirable constraint, the two-parameter extension of IBP is proposed in Ghahramani et al. (2007), which can be defined as follows:

$$\pi_k \sim \mathrm{Beta}(\frac{\alpha\beta}{K}, \beta) \tag{2.10}$$

$$z_{ik}|\pi_k \sim \mathrm{Bernoulli}(\pi_k) \tag{2.11}$$

A two-parameter generalisation allows the average number of features per object and the total number of features in a set of N objects to be tuned independently. The average number of features per object is characterised by α. Using β, the model can be defined such that the overall number of features can range from α, where all objects share the same features, to $N\alpha$, where no features are shared at all. Thus we can tune the overall number of features while maintaining the average number of features per object at α.

To see how β affects the overall number of features, that is, the number K_+ of k with $m_k > 0$, consider the expected value of the overall number of features. One obtains directly from the buffet interpretation that the distribution of K_+ is Poisson with mean $\overline{K}_+ = \alpha \sum_{i=1}^{N} \frac{\beta}{\beta+i-1}$. The expected number of features used therefore increases as β increases. For a fixed number of objects N, one can see that $\overline{K}_+ \to N\alpha$ as $\beta \to \infty$. In this limit, no features are shared among N objects. Conversely, $\overline{K}_+ \to \alpha$ as $\beta \to 0$, in which all objects share the same features as the first one. As a result, β affects the feature vector variance. At small values of β, objects tend to share more features, so the feature vector variance is low. However, at high values of β, the probability of two objects possessing the same feature is low, which results in high feature vector variance.

2.1.4 Completely random measure

Consider a measure space $(\Omega, \mathcal{A})$, where Ω is a measurable space and $\mathcal{A}$ a σ-algebra. A random measure μ over $(\Omega, \mathcal{A})$ is a stochastic process whose index set is $\mathcal{A}$. For any set $A \in \mathcal{A}$, $\mu(A)$ is the random variable. A completely random measure μ is a random measure such that $\mu(A)$ and $\mu(B)$ are independent for any disjointed subsets $A, B \in \mathcal{A}$.

Consider a simple way to construct a completely random measure from the non-homogeneous Poisson process (Kingman, 1967). Given a product space $(\Omega, \mathbb{R})$ and a σ-finite product measure η, which is treated as the rate measure for a non-homogeneous Poisson process, draw a sample $\{(\omega_i, p_i)\}$ from this Poisson process. Next, a measure G on Ω is formed using this sample, as follows:

$$G = \sum_{i=1}^{\infty} p_i \delta_{\omega_i} \tag{2.12}$$

where $\{\omega_i\}$ is the atoms and $\{p_i\}$ is the weight associated to each atom. Related to the IBP is a completely random measure called beta process. The rate measure η of beta process is a product of an arbitrary σ-finite measure B_0 on Ω and an improper beta distribution on $(0, 1)$, which can be defined for $c > 0$ as $\eta(d\omega, dp) = cp^{-1}(1-p)^{c-1} dp B_0(d\omega)$. The resulting completely random measure thus can be written as

$$B = \sum_{i=1}^{\infty} p_i \delta_{\omega_i} \tag{2.13}$$

where $p_i \in (0,1)$. The random measure B is said to be drawn from the beta process with a concentration parameter $c > 0$ and a base measure B_0 if it is generated as

$$B \sim \mathrm{BP}(c, B_0) \tag{2.14}$$

From Equation (2.13), a draw from the beta process is simply a collection of atoms in Ω, each of which is associated with a weight p_i drawn from a beta distribution whose parameters are defined by the concentration parameter c and the base measure B_0.

2.1.5 The Bernoulli process

It is natural to see a draw B from beta process as a collection of coin-tossing probabilities. If coins are tossed independently, the following completely random measure is obtained:

$$Z = \sum_{i=1}^{\infty} b_i \delta_{\omega_i} \tag{2.15}$$

where b_i are independent Bernoulli variables with the probability that $b_i = 1$ is p_i. It is possible to see Ω as a set of potential features. A draw from the beta process then encodes the probability for each feature to be active. Because η has infinite mass near zero, infinitely many atoms in Ω will be assigned small weights. Therefore, only the finite number of features will be active in a draw from the Bernoulli process. Using this formula, we define a draw Z from Bernoulli process as follows:

$$\begin{aligned} B &\sim \mathrm{BP}(B_0) & (2.16)\\ Z|B &\sim \mathrm{BeP}(B) & (2.17) \end{aligned}$$

where B_0 is the base measure defining the initial set of atoms. The probability p_i associated with each atom in a draw B from the beta process can be considered as π_i defined in the IBP. This shows the connection between IBP and Beta-Bernoulli processes. Indeed, the beta process is shown to be the underlying de Finetti mixing distribution of IBP (Thibaux and Jordan, 2007).

2.2 Hierarchical beta processes

Hierarchical modelling has been used within the framework of Bayesian nonparametrics (Courville et al., 2009; Cowans, 2004, 2006; Doshi-Velez and Ghahramani, 2009; Teh et al., 2006; Thibaux and Jordan, 2007; Xing et al., 2006). In classical parametric models, it is possible to think of hierarchical modelling as a way to build hierarchies on finite-dimensional parameters. Bayesian nonparametric models also incorporate these parameters and thus hierarchies can be built. In the framework of Bayesian nonparametrics, however, it is more common to build the hierarchies of the infinite-dimensional parameters of nonparametric models as it results in increased flexibility and ease of interpretation of the models.

The fundamental elements that form the basis of the hierarchical models are the completely random measures discussed in Section 2.1.4. Specifically, the set of completely random measures $\{G_1, G_2, ..., G_M\}$ are conditionally independent given a base measure G_0, which is itself a completely random measure. For instance, the Hierarchical Dirichlet Process (HDP) (Teh et al., 2006) can be used to model related groups of data simultaneously, where each group is to be modelled as a DP mixture. Since these groups of data are related, it is useful to couple the underlying random measures through the hierarchies. In Hierarchical Beta Process (HBP) (Thibaux and Jordan, 2007), the underlying random measures are simply the beta processes.

The HBP can be defined as follows:

$$\begin{aligned} B &\sim \mathrm{BP}(c_0, B_0) \\ B_j &\sim \mathrm{BP}(c, B) \qquad \forall j \leq M \\ Z_{ji} &\sim \mathrm{BeP}(B_j) \qquad \forall i \leq M_j \end{aligned} \tag{2.18}$$

where M is the number of groups and there are M_j individuals in group j. Consider when the measure B_0 is discrete. A draw B from the beta process has atoms at the same locations as B_0. As the random measure, B_j, for each group shares the same base measure, B, which renders all B_j to be related to each other, there are some overlaps among the atoms chosen in the different groups. The degree of coupling between groups, i.e., between B_j, is controlled by hyper-parameter c. For larger values of c, B_j are closer to B and thereby have more atoms in common.

2.3 Nested beta processes

Hierarchical models in Bayesian nonparametric couple multiple random measures through the common base measure, providing a way to share atoms among random measures. This gives the flexibility to model the related groups of data. It is also worthwhile to consider the nested structure when atoms are separated into non-interacting groups. In this case, the random measures have a unique set of atoms and thereby do not share atoms with other random measures. Nesting strategies allow the complex model to be decomposed into several simpler components which may be solved more easily. Lack of coupling among random measures simplifies the inference algorithms and allows the use of the divide-and-conquer strategy in the inference.

The first model that makes use of the nesting strategy is the nested Dirichlet process (nDP) (Abel et al., 2008). In this model, the collection of distributions $\{G_1, ..., G_J\}$ is said to follow the nDP if $G_j \overset{\text{iid}}{\sim} Q$ with $Q \sim \text{DP}(\alpha\text{DP}(\beta H))$, which can be thought of as a DP where the base measure is also a DP. By imposing the nesting structure, each draw from nDP is a stochastic process. Due to the discrete property of DP, with probability one, some of the stochastic processes drawn from nDP are identical. Therefore, the nDP induces the clustering of stochastic processes. If each stochastic process is used to model the data, it will also induce the clustering of the data as the stochastic process is simply a DP. As a result, the nDP induces the clustering of groups of data as well as the data in each group simultaneously.

Specifically, the two-level nDP can be written in the following forms:

$$\begin{aligned} G &\sim \sum_{k=1}^{\infty} \pi_k^* \delta_{G_k^*} \\ G_k^* &= \sum_{j=1}^{\infty} \pi_{kj} \delta_{\theta_{kj}} \end{aligned} \tag{2.19}$$

where the weights $\{\pi_{kj}\}$ and $\{\pi_k^*\}$ are defined as in Equation (2.12). From Equation (2.19) it can be seen that G is a draw from DP where the atoms in the measure space are the DPs, G_k^*. That is, it is selected among an infinite set of components $\{G_k^*\}$. Moreover, the atoms associated with lower-level DPs are distinct. Although the model defined in Equation (2.19) considers only two levels of DPs, the nDP can also be extended to an arbitrary number of levels.

Abel et al applied the nDP in the multicenter studies. With the nDP, data in each center and the centers themselves can be clustered simultaneously by borrowing information across centers. A similar idea is adopted in multi-task learning for infinite Hidden Markov Model (iHMM) (Ni et al., 2007). Imposing the nDP prior on the base distributions allows task-level and data-level clustering to be performed simultaneously. In contrast to the hierarchical DPs, the atoms are shared only when upper-level groups fall in the same cluster.

The same nesting structure is applied in the nested Chinese restaurant process (nCRP) (Blei et al., 2003, 2010), which is proposed in the context of topics modelling. The nCRP metaphor involves a set of restaurants organised according to branching structure which forms a tree whose nodes represent restaurants. All customers enter the restaurant at the root node and select the table according to the usual CRP. Associated with each table in the restaurant is a note providing the address of the restaurant where the customers will visit next. The repetition of this process will yield a tree with infinite branching factors and infinite depth, where the paths down the tree correspond to customers. There will be up to N paths if there are N customers, as some paths may be selected by multiple customers.

The same nesting strategy can also be applied for beta process. Thus the two-level nested Beta process (nBP) can be formally defined as:

$$\begin{aligned} B &\sim \text{BeP}\left(\sum_{k=1}^{\infty} p_k^* \delta_{B_k^*}\right) \\ B_k^* &= \sum_{j=1}^{\infty} p_{kj} \delta_{\theta_{kj}} \end{aligned} \tag{2.20}$$

where the weights $\{p_k^*\}$ and $\{p_{kj}\}$ are defined as in Equation (2.13). The intuition behind the nested BP is similar to what has been discussed for nDP. More specifically, the two-level nBP defined in Equation (2.20) is the beta process over an infinite set of atoms, each of which is itself the beta process, B_k^*. In resemblance to nDP, a finite number of beta processes are selected among an infinite set of beta processes, $\{B_K^*\}$. However, the draw from Bernoulli process, B, will be a finite collection of components, as opposed to nDP, where the draw from DP, G consists of only a single component. The nBP can also be extended to an arbitrary number of levels.

An extension of IBP to nIBP can also be obtained using the same definition of nCRP. The idea of nIBP can described using the Indian buffet metaphor. Customers visit a city in which there is an infinite number of Indian buffet restaurants. For each restaurant, each of dishes is uniquely provided by another restaurant. Assume that there is a single restaurant that does not provide its dish to any other restaurant. Thus

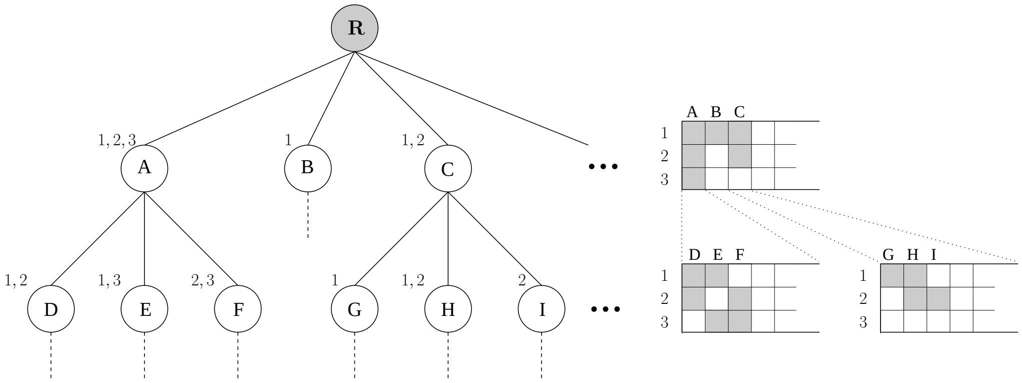


Figure 2.2 – The graph and binary matrices generated by the nIBP

this restaurant will correspond to the root of the tree. The customer starts at the first restaurant choosing some number of dishes. Associated with each dish is a note indicating which restaurant it comes from. After trying the chosen dishes, on the next day, the same customer goes to the restaurants that supply the dishes he has chosen. He chooses the dishes and repeats this process forever. Subsequent customers enter the first restaurant and follow the same process as the previous customers.

Figure 2.2 illustrates the nested Indian buffet restaurant process with 3 customers. The figure shows the tree constructed from the nIBP and the corresponding binary matrices generated at each level of the tree. The dishes are denoted as nodes in the tree, although they can also be interpreted as the Indian restaurants. According to this, the root node of the tree has no significant role. The numbers next to each node indicate the customers that have chosen the dishes. The first customer enters the restaurant at the root node choosing dishes A, B, and C. Therefore, the entries in the first row of the top binary matrix will be active. The customer then proceeds to the next level and choose more sets of dishes based on the dishes from the first level. From A, the dishes D and E are chosen by the first customer. From C, the customer selects the dishes G and H. Thus the entries in the lower-level binary matrices according to the first customer and the chosen dishes will be active as illustrated in the figure. Note that the third row of the second lower-level binary matrices is empty as the third customer does not choose dish C.

The previous example of the nIBP indicates that the customers can choose multiple paths down the tree. This is the key difference between nCRP and nIBP. Although both processes can be used to define the distribution over trees with infinite branching factors

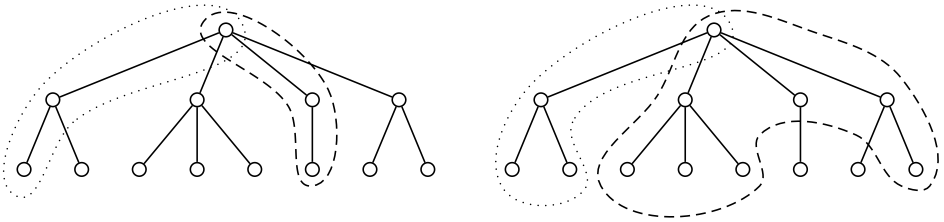


Figure 2.3 – the distribution over paths (left) and sub-trees (right) defined by nCRP and nIBP for a fixed tree

and infinite depth, for a fixed tree, the nCRP induces the distribution over all possible paths down the tree, whereas the nIBP induces the distribution over all possible sub-trees. This difference is illustrated in Figure 2.3. Note that paths are also contained in a set of sub-trees. Thus which model is more appropriate depends on the choice of the applications.

It is also worth mentioning the interpretation of parameters in the nIBP. Recall that the average number of features per object and the overall number of features are characterised by α and β, respectively. Denote by $\{\alpha_{ij}, \beta_{ij}\}$ the parameters of the IBPs at the i-th level of nIBP for $i = 1, ..., \infty$ and $j = 1, ..., P_i$, where P_i is the number of IBPs at the i-th level. At a particular node in the tree, α_{ij} and β_{ij} define the average branching factors per object and the overall branching factors. Consider when α_{ij} and β_{ij} are drawn from a common distribution with means $\overline{\alpha}$ and $\overline{\beta}$, respectively. The variances are assumed very small, such that all draws from the distributions are roughly equal. Therefore, the roles of $\{\alpha_{ij}, \beta_{ij}\}$ can be discussed using $\overline{\alpha}$ and $\overline{\beta}$.

Assume that parameters $\{\alpha_{ij}, \beta_{ij}\}$ of the nIBP are obtained through $\overline{\alpha}$ and $\overline{\beta}$. The average branching factors of trees drawn from this nIBP is $\overline{\alpha}$. If N trees are drawn from this distribution, the values of $\overline{\beta}$ control the overlap among these trees. At small values of $\overline{\beta}$, the trees tend to overlap, that is, they share a high number of paths down the tree. Conversely, at high values of $\overline{\beta}$ there is a high degree of feature repulsion for the IBP associated with each node, with the probability of two trees sharing the same path being low. Note that $\overline{\alpha}$ and $\overline{\beta}$ are used only to simplify the analysis. It is also possible to define the distribution over $\{\alpha_{ij}, \beta_{ij}\}$ separately for each level, or even for each node, in the tree. In which case, the average branching factors and the probability of two trees sharing the same path will be different at each level and each node depending on the

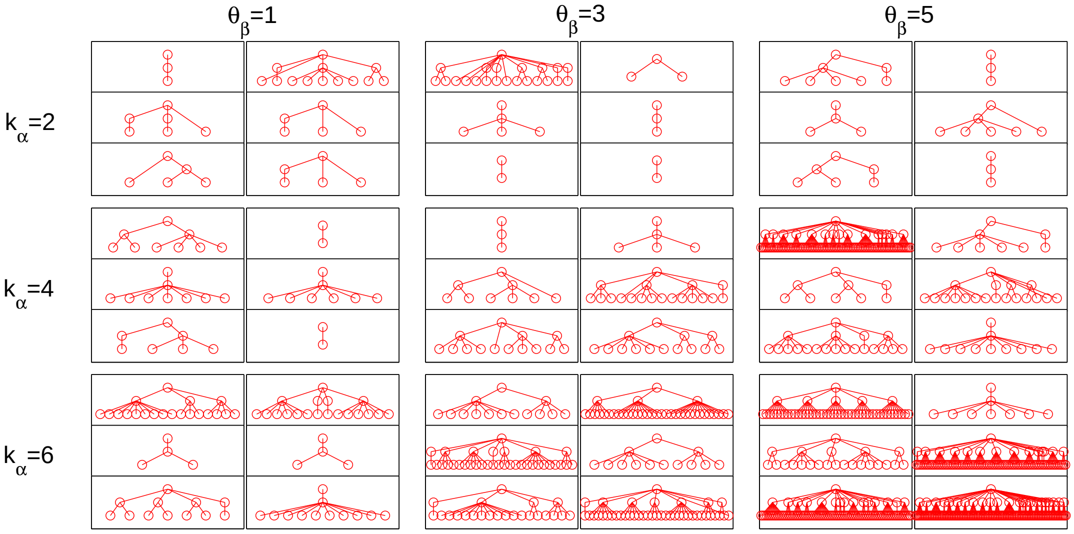


Figure 2.4 – The trees sampled from nIBP with different values of $\{\alpha_{ij}, \beta_{ij}\}$

values of α_{ij} and β_{ij}, respectively. How to choose appropriate strategy for allocating the distributions over $\{\alpha_{ij}, \beta_{ij}\}$ depends on the applications.

To illustrate effects of nIBP parameters, several samples from nIBP are generated with different parameter settings. The trees induced by these samples are depicted in Figure 2.4. A straightforward generative model is used as follows. Firstly, draw α_{11} and β_{11} from the priors and then draw a binary matrix $\mathbf{Z}_{11}$ from the IBP with these parameters. Secondly, for each column of $\mathbf{Z}_{11}$, a new binary matrix is drawn from the IBP with parameters drawn from their priors. That is, if $\mathbf{Z}_{11}$ has M columns, draw M binary matrices $\mathbf{Z}_{21}, ..., \mathbf{Z}_{2M}$ independently from IBP. Then each column of the new matrices is treated as a root node of the sub-trees, which will be generated by the repetition of these steps. In this example, the process is stopped at the predefined depth of the tree. Despite, this generative process requires a great amount of computation, as the number of nodes grows exponentially with respect to the depth of the tree. Therefore, it is of interest to find an alternative way to sample from the nIBP efficiently and compare this process with some existing stochastic processes such as the branching process.

For each sample from the nIBP, the common priors are defined over $\{\alpha_{ij}\}$ and $\{\beta_{ij}\}$. These parameters are drawn from the Gamma distributions with different shape and scale parameters, as shown in Figure 2.5. For α, the scale parameter is fixed at $\theta = 0.3$ and vary the values of shape parameter k, while the shape parameter is fixed at 1 and the scale parameter varies for β. The maximum depth of all samples is 2 and

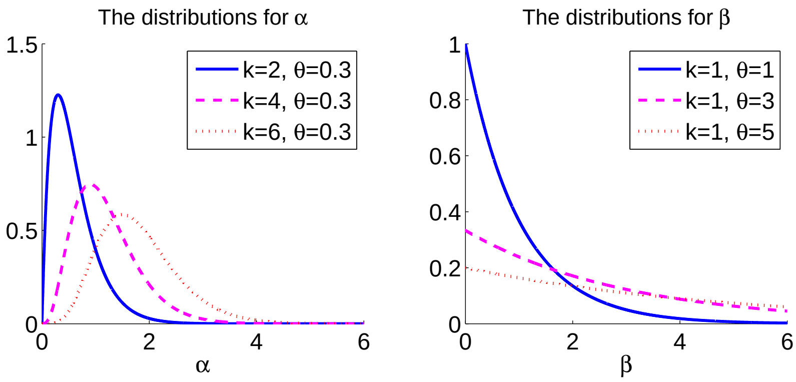


Figure 2.5 – The different distributions over α and β, which are drawn from the gamma distribution with shape parameters in k and scale parameters in θ

$N = 5$. Therefore, each tree shown in Figure 2.4 will consist of 5 overlapping sub-trees. For each distribution over α and β, 6 trees are generated, resulting in a total of 54 trees.

As shown in Figure 2.4, diverse tree structures can be observed. However, identical tree structures can also be observed in this set of samples. Recall that α and β affect the average number of features per object and the overall number of features, respectively. Consider when $\theta_\beta = 1$ and $k_\alpha = 2, 4, 6$ in which the Gamma distribution with high value of k_α tends to concentrate more on larger α. With large values of α, the sampled tree tends to have a high branching factor, as illustrated in Figure 2.4. Changing θ_β also affects the branching factor of the samples since β can be interpreted as the degree of features shared between objects. With small β, a number of features are shared among all objects, so the branching factor is approximately α. However, for large values of β, the overall number of features is increased to preserve the average number of features at α. These cases are illustrated in Figure 2.4 when the values of θ_β vary.

To the best of our knowledge, although some applications of nCRP have already been proven successful, there are still no concrete applications of the nIBP. In Chapter 3, an extension of the ISA using the two-level nIBP is proposed.

Chapter 3

Infinite Independent Subspace Analysis

This chapter discusses the Independent Component Analysis (ICA) and Independent Subspace Analysis (ISA). The proposed nonparametric models of ISA, which is called the infinite Independent Subspace Analysis (iISA), using the nIBP is also introduced. The chapter begins with an introduction of the basic ICA model and related works. A natural extension of ICA is then proposed in the following section. In this chapter, ongoing theoretical researches and applications of ISA, as well as its limitations, are reviewed. To overcome these limitations, we propose the nonparametric ISA model, which offers more flexibility in learning the hidden structures in the data and can therefore be applied in a wider range of applications.

3.1 Independent component analysis

ICA has been extensively studies for decades (Comon, 1994; Jutten and Herault, 1991). It is one of the most widely used methods for performing blind source separation (BSS), which aims to recover unknown independent signals from observed linear superposition of them. For example, consider the cocktail-party problem in which you are at a party with a number of people. There are several microphones in different locations in this party that are used to record the conversations among these people. Given the recorded signals from these microphones, you are interested in extracting the speech signals emitted by individual speakers or identifying the speech signals of particular speakers. The recorded time signals from the microphones can be denoted by $x_i(t)$, where $i = 1, ..., N$ and the speech signals of each speaker by $s_j(t)$, where $j = 1, ..., K$. In this case, there are N microphones and K speakers. Each of these recorded signals $x_i(t)$ can be expressed as a linear combination of speech signals $s_j(t)$ that can be written as a linear equation:

$$x_i(t) = a_{i1}s_1(t) + a_{i2}s_2(t) + ... + a_{iK}s_K(t) \ \ , \tag{3.1}$$

where $a_{i1}, ..., a_{iK}$ are the coefficients. The goal of BSS is to recover the signals s_j and the coefficient a_{ij}.

Although the statistical properties of source signals are unknown, it has been shown that the assumptions of statistical independence and nongaussianity of source signals suffice for most practical applications. Under these conditions, the source signals can be recovered up to a permutation, scales, and signs. ICA has been proven successful in many applications, including separating artifacts in MEG data, finding hidden factors in financial data, reducing noise in natural images, and inferring gene signatures from the gene expression data. There is, however, very little knowledge on mixing matrix and assumptions on the source signals need to be made.

The general assumption of ICA is that the observed data $\mathbf{y}_t$ can be written in terms of a linear superposition of independent hidden sources, $\mathbf{x}_t$ as

$$\mathbf{y}_t = \mathbf{x}_t\mathbf{A} + \boldsymbol{\epsilon}_t \tag{3.2}$$

where $\mathbf{A}$ is the mixing matrix and $\boldsymbol{\epsilon}_t$ is Gaussian noise. Note that only the random vector $\mathbf{y}_t$ is observed and both $\mathbf{x}_t$ and $\mathbf{A}$ must be estimated from the data.

The most important assumption regarding the hidden sources in ICA is that they are mutually independent. That is, N random variables $x_1, ..., x_N$ are said to be mutually independent if and only if the joint pdf can be factorised into the following way:

$$p(x_1, ..., x_N) = \prod_{i=1}^{N} p(x_i) \tag{3.3}$$

Another form of independence is uncorrelatedness. Two variables are said to be uncorrelated if their covariance is zero. If the variables are independent, they are uncorrelated, but not vice versa. Due to this property, many ICA algorithms constrain the estimation procedure to give uncorrelated estimates of the independent components. Moreover, the data is usually preprocessed to remove any correlation in the data before the application of the ICA algorithms. Thus, in the noiseless case, we want to find the *demixing* *matrix* $\mathbf{W}$ such that

$$\mathbf{x}_t = \mathbf{y}_t \mathbf{W} \tag{3.4}$$

gives the components that are maximally mutually independent. In this situation, it was shown by Comon that the mixing matrix $\mathbf{A}$ is identifiable up to scaling and permutation if at most one source is Gaussian (Comon, 1994). Intuitively, nongaussianity of the sources implies independence due to the Central Limit Theorem. Therefore, the sources could be recovered by taking as $\mathbf{W}$ a matrix that maximize the nongaussianity of $\mathbf{y}_t\mathbf{W}$.

Several quantitative measures of nongaussianity have been used for ICA estimation. The kurtosis or the fourth-order cumulant is a classical measure of nongaussianity. Althought kurtosis is theoretically and computationally simple, it can be very sensitive to outliers, and therefore not a robust measure of nongaussianity. From the perspective of information theory, negentropy is another important measure of nongaussianity based on the differential entropy. It is known that the entropy can be used to measure the degree of information provided by the random variables. Since a Gaussian variable has the largest entropy among all random variables of equal variance (Cover and Thomas, 1991), the entropy can be used as a measure of nongaussianity. Negentropy is the differential entropy of a component $\mathbf{x}_t$, minus the differential entropy of a Gaussian component with the same covariance. Thus, it is always positive and zero only if the component is Gaussian. Negentropy is a stable estimate (Hyvärinen, 1999b) but difficult to calculate. Therefore, the ICA estimation is often based on an approximation of negentropy. In addition to the kurtosis and negentropy, the independent sources can be recovered by minimizing the mutual information, which can be defined using the differential entropy. It is a measure of dependency between random variables. It can be shown that finding the independent sources by minimizing the mutual information is roughly equivalent to maximizing the negentropy (Hyvärinen, 1999b).

Another important approach to ICA estimation is the *maximum likelihood* (Belouchrani and Cardoso, 1995; Karvanen et al., 2000; Pham, 1992). In this framework, one begins with the probabilistic model which assigns a probability density $p(\{x_i\}|\mathbf{W})$ to the data set $\{x_i\}$ (See, e.g., Pham (1992)). Then we find the demixing matrix by maximizing $p(\{x_i\}|\mathbf{W})$ with respect to $\mathbf{W}$ using standard approaches in optimization, e.g., stochastic gradient descent (Pearlmutter and Parra, 1997). The maximum likelihood method was shown to have close connections to several methods in ICA estimation. For example, several works (Cardoso, 1997; Pearlmutter and Parra, 1997) have shown that the maximum likelihood method is equivalent to the infomax principle from the neural network viewpoint (Bell. and Sejnowski, 1995) as well as the mutual information.

3.2 Independent subspace analysis

The important assumption of ICA is that all hidden sources are mutually independent which may not be true for some real-world applications. This assumption prevents the model from recovering the sources that are dependent on one another. Instead, in Independent Subspace Analysis (ISA) the assumption is that there are M of D-dimensional independent sources denoted by $\mathbf{x}^1, ..., \mathbf{x}^M$ where $\mathbf{x}^i \in \mathbb{R}^D$. If all hidden sources are written as $\mathbf{x} = [(\mathbf{x}^1), ..., (\mathbf{x}^M)] \in \mathbb{R}^{DM}$, the observed data is assumed to be generated from the generative model identical to ICA shown in Equation (3.2), where $\mathbf{A} \in \mathbb{R}^{DM \times DM}$. In the ISA model, we assume that $\mathbf{x}^i$ is independent of $\mathbf{x}^j$ for $i \neq j$. Note that the dimensionality D of each $\mathbf{x}^i$ need not be equal. If the dimensionalities of M independent sources $\mathbf{x}^1, ..., \mathbf{x}^M$ are $D_1, ..., D_M$, respectively, all hidden sources can be represented as $\mathbf{x} \in \mathbb{R}^{D^*}$, where $D^* = D_1 + \cdots + D_M$. For the special case of $D_i = 1$ for all i, the ICA problem is recovered.

In the ICA problem, given signals $\mathbf{y}_t$, the sources $\mathbf{x}_t^i$ can be recovered only up to sign, up to arbitrary scaling factors, and up to an arbitrary permutation (Theis, 2004). The ISA task has even more freedom, the sources $\mathbf{x}_t^i$ can be recovered up to an arbitrary permutation and an D-dimensional linear invertible transformation. This can be seen by considering matrix $C \in \mathbb{R}^{DM \times DM}$ made of a permutation matrix of size $D \times D$ with each element made of $M \times M$ block-matrix with invertible C_i blocks places only to the non-zero elements of the permutation matrix. Then, $\mathbf{y} = \mathbf{x}\mathbf{A} = \mathbf{x}C^{-1}C\mathbf{A}$, and because $\mathbf{x}^i$ is independent of $\mathbf{x}^j$, thus $C_i\mathbf{x}^i$ is independent of $C_j\mathbf{x}^j$ $\forall i \neq j$. Therefore, in the ISA model, matrices $\mathbf{A}$ and $\mathbf{A}C^{-1}$ and sources $\mathbf{x}^i$ and $C_i\mathbf{x}^i$ are indistinguishable.

The first generalization of the ICA model includes the dependencies between components known as multidimensional ICA (MICA) has been introduced by Cardoso (Cardoso, 1998) based on a geometric parameterization. Although providing the general framework for the ISA model, no uniqueness results were presented. Moreover, it is not clear how to apply MICA to the arbitrary random vectors. In a special case of equal group sizes (k-ISA), the uniqueness up to permutation and scaling is not guaranteed as it is limited to only random vectors following the generative model of k-ISA. For an arbitrary random vector, the decomposition into groups based on the independence assumption cannot be unique. Later the notion of *irreducibility* is presented on which the uniqueness results (Gutch and Theis, 2007; Theis, 2006) are based in the case of k-ISA. The combination of k-ISA with invariant feature subspace analysis is introduced in Hyvärinen and Hoyer (2000). The dependence with a subspace can be modelled explicitly, leading to an efficient algorithm without performing the problematic multidimensional density estimation. Bach and Jordan (2003) introduce the ISA model, which assumes that the components can be grouped into clusters. The components within cluster are dependent, whereas are independent between clusters. The ISA mode utilizing the k-nearest neighbor distance between data points (Póczos and Lörincz, 2005) finds the dependent components by estimating the differential entropies.

3.3 Infinite independent component analysis

A nonparametric Bayesian extension of ICA is proposed by Knowles and Ghahramani (2007). The observed data is modelled by a linear superposition, $\mathbf{A}$, of the potentially infinite number of hidden sources. The infinite binary matrix is used to indicate whether a given source is active. In infinite Independent Component Analysis (iICA), for each data point t, a binary vector $\mathbf{z}_t$ is defined to indicate which elements of $\mathbf{x}_t$ are active. Using the same formula of standard ICA in Equation (3.2), the observed data $\mathbf{Y}$ is assumed to be generated as follows:

$$\mathbf{Y} = (\mathbf{Z} \circ \mathbf{X})\mathbf{A} + \mathbf{E} \tag{3.5}$$

where $\circ$ is the Hadamard, i.e., elementwise product, and $\mathbf{Y}$, $\mathbf{Z}$, $\mathbf{X}$, and $\mathbf{E}$ are concatenated matrices of $\mathbf{y}_t$, $\mathbf{z}_t$, $\mathbf{x}_t$, and $\boldsymbol{\epsilon}_t$, respectively. Note that the data points are arranged in the rowwise manner. In this model, the number of hidden sources is potentially unbounded, so the number of columns of $\mathbf{Z}$ can be infinite. However, the actual number of non-zero entries will be finite. This allows greater flexibility as the number of sources need not be known a priori, but will be inferred from the observed data.

In this model, the hidden sources $\mathbf{x}_t$ has a Laplacian prior with fixed variance as any variation can be absorbed into elements of mixing matrix $\mathbf{A}$. The prior of the

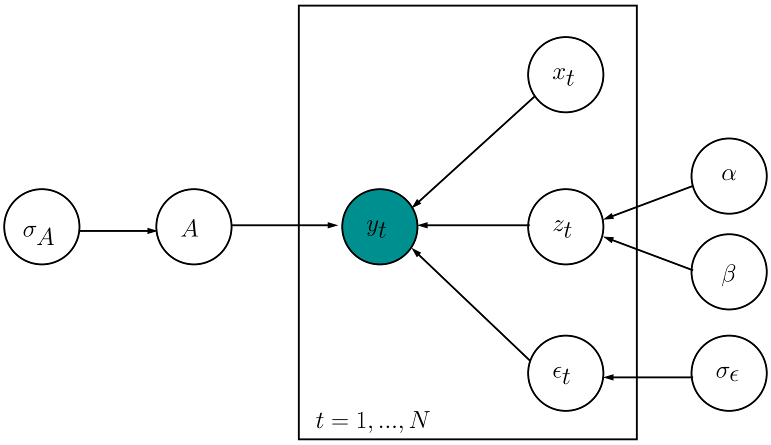


Figure 3.1 – The graphical model of iICA model

elements of $\mathbf{A}$ is Gaussian with variance σ_A^2, which has an inverse Gamma prior. The prior on binary matrix $\mathbf{Z}$ is the IBP with parameters α and β. The graphical model of iICA is shown in Figure 3.1. The generative model can be summarized below.

$$\begin{array}{llll} x_{tk} & \sim \mathcal{L}(1) & \mathbf{Z} & \sim \mathcal{IBP}(\alpha, \beta) \\ \mathbf{a}_k & \sim \mathcal{N}(0, \sigma_A^2) & \sigma_\epsilon^2 & \sim \mathcal{IG}(a, b) \\ \sigma_A^2 & \sim \mathcal{IG}(c, d) & \alpha & \sim \mathcal{G}(e, f) \end{array}$$

The iICA model is implemented in this thesis to evaluate its performance and to gain insight that may be useful for developing the iISA model. The iICA and iISA models have various shared properties as discussed in Section 3.4.

3.4 Independent subspace analysis using nested IBP

We propose an extension of the ISA model called *infinite ISA* which is based on the iICA model. Using two-level nIBP, the proposed model can be used to recover the hidden sources that are not mutually independent. Let J be the number of subspaces, and denote by K_j and $K = \sum_j K_j$ the dimension of subspace j for $j = 1, ..., J$, and the total number of sources, respectively. Given a set of i.i.d observed data points $\{\mathbf{y}_t\}_{t=1}^N$ where $\mathbf{y}_t \in \mathbb{R}^D$, we assume each data point is generated according to the infinite ISA (iISA) model as follows:

```
• For each subspace $j = 1, ..., J$
    – sample $\omega_j \sim \mathrm{Beta}(\frac{\alpha\beta}{J}, \beta)$
    – for each source $k = 1, ..., K_j$:  sample $\pi_{jk} \sim \mathrm{Beta}(\frac{\alpha_j\beta_j}{K_j}, \beta_j)$
• For each row $k = 1, ..., K$:  sample $\mathbf{a}_k \sim \mathcal{N}(0, \sigma_A^2\mathbf{I})$
• For each data point $\mathbf{y}_t$, $t = 1, ..., N$
    – sample $\mathbf{u}_t \sim \mathrm{Bernoulli}(\boldsymbol{\omega})$
    – for each active subspace $u_{tj} = 1$:
        ∗ sample $v_{tj} \sim \mathrm{Uniform}(0, 1)$
        ∗ sample $\mathbf{z}_{tj} \sim \mathrm{Bernoulli}(\boldsymbol{\pi}_j)$
        ∗ for each active source $z_{tjk}$, $k = 1, ..., K_j$:  sample $x_{tjk} \sim \mathcal{L}(1)$
    – sample $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \sigma_\epsilon^2\mathbf{I})$
    – $\mathbf{y}_t = \sum_{j=1}^{J} u_{tj} v_{tj} \left[(\mathbf{z}_{tj} \circ \mathbf{x}_{tj})\mathbf{A}_j\right] + \boldsymbol{\epsilon}_t$
```

where $\mathbf{A}_j$ is the $K_j \times D$ mixing matrix and $\mathbf{x}_{tj}$ is the $1 \times K_j$ vector of hidden source signals in subspace j. In this generative model, the vector $\mathbf{u}_t$ acts as a binary vector indicating which subspaces are active for the data point $\mathbf{y}_t$. The binary vector $\mathbf{z}_{tj}$ specifies which source signals are active for active subspace j.

Let $\mathbf{U}$ be the binary matrix with an infinite number of columns created by stacking the binary vector $\mathbf{u}_t$. Although the number of columns of $\mathbf{U}$ is potentially unbounded, the IBP guarantees that the actual number of columns will be finite. Therefore, the number of subspaces J for a given data set is the actual number of columns of $\mathbf{U}$. Similarly, for each subspace $j = 1, ..., J$, let $\mathbf{Z}_j$ be a binary matrix created by stacking the vector $\mathbf{z}_{tj}$ for data point t. If the subspace j for data point t is not active, the t-th row of $\mathbf{Z}_j$ is assumed to be zero. Thus, the number of active columns of $\mathbf{Z}_j$ can be interpreted as a dimension of subspace j.

Due to the nesting strategy, some source signals can be dependent through the definition of subspaces, i.e., source signals in the same subspace are dependent, whereas those in the distinct subspaces are mutually independent. The source signals become dependent through the uniformly distributed variable $\mathbf{v}_t$. Therefore, the assumption of independence is imposed on subspaces rather than on individual source signals. Furthermore, by allowing the number of subspaces and their dimensions to be unbounded

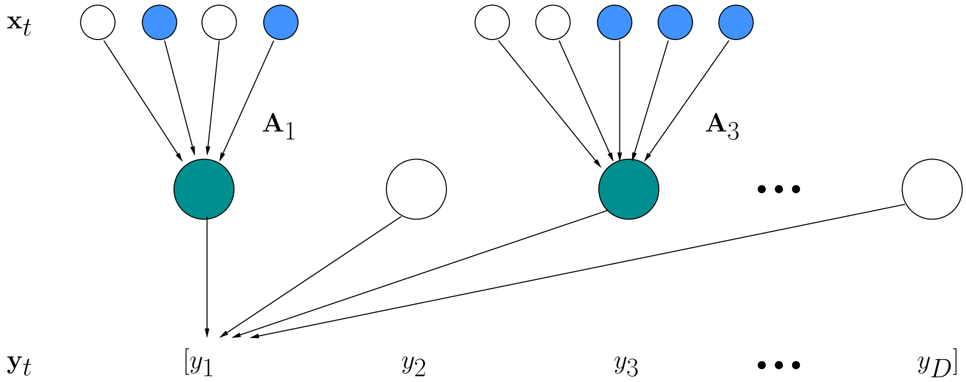


Figure 3.2 – The neural network view of the iISA

through the IBPs, the proposed model automatically find the appropriate number of subspaces and their dimensions for the given data set.

According to the iISA model, an entry to the hidden source signal is assumed to be drawn from the double exponential distribution $\mathcal{L}(1) = \frac{1}{2}\exp(-|x|)$ and each row of mixing matrix $\mathbf{A}$ is assumed to be drawn from an isotropic Gaussian distribution. Here, define the $1 \times K$ vector $\mathbf{s}_t$ as a block-binary vector whose values in each block are dictated by $\mathbf{u}_t$. To be precise, the definition of $\mathbf{s}_t$ can be illustrated as:

$$\mathbf{s}_t = \{\underbrace{11...1}_{K_1}\underbrace{00...0}_{K_2}...\underbrace{11...1}_{K_J}\} \tag{3.6}$$

Blocks in $\mathbf{s}_t$ correspond to the subspaces. All entries in block j are 1 if subspace j is active and 0 otherwise. From Equation (3.6), the subspaces 1 and J are active, whereas subspace 2 is inactive. We also define the vector $\mathbf{q}_t$ to have the same structure as $\mathbf{s}_t$, but its entries are instead dictated by the entries of vector $\mathbf{v}_t$. This variable renders the sources in each subspace to be dependent. Moreover, other relevant objects are defined as follows:

$$\mathbf{z}_t = [\mathbf{z}_{t1}...\mathbf{z}_{tJ}], \quad \mathbf{x}_t = [\mathbf{x}_{t1}...\mathbf{x}_{tJ}], \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_J \end{bmatrix} \tag{3.7}$$

Figure 3.2 illustrates the neural network view of the proposed model. The observed data $\mathbf{y}_t$ is a linear superposition, $\mathbf{A}$, of the potentially infinite number of hidden sources. In contrast to the iICA model, the assumption in the iISA model is that the hid-

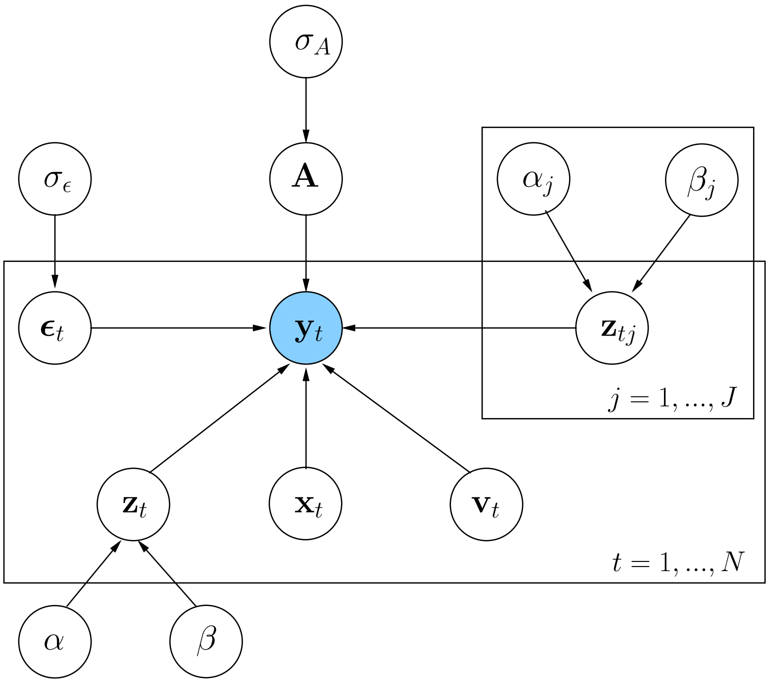


Figure 3.3 – The graphical model for the finite version of iISA model

den sources are separated into several groups, each of which is composed of dependent sources. The groups can be thought of as the subspaces spanned by hidden sources. Thus, the iISA model is essentially the ISA algorithms that is capable of recovering the dependent hidden sources where the number of subspaces and the number of hidden sources are unbounded.

For the first component y_1 illustrated in Figure 3.2, the first and second subspaces are active, which is specified by the upper-level IBP. Then the active sources in each subspace are specified by the lower-level IBPs. Although the mixing matrix can be partitioned into submatrices according to the subspaces, we assume the identical prior on these submatrices. For inactive subspaces, e.g., second and third ones in the figure, the sources in those subspaces have no influence on the observed component y_1.

Using the definitions in Equation (3.7), we can write the generative process of observed data as $\mathbf{y}_t = (\mathbf{s}_t \circ \mathbf{q}_t \circ \mathbf{z}_t \circ \mathbf{x}_t)\mathbf{A} + \boldsymbol{\epsilon}_t$, which can be written in the matrix form as

$$\mathbf{Y} = (\mathbf{S} \circ \mathbf{Q} \circ \mathbf{Z} \circ \mathbf{X})\,\mathbf{A} + \mathbf{E} \tag{3.8}$$

where $\mathbf{Y}, \mathbf{S}, \mathbf{Q}, \mathbf{Z}, \mathbf{X}$, and $\mathbf{E}$ are concatenated matrices of $\mathbf{y}_t, \mathbf{s}_t, \mathbf{q}_t, \mathbf{z}_t, \mathbf{x}_t$, and $\boldsymbol{\epsilon}_t$, respectively. The graphical model of iISA is depicted in Figure 3.3. Since the value of hyperparameters are not known a priori, we endow them with prior distributions defined below:

$$\begin{aligned} \sigma_\epsilon^2 &\sim \mathcal{IG}(a,b) \quad & \sigma_A^2 &\sim \mathcal{IG}(c,d) \\ \alpha &\sim \mathcal{G}(e,f) \quad & \alpha_j &\sim \mathcal{G}(g,h) \end{aligned}$$

Another important point to note about the iISA model is the maximality of the decomposition (Cardoso, 1998). Assume that the hidden sources $\mathbf{x}$ can be decomposed into three groups $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$. Then, according to the definition of iISA model, the coarser decomposition in two groups $\{\mathbf{x}_1 + \mathbf{x}_2, \mathbf{x}_3\}$ is also feasible. However, the latter decomposition is weaker than the former as the first group in the second case can be further decomposed. Therefore, to avoid trivialities, iISA model should be able to find the finest decomposition, i.e., find as many independent subspaces as possible.

Chapter 4

Inference

This chapter presents the inference algorithm for the proposed iISA model. The conditional probability of each variable in the model is derived for inference using Gibbs sampling. The MCMC sampling method based on the Metropolis-Hasting update is also introduced for sampling the active subspace, updating the number of sources in the subspaces, and updating the number of subspaces. Lastly, all relevant parameters that are important for the inference algorithm to be successful are summarised.

4.1 Likelihood

From the iISA model, we have $\mathbf{y}_t = (\mathbf{s}_t \circ \mathbf{q}_t \circ \mathbf{z}_t \circ \mathbf{x}_t)\mathbf{A} + \boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t$ is a Gaussian noise. Let $\mathbf{g}_t$ be the result of elementwise products of all relevant quantities, i.e., $\mathbf{g}_t = \mathbf{s}_t \circ \mathbf{q}_t \circ \mathbf{z}_t \circ \mathbf{x}_t$. Hence, the likelihood function for a single data point $\mathbf{y}_t$ is

$$\begin{aligned} P(\mathbf{y}_t|\mathbf{A}, \mathbf{s}_t, \mathbf{q}_t, \mathbf{z}_t, \mathbf{x}_t, \sigma_\epsilon^2) &= \mathcal{N}(\mathbf{y}_t;\ \mathbf{g}_t\mathbf{A}, \sigma_\epsilon^2\mathbf{I}) \\ &= \frac{1}{\sqrt{2\pi}\sigma_\epsilon} \exp\left\{ -\frac{1}{2\sigma_\epsilon^2}(\mathbf{y}_t - \mathbf{g}_t\mathbf{A})(\mathbf{y}_t - \mathbf{g}_t\mathbf{A})^\mathsf{T} \right\} \end{aligned} \tag{4.1}$$

Given N i.i.d data points $\{\mathbf{y}_t\}_{t=1}^N$, the likelihood function for the whole data set is

$$\begin{aligned} P(\mathbf{Y}|\mathbf{A}, \mathbf{S}, \mathbf{Q}, \mathbf{Z}, \mathbf{X}) &= \prod_{t=1}^{N} P(\mathbf{y}_t|\mathbf{A}, \mathbf{s}_t, \mathbf{q}_t, \mathbf{z}_t, \mathbf{x}_t) \\ &= \frac{1}{(2\pi\sigma_\epsilon^2)^{\frac{ND}{2}}} \exp\left\{ -\frac{1}{2\sigma_\epsilon^2}\mathrm{tr}(\mathbf{Y} - \mathbf{G}\mathbf{A})(\mathbf{Y} - \mathbf{G}\mathbf{A})^\mathsf{T} \right\} \end{aligned} \tag{4.2}$$

where $\mathbf{G}$ is the concatenated matrix of $\mathbf{g}_t$.

4.2 The hidden source $\mathbf{x}_t$

The conditional probability distribution for entries in $\mathbf{X}$ is derived. Because the entries in $\mathbf{x}_t$ are partitioned into blocks according to the subspaces and they have the same form of conditional distribution, the conditional probability will be derived based on a single subspace. However, this result also applies to the entries of $\mathbf{x}_t$ in other subspaces. For subspace j, we sample each element of $\mathbf{x}_t$ for which $s_{tjk} = 1$ and $z_{tjk} = 1$. Denote the k-th row of $\mathbf{A}_j$ by $\mathbf{a}_{jk}$. Since the prior on x_{tjk} is Laplace distribution, the conditional probability of x_{tjk} given all other variables is the piecewise Gaussian distribution given by

$$P(x_{tjk}|\mathbf{A}, \mathbf{s}_t, \mathbf{q}_t, \mathbf{z}_t, \mathbf{x}_{tj\neg k}) = \begin{cases} \frac{\Upsilon_+}{\Omega}\mathcal{N}(x_{tjk}; \mu_+, \sigma) & \text{if } x_{tjk} > 0 \\ \frac{\Upsilon_-}{\Omega}\mathcal{N}(x_{tjk}; \mu_-, \sigma) & \text{if } x_{tjk} < 0 \end{cases} \tag{4.3}$$

where the means μ_+, μ_- and variance σ^2 can be defined as follows:

$$\mu_+ = \frac{\mathbf{a}_{jk}\boldsymbol{\epsilon}_{tj\neg k}^{\mathsf{T}} - \sigma_\epsilon^2}{\mathbf{a}_{jk}\mathbf{a}_{jk}^{\mathsf{T}}}, \ \ \mu_- = \frac{\mathbf{a}_{jk}\boldsymbol{\epsilon}_{tj\neg k}^{\mathsf{T}} + \sigma_\epsilon^2}{\mathbf{a}_{jk}\mathbf{a}_{jk}^{\mathsf{T}}}, \ \ \sigma^2 = \frac{\sigma_\epsilon^2}{\mathbf{a}_{jk}\mathbf{a}_{jk}^{\mathsf{T}}} \tag{4.4}$$

This distribution is guaranteed to be continuous by the choice of Υ_+ and Υ_- and to be correctly normalized by the choice of Ω.

$$\Upsilon_+ = \mathcal{N}(0; \mu_-, \sigma) \tag{4.5}$$

$$\Upsilon_- = \mathcal{N}(0; \mu_+, \sigma) \tag{4.6}$$

$$\Omega = \Omega_-\Upsilon_- + \Omega_+\Upsilon_+ \tag{4.7}$$

where $\Omega_- = F(0; \mu_-, \sigma)$ and $\Omega_+ = 1 - F(0; \mu_+, \sigma)$. To sample the hidden sources, we use the technique described in Knowles and Ghahramani (2007).

4.3 The source dependency $\mathbf{v}_t$

As defined above, the block vector $\mathbf{q}_t$ is contructed from the vector $\mathbf{v}_t$ whose entries are uniformly distributed. These variables render the source signals in the same subspace to be dependent. For each subspace j, we sample v_{tj} from the conditional distribution given all other variables with $u_{tj} = 1$. Since each element in $\mathbf{v}_t$ is tied to multiple entries in $\mathbf{q}_t$, we define the following notations. Let $\mathbf{q}_{tj:}$ be the j-th block of $\mathbf{q}_t$ and $\mathbf{x}_{tj:}$, $\mathbf{z}_{tj:}$, and $\mathbf{s}_{tj:}$ be the corresponding blocks of $\mathbf{x}_t$, $\mathbf{z}_t$, and $\mathbf{s}_t$, respectively. Given the current values of all other variables, the conditional probability of the source dependency

$\mathbf{q}_{tj:}$ can be written as follows:

$$\begin{aligned} P(\mathbf{q}_{tj:}|\mathbf{A},\mathbf{x}_t,\mathbf{z}_t,\mathbf{s}_t,\mathbf{q}_{t\neg(j:)}) &\propto P(\mathbf{y}_t|\mathbf{A},\mathbf{x}_t,\mathbf{z}_t,\mathbf{s}_t,\mathbf{q}_t,\sigma_\epsilon^2)P(\mathbf{q}_{tj:}|\mathbf{q}_{t\neg(j:)}) \\ &\propto P(\mathbf{y}_t|\mathbf{A},\mathbf{x}_t,\mathbf{z}_t,\mathbf{s}_t,\mathbf{q}_t,\sigma_\epsilon^2)\mathbb{I}_{\{0<\mathbf{q}_{tj:}<1\}} \end{aligned} \tag{4.8}$$

The Equation (4.8) follows from the fact that the prior of $\mathbf{v}_{tj}$ is the standard uniform distribution. Replacing $\mathbf{q}_{tj:}$ with v_{tj}, the conditional probability of v_{tj} can be written as

$$P(v_{tj}|\mathbf{A},\mathbf{x}_t,\mathbf{z}_t,\mathbf{s}_t,\mathbf{v}_{t\neg j}) \propto \mathcal{N}(v_{tj};\mu_v,\sigma_v^2)\mathbb{I}_{\{0<v_{tj}<1\}} \tag{4.9}$$

which is a two-sided truncated normal distribution with mean μ_v and variance σ_v^2 defined as follows.

$$\mu_v = \frac{(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j\boldsymbol{\epsilon}_{t\neg(j:)}^\mathsf{T}}{(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j\mathbf{A}_j^\mathsf{T}(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})^\mathsf{T}},\quad \sigma_v^2 = \frac{\sigma_\epsilon^2}{(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j\mathbf{A}_j^\mathsf{T}(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})^\mathsf{T}} \tag{4.10}$$

To generate samples from two-sided truncated normal distribution, we use the following method given that we know the cumulative distribution function $F(\cdot)$ and the inverse error function $\text{erf}^{-1}(\cdot)$.

$$\phi_l = F\left(\frac{(a-\mu_v)}{\sigma_v};0,1\right),\quad \phi_r = F\left(\frac{(b-\mu_v)}{\sigma_v};0,1\right) \tag{4.11}$$

$$v = \mu_v + \sigma_v\left(\sqrt{2}\text{erf}^{-1}(2(\phi_l+(\phi_r-\phi_l)w)-1)\right) \tag{4.12}$$

where a and b are the lower and upper truncation points, respectively, and w is a sample from $\text{Uniform}(a,b)$.

4.4 The mixing matrix A

From Bayes' rule, the conditional probability distribution of each row $\mathbf{a}_k$ of mixing matrix $\mathbf{A}$ can be written as

$$P(\mathbf{a}_k|\mathbf{A}_{\neg k},\mathbf{X},\mathbf{Y},\mathbf{Z},\mathbf{S},\mathbf{Q},\sigma_\epsilon^2,\sigma_A^2) \propto P(\mathbf{Y}|\mathbf{A},\mathbf{X},\mathbf{Z},\mathbf{S},\mathbf{Q},\sigma_\epsilon^2)P(\mathbf{a}_k|\sigma_A^2) \tag{4.13}$$

Let $\mathbf{E}_{\neg k}$ be the error matrix $\mathbf{E} = \mathbf{Y} - (\mathbf{S}\circ\mathbf{Q}\circ\mathbf{Z}\circ\mathbf{X})\mathbf{A}$ evaluated with $\mathbf{a}_k = 0$ and denote the k-th column of $(\mathbf{S}\circ\mathbf{Q}\circ\mathbf{Z}\circ\mathbf{X})$ by $\mathbf{g}_k^\mathsf{T}$. Following the derivation in Appendix B, the conditional distribution of each row of the mixing matrix given other variables is Gaussian $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Lambda})$ with mean and variance defined as

$$\boldsymbol{\mu} = \frac{\sigma_A^2}{\mathbf{g}_k\mathbf{g}_k^\mathsf{T}\sigma_A^2+\sigma_\epsilon^2}\mathbf{E}_{\neg k}^\mathsf{T}\mathbf{g}_k^\mathsf{T} \tag{4.14}$$

$$\boldsymbol{\Lambda} = \left(\frac{\mathbf{g}_k\mathbf{g}_k^\mathsf{T}}{\sigma_\epsilon^2}+\frac{1}{\sigma_A^2}\right)\mathbf{I}_{D\times D} \tag{4.15}$$

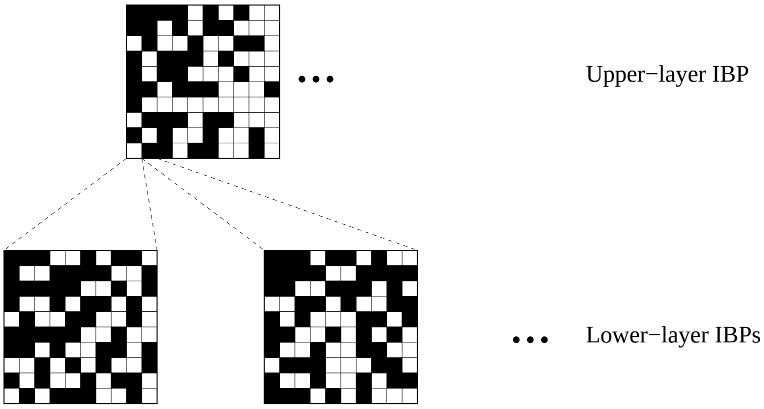


Figure 4.1 – The relation of binary matrices in the two-level nested IBP

4.5 The noise variance

The conditional distribution of σ_ϵ^2 can be obtained using Bayes's rule as follows:

$$\begin{aligned} P(\sigma_\epsilon^2|\mathbf{E}, a, b) &\propto P(\mathbf{E}|\sigma_\epsilon^2)P(\sigma_\epsilon^2|a, b) \\ &\propto \mathcal{IG}(\sigma_\epsilon^2; a + \frac{ND}{2}, \frac{b}{1 + \frac{b}{2}\text{tr}(\mathbf{E}\mathbf{E}^\mathsf{T})}) \end{aligned} \quad (4.16)$$

The detailed derivation of $P(\sigma_\epsilon^2|\mathbf{E}, a, b)$ can be found in Appendix B.

4.6 The mixing matrix variance

The conditional probability distribution of σ_A^2 can also be obtained using Bayes' rule as follows:

$$\begin{aligned} P(\sigma_A^2|\mathbf{A}, c, d) &\propto P(\mathbf{A}|\sigma_A^2)P(\sigma_A^2|c, d) \\ &\propto \mathcal{IG}(\sigma_A^2; c + \frac{DK}{2}, \frac{d}{1 + \frac{d}{2}\text{tr}(\mathbf{A}\mathbf{A}^\mathsf{T})}) \end{aligned} \quad (4.17)$$

For the detailed derivation of $P(\sigma_A^2|\mathbf{A}, c, d)$, please see Appendix B.

4.7 Active sources

In this section, we consider a conditional probability of the active source in a particular subspace. The lower-level IBPs separately specify which sources are active in each subspace for a particular data point. Therefore, the number of lower-level IBPs is equal to the number of subspaces, which is specified by the upper-level IBP. Figure 4.1 illustrates these relations in the two-level nIBP in which each column in the upper-level IBP corresponds to a lower-level IBP. Although the conditional probabilities of

z_{tjk} for all lower-level IBPs are identical, they still depend on the sources in other subspaces. To sample $\mathbf{Z}_j$, we define the ratio of the conditionals r such that $P(z_{tjk} = 1|\mathbf{A}, \mathbf{X}_{\neg(tjk)}, \mathbf{Y}, \mathbf{Z}_{\neg(tjk)}) = r/(r+1)$.

$$\begin{aligned} r &= \frac{P(z_{tjk} = 1|\mathbf{A}, \mathbf{X}_{\neg(tjk)}, \mathbf{Y}, \mathbf{Z}_{\neg(tjk)})}{P(z_{tjk} = 0|\mathbf{A}, \mathbf{X}_{\neg(tjk)}, \mathbf{Y}, \mathbf{Z}_{\neg(tjk)})} && (4.18) \\ &= \frac{P(\mathbf{y}_t|\mathbf{A}, \mathbf{x}_{tj\neg k}, \mathbf{z}_{tj\neg k}, z_{tjk} = 1, \sigma_\epsilon^2)P(z_{tjk} = 1|\mathbf{z}_{tj\neg k})}{P(\mathbf{y}_t|\mathbf{A}, \mathbf{x}_{tj\neg k}, \mathbf{z}_{tj\neg k}, z_{tjk} = 0, \sigma_\epsilon^2)P(z_{tjk} = 0|\mathbf{z}_{tj\neg k})} && (4.19) \end{aligned}$$

From the IBP prior, the ratio of priors is

$$\frac{P(z_{tjk} = 1|\mathbf{z}_{tj\neg k})}{P(z_{tjk} = 0|\mathbf{z}_{tj\neg k})} = \frac{m_{\neg tjk}/(\beta_j + N - 1)}{1 - m_{\neg tjk}/(\beta_j + N - 1)} = \frac{m_{\neg tjk}}{\beta_j + N - 1 - m_{\neg tjk}} \tag{4.20}$$

where $m_{\neg tjk}$ is the number of rows of $\mathbf{Z}_j$ other than t that are active at column k. To compute the ratio of likelihoods, we start by finding the likelihood with $z_{tjk} = 0$, which is

$$P(\mathbf{y}_t|\mathbf{A}, \mathbf{x}_{tj\neg k}, \mathbf{z}_{tj\neg k}, z_{tjk} = 0) = \frac{1}{(2\pi\sigma_\epsilon^2)^{\frac{D}{2}}} \exp\left(-\frac{\boldsymbol{\epsilon}_{tj\neg k}\boldsymbol{\epsilon}_{tj\neg k}^{\mathsf{T}}}{2\sigma_\epsilon^2}\right) \tag{4.21}$$

where $\boldsymbol{\epsilon}_{tj\neg k}$ is the error vector $\boldsymbol{\epsilon}_t$ evaluated with $z_{tjk} = 0$. If $z_{tjk} = 1$, we must integrate over all possible values of x_{tjk} to find the corresponding likelihood as follows:

$$P(\mathbf{y}_t|\mathbf{A}, \mathbf{x}_{tj\neg k}, \mathbf{z}_{tj\neg k}, z_{tjk} = 1, \sigma_\epsilon^2) = \int P(\mathbf{y}_t|\mathbf{A}, \mathbf{x}_{tj}, \mathbf{z}_{tj\neg k}, z_{tjk} = 1, \sigma_\epsilon^2)P(x_{tjk})\mathbf{d}x_{tjk} \tag{4.22}$$

After completing the square, the ratio of likelihoods is

$$\sigma\sqrt{\frac{\pi}{2}}\left[F(0;\mu_+,\sigma)\exp\left(\frac{\mu_+^2}{2\sigma^2}\right) + (1 - F(0;\mu_-,\sigma))\exp\left(\frac{\mu_-^2}{2\sigma^2}\right)\right] \tag{4.23}$$

where $F(0;\mu_+,\sigma) = \int_{-\infty}^{0}\mathcal{N}(x_{tjk};\mu_+,\sigma)\mathbf{d}x_{tjk}$ and $F(0;\mu_-,\sigma) = \int_{-\infty}^{0}\mathcal{N}(x_{tjk};\mu_-,\sigma)\mathbf{d}x_{tjk}$ are the cumulative distribution functions of Gaussian distribution with mean and variance defined in Equation (B.4).

4.8 Active subspaces

The matrix S specifies the active sources based on active subspaces which is determined by the matrix U. Recall that each row of S is the block vector in which each block corresponds to a subspace. All the values in the corresponding block will be 1 for the active subspace and 0 otherwise. That is, for a particular data point, if the subspace is not active, all sources in that subspace will not be used. On the other hand, if the subspace is active, the matrix $\mathbf{Z}$ further indicates which sources are used by the data point.

In contrast to the case of lower-level IBPs, the conditional probability of $\mathbf{U}$ becomes more complicated to derive as it involves integrating out the sparse binary vector $\mathbf{z}_{tj:}$ which cannot be done exactly due to its combinatorial nature. Therefore, some approximation techniques are needed to be employed. Additionally, the Gibbs sampler will need to operate in a very high dimensional space due to the nested structure of the iISA model, leading to a very slow mixing time. To resolve these problems, we propose the MCMC method to sample the active subspace variables u_{tj}. This proposed MCMC algorithm is based on the split-merge MCMC algorithm for the Dirichlet process mixture model (Jain and Neal, 2000). The split-merge method is based on the Metropolis-Hastings procedure in which appropriate proposals and transition probabilities are obtained by conducting the restricted Gibbs sampling scans.

We begin by discussing the basic idea of Metropolis-Hastings updates (Hastings, 1970; Metropolis et al., 1953). Assume that we want to generate a sequence of random samples from the probability distribution with density $\pi(x)$. A direct sampling from $\pi(x)$ is difficult, but we are able to evaluate the density given samples. The Metropolis-Hasting algorithm first draws a proposed state x^* from a proposal density $q(x^*|x)$, which depends on the current state. The new state x^* is accepted with the probability

$$r_{x \to x^*} = \min \left[1, \frac{q(x|x^*)}{q(x^*|x)} \frac{\pi(x^*)}{\pi(x)} \right] \tag{4.24}$$

and the next state is set to this proposed state. Otherwise, the current state is retained $x^* = x$.

To sample the binary-valued variable u_{tj}, we use the Metropolis-Hasting step to propose a change to this variable. That is, if the current state is $u_{tj} = 0$, the proposal state will be $u_{tj} = 1$ and vice versa. As each variable u_{tj} corresponds to a single lower-level IBP, the proposal distribution of the Metropolis-Hasting step is constructed by conducting several intermediate restricted Gibbs sampling scans on this lower-level IBP. For each subspace indicator variable u_{tj}, the proposal state is defined as follows:

$$\boldsymbol{\kappa} = \{u_{tj}, v_{tj}, \mathbf{z}_{tj:}, \mathbf{x}_{tj:}\} \tag{4.25}$$

which is composed of all variables associated with the subspace j. Note that there are two possibilities in proposing the new value for the variable u_{tj}. That is, we can turn the value of u_{tj} either **on** or **off** depending on its current value. We denote these proposal states by $\boldsymbol{\kappa}^{\text{on}}$ and $\boldsymbol{\kappa}^{\text{off}}$ accordingly. When updating the state proposal $\boldsymbol{\kappa}$ to $\boldsymbol{\kappa}^*$, the Metropolis-Hasting acceptance probability takes the following form:

$$r_{\boldsymbol{\kappa}\to\boldsymbol{\kappa}^*} = \min\left[1, \frac{q(\boldsymbol{\kappa}|\boldsymbol{\kappa}^*)}{q(\boldsymbol{\kappa}^*|\boldsymbol{\kappa})}\frac{\pi(\boldsymbol{\kappa}^*)}{\pi(\boldsymbol{\kappa})}\right] \tag{4.26}$$

where $\boldsymbol{\kappa}^*$ is either $\boldsymbol{\kappa}^{\text{on}}$ or $\boldsymbol{\kappa}^{\text{off}}$ depending on the type of proposal. The steps in computing the Metropolis-Hasting acceptance probability are described below.

Turn on the subspace $\boldsymbol{\kappa}^{\text{on}}$

If the current state is $u_{tj} = 0$, i.e., the subspace j is inactive, the proposed state is $\boldsymbol{\kappa}^{\text{on}} = \{u_{tj} = 1, v^*_{tj}, \mathbf{z}^*_{tj:}, \mathbf{x}^*_{tj:}\}$. The current values of v_{tj}, $\mathbf{z}_{tj:}$, and $\mathbf{x}_{tj:}$ are neglected because $u_{tj} = 0$ so they do not affect the data. Define the *launch state*, $\boldsymbol{\kappa}^0$, that will be used to compute the Gibbs sampling transition probabilities. To compute the Metropolis-Hastings transition probability, we perform the following steps:

1. Set $u_{tj} = 1$ and initialize v_{tj}, $\mathbf{z}_{tj:}$, and $\mathbf{x}_{tj:}$ from their priors (i.e., conditional probabilities given all other variables except v_{tj}, $\mathbf{z}_{tj:}$, and $\mathbf{x}_{tj:}$). This is the launch state $\boldsymbol{\kappa}^0$.
2. Modify the launch state $\boldsymbol{\kappa}^0$ by performing M intermediate restricted Gibbs sampling scans.
3. Given the launch state $\boldsymbol{\kappa}^0$, we compute the value of v_{tj}, $\mathbf{z}_{tj:}$, and $\mathbf{x}_{tj:}$ by conducting one final Gibbs sampling scan. The final values of the variables constitute the proposed state $\boldsymbol{\kappa}^{\text{on}}$.
4. Calculate the transition probability $q(\boldsymbol{\kappa}^{\text{on}}|\boldsymbol{\kappa})$ by computing the Gibbs sampling transition probability from the launch state $\boldsymbol{\kappa}^0$ to the final proposed state $\boldsymbol{\kappa}^{\text{on}}$. It is the product of the probabilities of setting v_{tj}, $\mathbf{z}_{tj:}$ and $\mathbf{x}_{tj:}$ in the launch state to the final values in the proposed state.
5. To compute the reverse proposal $q(\boldsymbol{\kappa}|\boldsymbol{\kappa}^{\text{on}})$, we can see that if we set u_{tj} to be zero, the corresponding v_{tj}, $\mathbf{z}_{tj:}$ and $\mathbf{x}_{tj:}$ will have no effect on the data. Thus, there is only one way to change u_{tj} from 1 to 0, so $q(\boldsymbol{\kappa}|\boldsymbol{\kappa}^{\text{on}}) = 1$.
6. Calculate the Metropolis-Hastings acceptance probability. If the proposed

state is accepted, we set $u_{tj} = 1$ and set $\boldsymbol{\kappa}^{\text{on}}$ to be the next state. Otherwise, the current state is retained.

Turn off the subspace $\boldsymbol{\kappa}^{\text{off}}$

If the current state is $u_{tj} = 1$, we propose to change its value in the same manner as the previous case. The proposed state can be similarly defined as $\boldsymbol{\kappa}^{\text{off}} = \{u_{tj} = 0, v^*_{tj}, \mathbf{z}^*_{tj:}, \mathbf{x}^*_{tj:}\}$. In contrast, the current values of the variables v_{tj}, $\mathbf{z}_{tj:}$, and $\mathbf{x}_{tj:}$ must also be considered. The sampling method in this case can be described as follows:

1. Set $u_{tj} = 0$ and store the current values of v_{tj}, $\mathbf{z}_{tj:}$, and $\mathbf{x}_{tj:}$.
2. Initialize v_{tj}, $\mathbf{z}_{tj:}$, and $\mathbf{x}_{tj:}$ from their priors and assign them to the launch state $\boldsymbol{\kappa}^0$.
3. Modify the launch state $\boldsymbol{\kappa}^0$ by performing M intermediate restricted Gibbs sampling scans.
4. Calculate the reverse transition probability $q(\boldsymbol{\kappa}|\boldsymbol{\kappa}^{\text{off}})$ by computing the Gibbs sampling transition probability from the launch state $\boldsymbol{\kappa}^0$ to the original state. That is, we need to calculate the probability of generating the original values of v_{tj}, $\mathbf{z}_{tj:}$, and $\mathbf{x}_{tj:}$ from the launch state.
5. The transition probability $q(\boldsymbol{\kappa}^{\text{off}}|\boldsymbol{\kappa})$ is 1.
6. Calculate the Metropolis-Hastings acceptance probability. If the proposed state is accepted, we set $u_{tj} = 0$ and generate the next state $\boldsymbol{\kappa}^{\text{off}}$ from their priors. Otherwise, the current state is retained.

The posterior distribution, $\pi(\boldsymbol{\kappa})$, in Equation (4.26) can be expanded into a product of the prior, $P(\boldsymbol{\kappa})$, and the likelihood, $P(\mathbf{y}_t|\boldsymbol{\kappa})$. The prior distribution, $P(\boldsymbol{\kappa})$, will be a product over variables in $\boldsymbol{\kappa}$, which yields the following prior distribution.

$$P(\boldsymbol{\kappa}) = P(u_{tj})P(v_{tj}) \prod_{z_{tjk}=0} P(z_{tjk}) \prod_{z_{tjk}=1} P(x_{tjk}|z_{tjk})P(z_{tjk}) \tag{4.27}$$

For the $\boldsymbol{\kappa}^{\text{on}}$ proposal, the prior distribution ratio reduces to the following:

$$\begin{aligned} \frac{P(\boldsymbol{\kappa}^{\text{on}})}{P(\boldsymbol{\kappa})} &= \frac{P(u_{tj} = 1)}{P(u_{tj} = 0)} \prod_{z_{tjk}=0} P(z_{tjk}) \prod_{z_{tjk}=1} P(x_{tjk}|z_{tjk})P(z_{tjk}) \\ &= R_{\neg tj} \prod_{z_{tjk}=0} (1 - M_{\neg tjk}) \prod_{z_{tjk}=1} \frac{M_{\neg tjk}}{2} \exp\left(-|x_{tjk}|\right) \end{aligned} \tag{4.28}$$

where $R_{\neg tj} = m_{\neg tj}/(\beta + N - m_{\neg tj})$ and $M_{\neg tjk} = m_{\neg tjk}/(\beta_j + N - 1)$. The $\boldsymbol{\kappa}$ is the original state in which the variable $u_{tj} = 0$. Notice that only the variables $\mathbf{z}_{tj:}$ and $\mathbf{x}_{tj:}$ associated with the proposed state $\boldsymbol{\kappa}^{\text{on}}$ contribute to the ratio of the prior in Equation (4.28) because when $u_{tj} = 0$, these variables do not contribute to the data. Similarly, for the $\boldsymbol{\kappa}^{\text{off}}$ proposal, the prior ratio is given as

$$\begin{aligned}\frac{P(\boldsymbol{\kappa}^{\text{off}})}{P(\boldsymbol{\kappa})} &= \frac{P(u_{tj}=0)}{P(u_{tj}=1)} \prod_{z_{tjk}=0} \frac{1}{P(z_{tjk})} \prod_{z_{tjk}=1} \frac{1}{P(x_{tjk}|z_{tjk})P(z_{tjk})} \\ &= \frac{1}{R_{\neg tj}} \prod_{z_{tjk}=0} \frac{1}{1 - M_{\neg tjk}} \prod_{z_{tjk}=1} \frac{2}{M_{\neg tjk} \exp\left(-|x_{tjk}|\right)} \qquad (4.29)\end{aligned}$$

where $\boldsymbol{\kappa}$ represents the original state in which the subspace is active, i.e., $u_{tj} = 1$. Note that the variables $\mathbf{z}_{tj:}$ and $\mathbf{x}_{tj:}$ used to evaluate the ratios in Equation (4.28) and (4.29) are different. In Equation (4.28), the values of these variables obtained from the final Gibbs sampling scan is used, whereas Equation (4.29) uses their original values in the initial state $\boldsymbol{\kappa}$.

The likelihood for the $\boldsymbol{\kappa}^{\text{on}}$ proposal is in Equation(4.1) evaluated with the values obtained from the final Gibbs sampling scan. In contrast, the likelihood for $\boldsymbol{\kappa}$ is evaluated without the values of those variables, which can therefore be written as

$$P(\mathbf{y}_t|\boldsymbol{\kappa}) = \frac{1}{(2\pi\sigma_\epsilon^2)^{\frac{D}{2}}} \exp\left(-\frac{\boldsymbol{\epsilon}_{t\neg j}\boldsymbol{\epsilon}_{t\neg j}^{\mathsf{T}}}{2\sigma_\epsilon^2}\right) \qquad (4.30)$$

where $\boldsymbol{\epsilon}_{t\neg j}$ is the error $\mathbf{y}_t - (\mathbf{s}_t \circ \mathbf{q}_t \circ \mathbf{z}_t \circ \mathbf{x}_t)\mathbf{A}$ evaluated with $u_{tj} = 0$. As a result, the likelihood ratio in the case of turning on the subspace is given as

$$\frac{P(\mathbf{y}_t|\boldsymbol{\kappa}^{\text{on}})}{P(\mathbf{y}_t|\boldsymbol{\kappa})} = \exp\left(-\frac{1}{2\sigma_\epsilon^2}(\mathbf{q}_{tj:} \circ \mathbf{z}_{tj:} \circ \mathbf{x}_{tj:})\mathbf{A}_j(\mathbf{A}_j^{\mathsf{T}}(\mathbf{q}_{tj:} \circ \mathbf{z}_{tj:} \circ \mathbf{x}_{tj:})^{\mathsf{T}} - 2\boldsymbol{\epsilon}_{t\neg j}^{\mathsf{T}})\right) \qquad (4.31)$$

Similarly, the likelihood ratio for the case of turning off the subspace can be derived as follows:

$$\frac{P(\mathbf{y}_t|\boldsymbol{\kappa}^{\text{off}})}{P(\mathbf{y}_t|\boldsymbol{\kappa})} = \exp\left(-\frac{1}{2\sigma_\epsilon^2}(\mathbf{q}_{tj:} \circ \mathbf{z}_{tj:} \circ \mathbf{x}_{tj:})\mathbf{A}_j(2\boldsymbol{\epsilon}_{t\neg j}^{\mathsf{T}} - \mathbf{A}_j^{\mathsf{T}}(\mathbf{q}_{tj:} \circ \mathbf{z}_{t[j]} \circ \mathbf{x}_{t[j]})^{\mathsf{T}})\right) \qquad (4.32)$$

As in Equation (4.28) and (4.29), the values of v_{tj}, $\mathbf{z}_{tj:}$, and $\mathbf{x}_{tj:}$ used to evaluated the likelihood ratios in Equation (4.31) and (4.32) are obtained from the final Gibbs sampling scan and the original state, respectively.

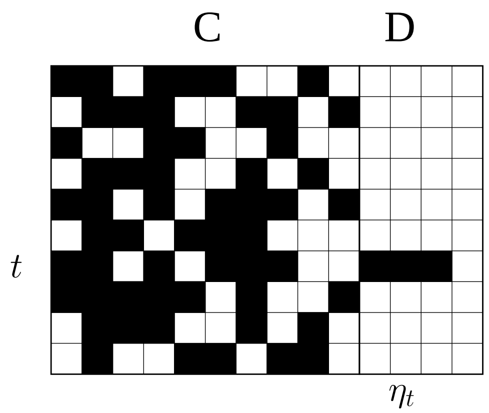


Figure 4.2 – Two sets of columns of binary matrix $\mathbf{Z}_j$

4.9 Update the number of sources

To update the number of sources, the columns of binary matrix $\mathbf{Z}_j$ are partitioned into two sets, C and D as shown in Figure 4.2. In set C, the sources are at least active at some data points other than t. On the other hand, the sources in set D are active only at data point t. Moreover, this set also includes the infinite number of inactive sources. Since they do not affect the data, the inactive sources in this set can be ignored. Therefore, we update the number of sources by replacing the current active sources in set D by the new sources.

Let η_t be the number of columns of $\mathbf{Z}_j$ which contain 1 only in row t. We propose the new sources by sampling Δ_t, the new number of features uniformly from $\{-b, -(b-1), ..., -1, 1, ..., b-1, b\}$ for the integer $b \geq 1$. Given Δ_t, the proposed state $\boldsymbol{\kappa}^*_{\text{src}}$ consisting of $\mathbf{x}^*_t$ and $\mathbf{A}^*$, respectively the new columns of $\mathbf{x}_t$ and the new rows of $\mathbf{A}$, are formed. Note that there are two possible proposed states, one in which $\Delta_t > 0$ (adding sources) and another in which $\Delta_t < 0$ (removing sources). These two proposal states are denoted as $\boldsymbol{\kappa}^+_{\text{src}}$ and $\boldsymbol{\kappa}^-_{\text{src}}$, respectively. Then the restricted Gibbs sampling is performed depending on the type of the proposed state. We use the Metropolis-Hastings update described in Section 4.8 to decide whether or not to accept the proposed state. The Metropolis-Hasting acceptance probability in this case takes the following form:

$$r_{\boldsymbol{\kappa}_{\text{src}} \to \boldsymbol{\kappa}^*_{\text{src}}} = \min\left[1, \frac{q(\boldsymbol{\kappa}_{\text{src}}|\boldsymbol{\kappa}^*_{\text{src}})}{q(\boldsymbol{\kappa}^*_{\text{src}}|\boldsymbol{\kappa}_{\text{src}})} \frac{\pi(\boldsymbol{\kappa}^*_{\text{src}})}{\pi(\boldsymbol{\kappa}_{\text{src}})}\right] \tag{4.33}$$

where $\boldsymbol{\kappa}^*$ is either $\boldsymbol{\kappa}^+_{\text{src}}$ or $\boldsymbol{\kappa}^-_{\text{src}}$ depending on the type of proposal. The proposed technique can then be summarized for $b = 2$ as follows:

1. Draw Δ_t uniformly from $\{-2, -1, 1, 2\}$.

2. If $\Delta_t > 0$ then

   (a) Add Δ_t columns to the matrix $\mathbf{Z}_j$ and set all z_{tjk} to 1 for new columns.

   (b) Form the proposal state $\boldsymbol{\kappa}_{\text{src}}^{+} = \{\Delta_t, \mathbf{x}_t^*, \mathbf{A}^*\}$ in which $\mathbf{x}_t^*$ and $\mathbf{A}^*$ are initialised from their conditional probabilities given other variables. This is the launch state $\boldsymbol{\kappa}_{\text{src}}^{0}$.

   (c) Modify the launch state $\boldsymbol{\kappa}_{\text{src}}^{0}$ by performing M intermediate restricted Gibbs sampling scans.

   (d) Given the launch state $\boldsymbol{\kappa}_{\text{src}}^{0}$ obtained from intermediate restricted Gibbs sampling, the proposed state $\boldsymbol{\kappa}_{\text{src}}^{+}$ is formed by conducting one final Gibbs sampling scan on $\mathbf{x}_t^*$ and $\mathbf{A}^*$.

   (e) Calculate the transition probability $q(\boldsymbol{\kappa}_{\text{src}}^{+}|\boldsymbol{\kappa}_{\text{src}})$ by computing the Gibbs sampling transition probability from the launch state $\boldsymbol{\kappa}_{\text{src}}^{0}$ to the final proposed state $\boldsymbol{\kappa}_{\text{src}}^{+}$. The Gibbs sampling transition probability is the product of the probabilities of setting $\mathbf{x}_t^*$ and $\mathbf{A}^*$ in the launch state to their final values in the proposed state.

   (f) To compute the reverse proposal $q(\boldsymbol{\kappa}_{\text{src}}|\boldsymbol{\kappa}_{\text{src}}^{+})$, we can see that this is the probability of removing Δ_t features from set D. As the features in set D are chosen at random for removal, the reverse proposal is simply the product of the probability of picking Δ_t, which is $1/4$ and the probability of randomly picking Δ_t features from the set D, which is simply $1/\binom{\eta_t}{\Delta_t}$.

   (g) Calculate the Metropolis-Hastings acceptance probability. If the proposed state is accepted, the number of sources and corresponding variables are updated. Otherwise, the current state is retained.

3. If $\Delta_t < 0$ then

   (a) Select $|\Delta_t|$ columns in set D at random and store the original values $\mathbf{x}_t^*$ and $\mathbf{A}^*$ associated with the selected columns.

   (b) According to the selected columns, initialize $\mathbf{x}_t^*$ and $\mathbf{A}^*$ from their conditional probabilities and assign them to the launch state $\boldsymbol{\kappa}_{\text{src}}^{0}$.

   (c) Modify the launch state $\boldsymbol{\kappa}_{\text{src}}^{0}$ by performing M intermediate restricted Gibbs sampling scans.

   (d) Calculate the reverse transition probability $q(\boldsymbol{\kappa}_{\text{src}}|\boldsymbol{\kappa}_{\text{src}}^{-})$ by computing the

Gibbs sampling transition probability from the launch state $\boldsymbol{\kappa}^{0}_{\mathrm{src}}$ to the original state. That is, we need to calculate the probability of generating the original values of $\mathbf{x}^{*}_{t}$, and $\mathbf{A}^{*}$ from the launch state.

(e) The transition probability $q(\boldsymbol{\kappa}^{-}_{\mathrm{src}}|\boldsymbol{\kappa}_{\mathrm{src}})$ is the probability of removing Δ_t features at random, which is again $1/(4\binom{\eta_t}{\Delta_t})$.

(f) Calculate the Metropolis-Hastings acceptance probability. If the proposed state is accepted, we remove the selected features. Otherwise, the current state is retained.

The posterior distribution, $\pi(\boldsymbol{\kappa}_{\mathrm{src}})$, in Equation (4.33) can be expanded into a product of prior, $P(\boldsymbol{\kappa}_{\mathrm{src}})$, and the likelihood $P(\mathbf{y}_t|\boldsymbol{\kappa}_{\mathrm{src}})$. The prior will be a product over priors of the variables in $\boldsymbol{\kappa}_{\mathrm{src}}$. This product yields the following prior distribution

$$P(\boldsymbol{\kappa}_{\mathrm{src}}) = P(\eta_t)P(\mathbf{x}'_t)P(\mathbf{A}') \tag{4.34}$$

where η_t is the number of features in set D, $\mathbf{x}'_t$ and $\mathbf{A}'$ are the elements of $\mathbf{x}_t$ and $\mathbf{A}$ corresponding to the sources in set D. From the IBP, the prior on the number of new features η_t is

$$P(\eta_t|\alpha_j,\beta_j) = \mathrm{Poisson}\left(\frac{\alpha_j\beta_j}{\beta_j+N-1}\right) \tag{4.35}$$

The prior on each element of $\mathbf{x}'_t$ and each row of $\mathbf{A}'$ are standard Laplacian $\frac{1}{2}\exp(-|x|)$ and Normal distribution $\mathcal{N}(0,\sigma^2_A\mathbf{I})$. For the $\boldsymbol{\kappa}^{+}_{\mathrm{src}}$ proposal, the prior distribution ratio reduces to the following:

$$\begin{aligned}\frac{P(\boldsymbol{\kappa}^{+}_{\mathrm{src}})}{P(\boldsymbol{\kappa}_{\mathrm{src}})} &= \frac{P(\eta_t+\Delta_t)}{P(\eta_t)}\frac{P(\mathbf{x}^{*}_t)}{P(\mathbf{x}'_t)}\frac{P(\mathbf{A}^{*}_j)}{P(\mathbf{A}'_j)}\frac{P(\mathbf{y}_t|\boldsymbol{\kappa}^{+}_{\mathrm{src}})}{P(\mathbf{y}_t|\boldsymbol{\kappa}_{\mathrm{src}})}\\ &= \frac{P(\eta_t+\Delta_t)}{P(\eta_t)}\frac{P(\mathbf{y}_t|\boldsymbol{\kappa}^{+}_{\mathrm{src}})}{P(\mathbf{y}_t|\boldsymbol{\kappa}_{\mathrm{src}})}\left[\prod_{i=1}^{\Delta_t}P(\mathbf{x}^{+}_{t,i})\right]\left[\prod_{i=1}^{\Delta_t}P(\mathbf{A}^{+}_{j,i})\right]\end{aligned} \tag{4.36}$$

where $\mathbf{x}^{*}_t$ and $\mathbf{A}^{*}_j$ are the elements of $\mathbf{x}_t$ and $\mathbf{A}_j$ associated with the sources in set D after adding new sources, respectively. The new elements of $\mathbf{x}_t$ and rows of $\mathbf{A}_j$ are represented by $\mathbf{x}^{+}_t$ and $\mathbf{A}^{+}_j$, respectively. As the prior of η_t is Poisson distribution given in Equation (4.35). The prior ratio for the new number of sources in Equation (4.36) is

$$\frac{P(\eta_t+\Delta_t)}{P(\eta_t)} = \left(\frac{\alpha_j\beta_j}{\beta_j+N-1}\right)^{\Delta_t}\frac{1}{(\eta_t+1)...(\eta_t+\Delta_t)} \tag{4.37}$$

The likelihood ratio in Equation (4.36) can be derived similarly as in the previous section, which is given as

$$\frac{P(\mathbf{y}_t|\boldsymbol{\kappa}^{+}_{\mathrm{src}})}{P(\mathbf{y}_t|\boldsymbol{\kappa}_{\mathrm{src}})} = \exp\left(-\frac{1}{2\sigma^2_\epsilon}\mathbf{x}^{+}_t\mathbf{A}^{+}_j(\mathbf{A}^{+\mathsf{T}}_j\mathbf{x}^{+\mathsf{T}}_t - 2\boldsymbol{\epsilon}^{\mathsf{T}}_t)\right) \tag{4.38}$$

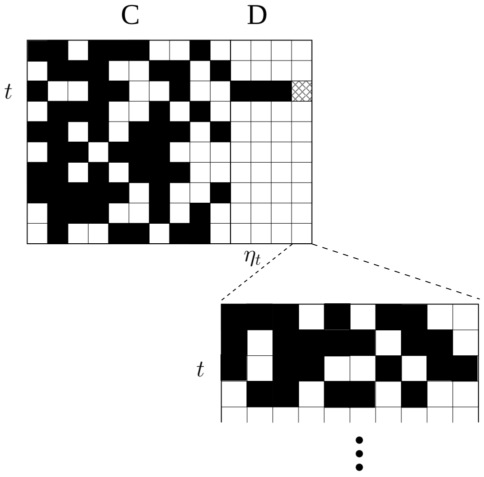


Figure 4.3 – A new subspace is added. This corresponds to the new column of binary matrix $\mathbf{Z}$ in the upper-level IBP and the new binary matrix $\mathbf{Z}_j$ for the lower-level IBPs.

Similarly, the prior distribution ratio for the $\boldsymbol{\kappa}_{\text{src}}^{-}$ proposal reduces to the following:

$$\begin{aligned}\frac{P(\boldsymbol{\kappa}_{\text{src}}^{-})}{P(\boldsymbol{\kappa}_{\text{src}})} &= \frac{P(\eta_t-\Delta_t)}{P(\eta_t)}\frac{P(\mathbf{x}_t^{*})}{P(\mathbf{x}_t^{\prime})}\frac{P(\mathbf{A}_j^{*})}{P(\mathbf{A}_j^{\prime})}\frac{P(\mathbf{y}_t|\boldsymbol{\kappa}_{\text{src}}^{-})}{P(\mathbf{y}_t|\boldsymbol{\kappa}_{\text{src}})} \\ &= \frac{P(\eta_t-\Delta_t)}{P(\eta_t)}\frac{P(\mathbf{y}_t|\boldsymbol{\kappa}_{\text{src}}^{-})}{P(\mathbf{y}_t|\boldsymbol{\kappa}_{\text{src}})}\frac{1}{\prod_{i=1}^{\Delta_t}P(\mathbf{x}_{t,i}^{-})\prod_{i=1}^{\Delta_t}P(\mathbf{A}_{j,i}^{-})}\end{aligned} \quad (4.39)$$

where $\mathbf{x}_t^{*}$ and $\mathbf{A}_j^{*}$ in Equation (4.39) are the elements of $\mathbf{x}_t$ and $\mathbf{A}_j$ associated with the sources in set D after removing the selected sources, respectively. The notations $\mathbf{x}_t^{-}$ and $\mathbf{A}_j^{-}$ represent the elements of $\mathbf{x}_t$ and rows of $\mathbf{A}_j$ associated with the sources that is to be removed, respectively. The prior ratio of the number of sources in this case is

$$\frac{P(\eta_t-\Delta_t)}{P(\eta_t)} = \left(\frac{\beta_j+N-1}{\alpha_j\beta_j}\right)^{\Delta_t}\eta_t(\eta_t-1)...(\eta_t-\Delta_t+1) \quad (4.40)$$

and the likelihood can be written as

$$\frac{P(\mathbf{y}_t|\boldsymbol{\kappa}_{\text{src}}^{-})}{P(\mathbf{y}_t|\boldsymbol{\kappa}_{\text{src}})} = \exp\left(-\frac{1}{2\sigma_\epsilon^2}\mathbf{x}_t^{-}\mathbf{A}_j^{-}(2\boldsymbol{\epsilon}_t^{\mathsf{T}}-\mathbf{A}_j^{-\mathsf{T}}\mathbf{x}_t^{-\mathsf{T}})\right) \quad (4.41)$$

4.10 Update the number of subspaces

The MCMC sampling method to create new sources in Section 4.9 is also applied in updating the number of subspaces. In this case, the new binary vectors $\mathbf{z}_{tj}$ where j ranges over the new subspaces and the new elements of $\mathbf{x}_t$ and $\mathbf{A}$ which correspond to the new subspaces are also included in the proposal.

Figure 4.3 illustrates the case when a new subspace is added which can be seen as the allocation of the new IBP in the set of lower-level IBPs. Although the restricted

Gibbs sampling is performed only at data point t in the lower-level IBP, the possessions of sources in the new subspace for all other data points are also necessary. Thus the entries of binary matrix $\mathbf{Z}_j$ for new subspace j are initialised from the IBP prior.

We follow the same procedure described in the previous section to update the number of subspaces. The only differences are the number of subspaces that will be updated, i.e., added or removed, at each iteration. Since there are a number of variables associated with the Metropolis-Hasting update, the restricted Gibbs sampler has to be conducted in a high dimensional space, which requires a great deal of computation. Thus a single subspace will be either added or removed at a time. In this case, the proposed state $\boldsymbol{\kappa}^*_{\text{subs}}$ consists of v^*_{tj}, $\mathbf{z}^*_{tj:}$, $\mathbf{x}^*_{tj:}$, and $\mathbf{A}^*_j$, which are the variables associated with the updated subspace. Similarly to the case of sources, the proposed states $\boldsymbol{\kappa}^+_{\text{subs}}$ and $\boldsymbol{\kappa}^-_{\text{subs}}$ represent adding and removing subspace proposal, respectively. The Metropolis-Hasting acceptance probability in this case takes the following form:

$$r_{\boldsymbol{\kappa}_{\text{subs}} \to \boldsymbol{\kappa}^*_{\text{subs}}} = \min\left[1, \frac{q(\boldsymbol{\kappa}_{\text{subs}}|\boldsymbol{\kappa}^*_{\text{subs}})}{q(\boldsymbol{\kappa}^*_{\text{subs}}|\boldsymbol{\kappa}_{\text{subs}})} \frac{\pi(\boldsymbol{\kappa}^*_{\text{subs}})}{\pi(\boldsymbol{\kappa}_{\text{subs}})}\right] \tag{4.42}$$

where $\boldsymbol{\kappa}^*$ is either $\boldsymbol{\kappa}^+_{\text{subs}}$ or $\boldsymbol{\kappa}^-_{\text{subs}}$ depending on the type of proposal. The MCMC sampling method for updating the number of subspaces is summarized below.

1. Draw Δ_t uniformly from $\{-1, 1\}$.
2. If $\Delta_t > 0$ then
   (a) Add Δ_t columns to the matrix $\mathbf{U}$ and set the new value of u_{tj} to 1.
   (b) Form the proposal state $\boldsymbol{\kappa}^+_{\text{subs}} = \{\Delta_t, v^*_t, \mathbf{z}^*_{tj:}, \mathbf{x}^*_{tj:}, \mathbf{A}^*\}$ in which all variables are initialised from their conditional probabilities given other variables. This is the launch state $\boldsymbol{\kappa}^0_{\text{subs}}$.
   (c) Modify the launch state $\boldsymbol{\kappa}^0_{\text{subs}}$ by performing M intermediate restricted Gibbs sampling scans.
   (d) Given the launch state $\boldsymbol{\kappa}^0_{\text{subs}}$ obtained from intermediate restricted Gibbs sampling, conduct one final Gibbs sampling scan to obtain the final values of associated variables. These variables constitute the proposed state $\boldsymbol{\kappa}^+_{\text{subs}}$.
   (e) Calculate the transition probability $q(\boldsymbol{\kappa}^+_{\text{subs}}|\boldsymbol{\kappa}_{\text{subs}})$ by computing the Gibbs sampling transition probability from the launch state $\boldsymbol{\kappa}^0_{\text{subs}}$ to the final proposed state $\boldsymbol{\kappa}^+_{\text{subs}}$. The Gibbs sampling transition probability is the product

of the probabilities of setting all variables in the launch state to the final values in the proposed state.

(f) To compute the reverse proposal $q(\boldsymbol{\kappa}_{\text{subs}}|\boldsymbol{\kappa}^{+}_{\text{subs}})$, we can see that this is the probability of removing Δ_t features from set D of the binary matrix $\mathbf{U}$. As only a single column in set D are chosen at random for removal, the reverse proposal is simply the probability of picking one column in set D, which is $1/\eta_t$.

(g) Calculate the Metropolis-Hastings acceptance probability. If the proposed state is accepted, we set $\boldsymbol{\kappa}^{+}_{\text{subs}}$ to be the next state. Otherwise, the current state is retained.

3. If $\Delta_t < 0$ then

(a) Select $|\Delta_t|$ columns in set D at random and store the current values of v_{tj}, $\mathbf{z}_{tj:}$, $\mathbf{x}_{tj:}$, and $\mathbf{A}_j$.

(b) Initialise v_{tj}, $\mathbf{z}_{tj:}$, $\mathbf{x}_{tj:}$ and $\mathbf{A}_j$ from their conditional probabilities and assign them to the launch state $\boldsymbol{\kappa}^{0}_{\text{subs}}$.

(c) Modify the launch state $\boldsymbol{\kappa}^{0}_{\text{subs}}$ by performing M intermediate restricted Gibbs sampling scans.

(d) Calculate the reverse transition probability $q(\boldsymbol{\kappa}_{\text{subs}}|\boldsymbol{\kappa}^{-}_{\text{subs}})$ by computing the Gibbs sampling transition probability from the launch state $\boldsymbol{\kappa}^{0}_{\text{subs}}$ to the original state. That is, we need to calculate the probability of generating the original values of v_{tj}, $\mathbf{z}_{tj:}$, $\mathbf{x}_{tj:}$, and $\mathbf{A}_j$ from the launch state.

(e) The transition probability $q(\boldsymbol{\kappa}^{-}_{\text{subs}}|\boldsymbol{\kappa}_{\text{src}})$ is the probability of removing $|\Delta_t| = 1$ features at random, which is again $1/\eta_t$.

(f) Calculate the Metropolis-Hastings acceptance probability. If the proposed state is accepted, we remove the selected features. Otherwise, the current state is retained.

Using the same analysis as in the previous section, the prior of $\boldsymbol{\kappa}_{\text{subs}}$ can be written as

$$P(\boldsymbol{\kappa}_{\text{subs}}) = P(\eta_t)P(v_{tj})\left[\prod_{z_{tjk}=0} P(z_{tjk})\right]\left[\prod_{z_{tjk}=1} P(\mathbf{x}_{tjk}, \mathbf{a}_{jk}|z_{tjk})P(z_{tjk})\right] \tag{4.43}$$

where η_t is the number of columns in set D of the matrix $\mathbf{U}$ in the upper-level IBP.

The corresponding variables v_{tj}, $\mathbf{z}_{tj:}$, $\mathbf{x}_{tj:}$, and $\mathbf{A}_j$ constitute the lower-level IBP that is to be added or removed. The prior on the number of new features η_t can be similarly obtained from the IBP as follows:

$$P(\eta_t|\alpha,\beta) = \text{Poisson}\left(\frac{\alpha\beta}{\beta+N-1}\right) \tag{4.44}$$

where α and β are the parameters of upper-level IBP. The prior on other variables are already shown in the previous section. Furthermore, the prior of v_{tj} is standard uniform distribution, so it can be neglected from the product in Equation (4.43).

For the $\boldsymbol{\kappa}^+_{\text{subs}}$ proposal, the prior distribution ratio reduces to the following:

$$\frac{P(\boldsymbol{\kappa}^+_{\text{subs}})}{P(\boldsymbol{\kappa}_{\text{subs}})} = \frac{P(\eta_t+1)}{P(\eta_t)}\left[\prod_{z^+_{tjk}=0} P(z^+_{tjk})\right]\left[\prod_{z^+_{tjk}=1} P(x^+_{tjk},\mathbf{a}^+_{jk}|z^+_{tjk})P(z^+_{tjk})\right] \tag{4.45}$$

where the new elements of $\mathbf{z}_t$, $\mathbf{x}_t$ and rows of $\mathbf{A}_j$ are represented by $\mathbf{z}_t^+$, $\mathbf{x}_t^+$ and $\mathbf{A}_j^+$, respectively. As the prior of η_t is Poisson distribution given in Equation (4.44), the prior ratio for the new number of subspaces in Equation (4.45) is

$$\frac{P(\eta_t+1)}{P(\eta_t)} = \left(\frac{\alpha\beta}{\beta+N-1}\right)\frac{1}{(\eta_t+1)} \tag{4.46}$$

The likelihood ratio can also be derived as in the previous section, which is given as

$$\frac{P(\mathbf{y}_t|\boldsymbol{\kappa}^+_{\text{subs}})}{P(\mathbf{y}_t|\boldsymbol{\kappa}_{\text{subs}})} = \exp\left(-\frac{1}{2\sigma_\epsilon^2}(\mathbf{q}_t^+\circ\mathbf{z}_t^+\circ\mathbf{x}_t^+)\mathbf{A}_j^+({\mathbf{A}_j^+}^\mathsf{T}(\mathbf{q}_t^+\circ\mathbf{z}_t^+\circ\mathbf{x}_t^+)^\mathsf{T} - 2\boldsymbol{\epsilon}_t^\mathsf{T})\right) \tag{4.47}$$

Similarly, the prior distribution ratio for the $\boldsymbol{\kappa}^-_{\text{src}}$ proposal reduces to the following:

$$\frac{P(\boldsymbol{\kappa}^-_{\text{subs}})}{P(\boldsymbol{\kappa}_{\text{subs}})} = \frac{P(\eta_t-1)/P(\eta_t)}{\left[\prod_{z^-_{tjk}=0} P(z^-_{tjk})\right]\left[\prod_{z^-_{tjk}=1} P(x^-_{tjk},\mathbf{a}^-_{jk}|z^-_{tjk})P(z^-_{tjk})\right]} \tag{4.48}$$

where $\mathbf{z}_t^-$, $\mathbf{x}_t^-$ and $\mathbf{A}_j^-$ represents the elements of $\mathbf{z}_t$, $\mathbf{x}_t$ and rows of $\mathbf{A}_j$ associated with the subspace that is to be removed, respectively. The prior ratio of the number of subspaces in this case is

$$\frac{P(\eta_t-1)}{P(\eta_t)} = \left(\frac{\beta+N-1}{\alpha\beta}\right)\eta_t \tag{4.49}$$

and the likelihood can be written as

$$\frac{P(\mathbf{y}_t|\boldsymbol{\kappa}^-_{\text{subs}})}{P(\mathbf{y}_t|\boldsymbol{\kappa}_{\text{subs}})} = \exp\left(-\frac{1}{2\sigma_\epsilon^2}(\mathbf{q}_t^-\circ\mathbf{z}_t^-\circ\mathbf{x}_t^-)\mathbf{A}_j^-(2\boldsymbol{\epsilon}_t^\mathsf{T} - {\mathbf{A}_j^-}^\mathsf{T}(\mathbf{q}_t^-\circ\mathbf{z}_t^-\circ\mathbf{x}_t^-)^\mathsf{T})\right) \tag{4.50}$$

4.11 IBP parameter α_j

Recall that the marginal probability of the binary matrix $\mathbf{Z}_j$ for $j = 1, ..., J$ can be written as

$$P(\mathbf{Z}_j|\alpha_j, \beta_j) = \frac{(\alpha_j\beta_j)^{K_+}}{\prod_{h>0} K_h!} \exp(-\alpha_j H_N(\beta_j)) \prod_{k=1}^{K_+} B(m_{jk}, N - m_{jk} + \beta_j) \qquad (4.51)$$

where $H_N(\beta_j) = \sum_{i=1}^{N} \frac{\beta_j}{\beta_j+i-1}$. Taking the log of this expression gives

$$\log P(\mathbf{Z}_j|\alpha_j) = K_+ \log \alpha_j - H_N(\beta_j)\alpha_j + \texttt{const} \qquad (4.52)$$

Using Bayes' rule together with the Gamma prior on α_j we have $P(\alpha_j|\mathbf{A}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}_j, \beta_j) = P(\alpha_j|\mathbf{Z}, \beta_j)$ which is given as

$$\begin{aligned} \log P(\alpha_j|\mathbf{Z}_j, \beta_j) &\propto \log P(\mathbf{Z}_j|\alpha_j, \beta_j) + \log P(\alpha_j) \\ &= (K_+ + \vartheta_j - 1)\log \alpha - \left(H_N(\beta_j) + \frac{1}{\lambda_j}\right)\alpha_j + \texttt{const} \qquad (4.53) \end{aligned}$$

Thus the conditional distribution of α_j can be written as

$$P(\alpha_j|\mathbf{Z}_j) = \mathcal{G}\left(\alpha_j; K_+ + \vartheta_j, \frac{\lambda_j}{1 + \lambda_j H(\beta_j)}\right) \qquad (4.54)$$

The conditional probability of α for the upper-level IBP can be obtained in a similar manner.

4.12 The relevant parameters

There are few parameters associated with the iISA model. In this section we summarize the important parameters that have to be tuned in order to get an effective inference algorithm. Although there are few more parameters, e.g., the hyperparameters of the priors, they have small effects on the performance of the model. Thus only parameters strongly contributed to the performance of the inference algorithm are mentioned in this section.

Table 4.1 summarizes the parameters of the inference algorithm. All of these parameters are the number of steps to conduct the Gibbs sampling. However, they correspond to different parts of the algorithm and will therefore affect the performance of the algorithm in the different ways. Their effects on the performance of the algorithm will be investigated in Chapter 5. The pseudocode summarizing the inference algorithm of the iISA model is shown in Appendix A.

Table 4.1 – The parameters for MCMC sampler of the iISA model

Parameter	Notation	Description
MAXITER	$\mathrm{M}_{\mathrm{GIBBS}}$	The number of iterations for MCMC sampler of the iISA model.
MAXAS	M_{AS}	The number of restricted Gibbs sampling scans for sampling the active subspace.
MAXSRC	$\mathrm{M}_{\mathrm{SRC}}$	The number of restricted Gibbs sampling scans for updating the number of sources.
MAXSUBS	$\mathrm{M}_{\mathrm{SUBS}}$	The number of restricted Gibbs sampling scans for updating the number of sources.
MAXGIBBS	$\mathrm{M}_{\mathrm{FULL}}$	The number of iterations of Gibbs sampling for the lower-level IBPs.

Chapter 5

Experiments and Analysis

In the experiments, the performance of the iISA is evaluated and compared with the standard ISA algorithm. The iICA algorithm is also tested and compared with the FastICA algorithm (Hyvärinen, 1999a). Firstly, the iICA and iISA models are evaluated using synthetic data generated from the models and compared with the general ICA and ISA algorithms. For the iISA model, the performance of MCMC sampling technique, described in the previous chapter, is also studied by examing the affects of various parameters. Lastly, these algorithms are evaluated on real-world data by applying them on the natural images.

5.1 Experimental setting

Prior to the experiment, a useful preprocessing strategy is to first whiten the observed data. As in the standard ICA, the whitening removes the correlation between mixed components in the observed data using a linear transformation, such as principal component analysis, that makes the data covariance matrix the identity matrix. This ensures that the extracted sources are mutually uncorrelated as any orthogonal basis in the data space defines uncorrelated sources with unit variance. In whitening, the goal is to find the whiten vector $\tilde{\mathbf{x}}$ such that $\mathbb{E}\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{\mathsf{T}}\} = \mathbf{I}$. One method for whitening is the eigenvalue decomposition of the covariance matrix $\mathbf{C} = \mathbf{E}\mathbf{D}\mathbf{E}^{\mathsf{T}}$, where $\mathbf{C}$ is the covariance matrix evaluated using the available samples, $\mathbf{E}$ is the orthogonal matrix of eigenvectors of $\mathbf{C}$, and $\mathbf{D}$ is the diagonal matrix of its eigenvalues. Thus the whiten vector $\tilde{\mathbf{x}}$ can be obtained by

$$\tilde{\mathbf{x}} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^{\mathsf{T}}\mathbf{x} \tag{5.1}$$

where $\mathbf{x}$ is the original observed data vector, which is assumed to be zero-mean. One benefit of whitening is that it reduces the number of parameters to be estimated.

To evaluate the algorithms on synthetic data, samples are generated according to the generative models of the iICA and iISA. For iICA model, all sources are assumed mutually independent, whereas in the iISA model, some can be dependent. Note that the synthetic data with independent sources is the special case of ISA where the number of subspaces equals the number of sources. As opposed to iICA, the iISA model generates data with more complex structure. Therefore, this must be done carefully to ensure that the synthetic data set contains the sufficient number of samples. In the experiments, the iICA and iISA models are compared with FastICA (Hyvärinen, 1999a) and ISA algorithms (Hyvärinen and Hoyer, 2000). In this case, the ground truth data is known so we can compare the inferred $\mathbf{A}$, $\mathbf{X}$, and $\mathbf{Z}$ using the *Amari error* (Amari et al., 1996) for ICA models and *generalised Amari error* for ISA models.

Let $\mathbf{W} = \mathbf{Z} \circ \mathbf{X}$, the iICA model can be written in the noiseless case as

$$\mathbf{Y} = \mathbf{W}\mathbf{A} \tag{5.2}$$

Since ICA can recover the hidden sources only up to sign, arbitrary scaling factors, and arbitrary permutations, the problem in Equation (5.2) can also be written as

$$\mathbf{Y} = \mathbf{W}\mathbf{P}\mathbf{C}\mathbf{P}^{-1}\mathbf{C}^{-1}\mathbf{A} \tag{5.3}$$

where $\mathbf{C}$ and $\mathbf{P}$ are the diagonal matrix and permutation matrix, respectively. Thus, Equation (5.3) shows the indeterminancies of the ICA problems, that is, the sources are scaled by a diagonal matrix $\mathbf{C}$ and permuted by a permutation matrix $\mathbf{P}$. Given the true sources $\mathbf{W}$ and mixing matrix $\mathbf{A}$, the inferred matrices are denoted as $\hat{\mathbf{W}} = \mathbf{W}\mathbf{B}$ and $\hat{\mathbf{A}} = \mathbf{B}^{-1}\mathbf{A}$. The sources can be recovered optimally if $\mathbf{B} = \mathbf{P}\mathbf{C}$. Thus $\mathbf{B}$ can be written as

$$\mathbf{B} = (\mathbf{W}^{\mathsf{T}}\mathbf{W})^{-1}(\mathbf{W}^{\mathsf{T}}\hat{\mathbf{W}}) \tag{5.4}$$

Next, the Amari error can be defined in term of elements b_{ij} of $\mathbf{B}$ as follows:

$$\rho(\mathbf{B}) = \frac{1}{2KK' - K' - K}\left(\sum_{i=1}^{K'}\left(\frac{\sum_{j=1}^{K}|b_{ij}|}{\max_k |b_{ik}|} - 1\right) + \sum_{j=1}^{K}\left(\frac{\sum_{i=1}^{K}|b_{ij}|}{\max_k |b_{kj}|} - 1\right)\right) \tag{5.5}$$

where K is the true number of sources and K' is the inferred number. Because the optimal recovery can be obtained when $\mathbf{B}$ is the permutation matrix, the Amari error thus measures the deviation of $\mathbf{B}$ from the permutation matrix, that is, the sum over rows and columns of the deviation from there only being one main entry per column, normalised so that the maximum error is 1.

The generalization of Amari error for ISA is straightforward. In this case, the optimal matrix $\mathbf{B}$ is the permutation matrix permuting $K_j \times K_j$ block matrices. Deviation from this matrix can be computed as follows. Let $\hat{b}_{ij}$ denote the sum of the absolute values of elements at the intersections of the $i(K_i - 1) + 1, ..., iK_i$ rows and the $j(K_j - 1) + 1, ..., jK_j$ columns of matrix $\mathbf{B} = (\mathbf{W}^{\mathsf{T}}\mathbf{W})^{-1}(\mathbf{W}^{\mathsf{T}}\hat{\mathbf{W}})$. Formally, for $1 \leq i, j \leq J$, let

$$\hat{b}_{ij} = \sum_{p=i(K_i-1)+1}^{iK_i} \sum_{q=j(K_j-1)+1}^{jK_j} |\mathbf{B}_{pq}| \tag{5.6}$$

which can be used to compute the Amari error using Equation (5.5). We see that $\rho(\mathbf{B}) \geq 0$ and it is zero if and only if matrix $\mathbf{B}$ is a permutation matrix permuting $K_j \times K_j$ block matrices.

To evaluate the performance of the iICA with respect to the FastICA algorithm, we apply the iICA and FastICA (Bingham and Hyvärinen, 2000; Hyvärinen, 1999a) algorithm with `tanh` nonlinearity on 30 synthetic data sets. The Amari error for each experiment is then computed. For the experiments on ISA models, the synthetic data is generated in which the hidden sources can be separated into disjointed groups which may contain a distinct number of sources. In this case, the performance between the ISA algorithm and the iISA model is compared. As the number of subspaces and their dimensions need not be specified in advance for the iISA model, the synthetic data generally possesses the groups of sources with different sizes. However, it is not clear how to cope with the data consisting of subspaces with different dimensions in the traditional ISA. Therefore, the dimensions are assumed equal for all subspaces. In the experiments, the ISA algorithm is tested by fixing the actual number of subspaces and varying the subspace dimension of 2, 4, and 8.

For iICA, the one-parameter IBP variant is used in the experiments as it is suggested in the original work of iICA that there is no significant difference between one-parameter and two-parameter variants of the IBP. Specifically, only α is sampled from its conditional probability, whereas $\beta = 1$ for all experiments of iICA. Likewise, we also follow this setting by sampling only parameters α associated with both upper-level

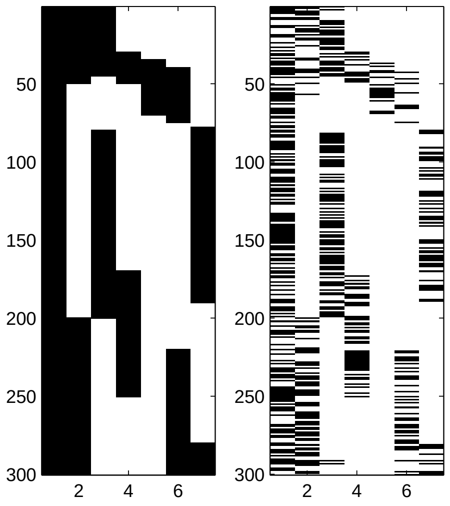


Figure 5.1 – The true and inferred synthetic matrix $\mathbf{Z}$ with $K = 7$ hidden sources

and lower-level IBPs. Although this setting reduces computational cost considerably for the iISA model, it is worth investigating the affects of both α and β as they become more influential in the nested model. However, since the iISA model corresponds to the two-level nIBP, we do not expect much loss of performance in the experiments.

5.2 Results

The results on both synthetic and real-world data are reported in this section. The first part illustrates the results obtained using synthetic data. Both iICA and iISA models are demonstrated to work well compared to the classical ICA and ISA algorithms. Furthermore, the proposed model is examined on several synthetic data sets with different parameter settings. The convergence of MCMC sampling techniques is also observed by using the trace plots of different variants of the iISA models on several synthetic data sets. Lastly, the results of iICA and iISA models on natural images are reported and compared with the classical ICA and ISA algorithms.

5.2.1 Synthetic data

The synthetic data from the iICA model is generated with $N = 300$, $D = 8$, and $K = 7$. The matrices $\mathbf{X}$ and $\mathbf{A}$ are drawn from their priors with binary matrix $\mathbf{Z}$, shown in Figure 5.1. Figure 5.2 depicts the log-likelihood and the parameter values during 1000 iterations of Gibbs sampling on this synthetic data. The Gibbs sampler

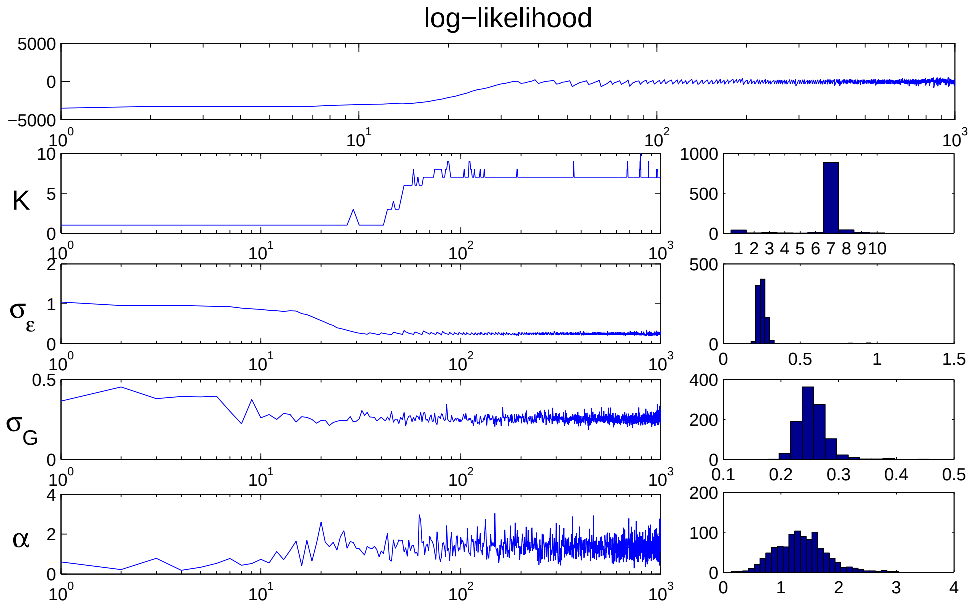


Figure 5.2 – The values of log-likelihood and parameters K, σ_ϵ, σ_A, and α over 1000 iterations of Gibbs sampling on the synthetic data. The histograms on the right depict the distributions of these parameters.

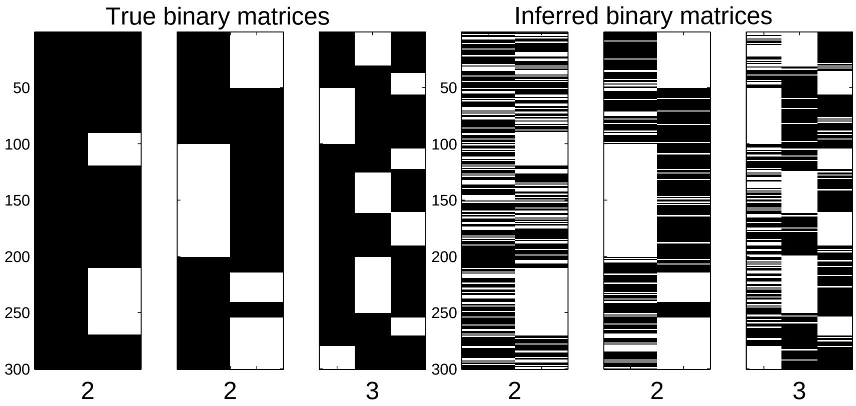


Figure 5.3 – The true matrices (left) and inferred versions (right) of $\mathbf{U}$, $\mathbf{Z}_1$, and $\mathbf{Z}_2$

starts to converge after approximately 30 iterations. The last 500 samples obtained from the Gibbs sampler are used to calculate the average parameter values. As can be seen in the figure, the algorithm can correctly infer the number of hidden sources, which is 7. The true synthetic matrix $\mathbf{Z}$ and the average binary matrix inferred by the algorithm are shown in Figure 5.1.

The synthetic data for iISA model is generated with $N = 300$, $D = 6$, and $J = 2$. The matrices $\mathbf{X}$ and $\mathbf{A}$ are drawn from their priors. The matrices $\mathbf{U}$ and $\mathbf{Z}_j$ for $j = 1, 2$ are shown in Figure 5.3. As shown in the figure, there are 2 subspaces, each of which

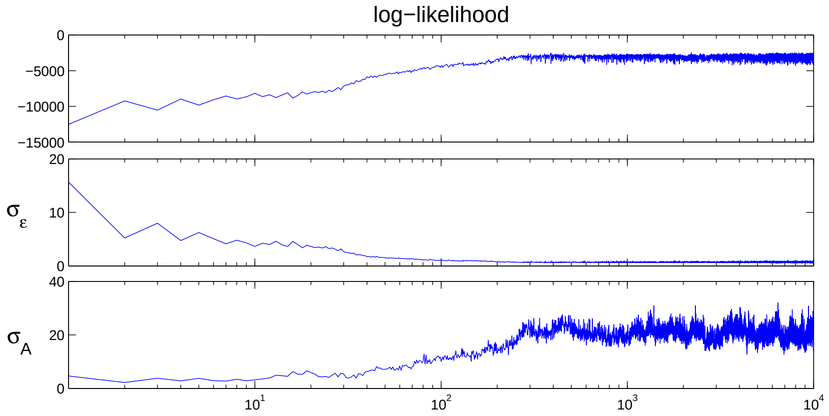


Figure 5.4 – The trace plots of the inference algorithm for iISA model

has a dimension of 2 and 3, respectively. To incorporate the dependencies among the components in each subspace, the data is created as follows. First, 300 samples are drawn from a 6-dimensional Laplace distribution. Since there are 2 subspaces, 300 samples are drawn from a 2-dimensional uniform distribution. Next, the components in each subspace are multiplied by the random variables from the uniform distribution. Due to the common sample from the uniform distribution, the components in the subspace have dependencies. Figure 5.3 shows the synthetic binary matrices corresponding to the upper-level IBP that indicates which subspaces are active for each data point and the binary matrices for each subspace that indicate which sources are active in each subspace.

Using synthetic data shown in Figure 5.3, the inference algorithm is conducted for 10,000 iterations. The parameter setting of the algorithm is $\mathrm{M}_{\mathrm{AS}} = 10$, $\mathrm{M}_{\mathrm{SRC}} = 5$, $\mathrm{M}_{\mathrm{SUBS}} = 5$, and $\mathrm{M}_{\mathrm{FULL}} = 1$. The trace plots obtained from this experiment are depicted in Figure 5.4, which shows only the trace of log-likelihood, σ_ϵ^2, and σ_A^2. According to the log-likelihood values, the variables starts to mix after roughly 300 iterations. The algorithm can infer the correct number of subspaces as well as their dimension as shown in Figure 5.3. However, the mixing matrix inferred by the algorithm seems to be problematic as σ_A^2 becomes larger compared to the true variance used to generate the matrix.

Figure 5.5 shows the boxplots of Amari errors obtained by applying the FastICA algorithm, ISA algorithm, iICA, and iISA on 30 synthetic data sets. Note that the data

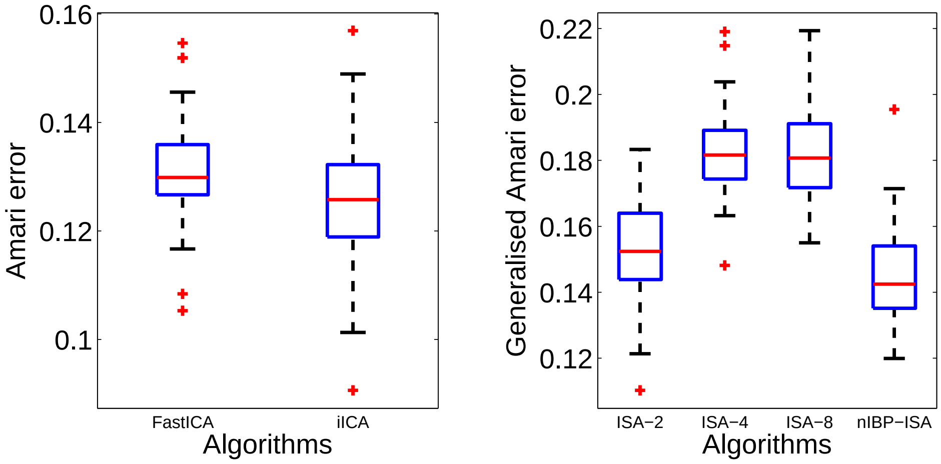


Figure 5.5 – The comparisons between the FastICA algorithm with the iICA (left) and traditional ISA algorithm with iISA (right) over 30 synthetic data sets.

used in these two cases are generated according to different models. We compare the FastICA algorithm versus the iICA and traditional ISA algorithm with different subspace sizes versus iISA. The plot can be interpreted as follows. The rectangular boxes correspond to the second and third quartile of the arranged values. Thus these rectangles correspond to typical medium results occuring in 50% of the realisations. The horizontal line roughly in the middle of each rectangle is the median error separating the second and third quartile for the corresponding algorithm. The red crosses outside this normal range are considered to be outliers.

As can be seen in Figure 5.5, the median Amari errors are similar for both the FastICA and iICA models. This is not surprising, as the sources are heavy-tailed and should be effectively recovered by both algorithms. Note that the true number of sources are used for the FastICA algorithm in all experiments. Although both the FastICA and the iICA algorithms have a similar performance, the iICA model has more flexibility in that the number of sources need not be specified in advance, but can be inferred from the data.

The results for ISA in Figure 5.5 show that the iISA model outperforms the traditional ISA algorithms on the synthetic data. This is intuitive because the data generally has subspaces with uneven number of sources. Thus, the iISA model performs better than traditional ISA algorithm as it can capture the unequal subspace size. This burden

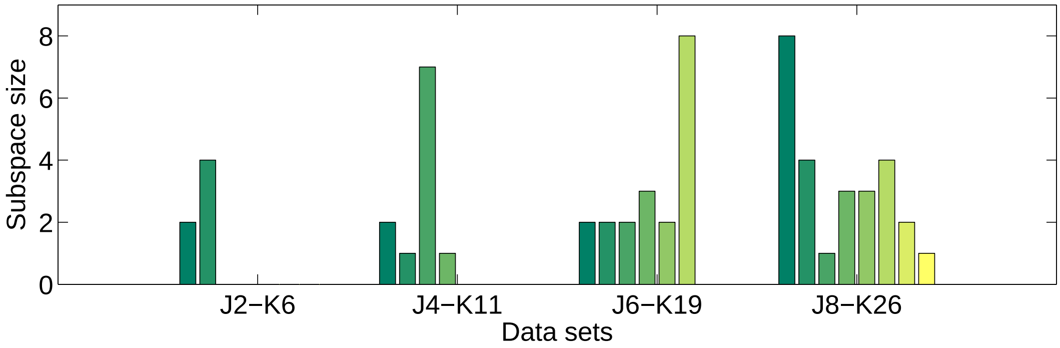


Figure 5.6 – The number of subspaces and the subspace size of four different synthetic data sets.

can sometimes limit the use of ISA algorithms on many real-world applications as the assumption of equal subspace dimensions is not realistic. Furthermore, since almost all subspaces in the synthetic data used in this experiment have low dimensions, the ISA algorithm with the subspace size of 2 yields better results than ISA algorithm with subspace size of 4 and 8.

5.2.1.1 Performance of the iISA algorithm

The MCMC sampling method is similar to the split-merge algorithm in that the intermediate Gibbs sampling scans are performed to make MCMC sampler mix faster. Therefore, different variants of iISA algorithms are compared, which are iISA(1,1,1,0), iISA(1,1,1,1), iISA(1,1,1,10), iISA(10,1,1,1), iISA(1,5,1,1), and iISA(1,1,5,1). The numbers in parentheses are the number of intermediate Gibbs sampling scans for upper-level IBP (M_{AS}), for updating the number of sources ($\mathrm{M}_{\mathrm{SRC}}$), for updating the number of subspaces ($\mathrm{M}_{\mathrm{SUBS}}$), and the number of full Gibbs sampling iterations if the subspace is active ($\mathrm{M}_{\mathrm{FULL}}$), respectively.

The performance of each iISA variant was evaluated by examining the trace plot of log-likelihood and the computation time per iteration as reported in Table 5.1. The variants of iISA are executed on four synthetic data sets with a varying number of subspaces and total number of sources. For four data sets in Table 5.1, the number of subspaces and total number of sources are (2,6), (4,11), (6,19), and (8,26), respectively. Each data set contains 500 data points with 6 attributes. The number of sources in each subspace for these data sets are shown in Figure 5.6.

Table 5.1 reports the average running time per iteration in second for each iISA

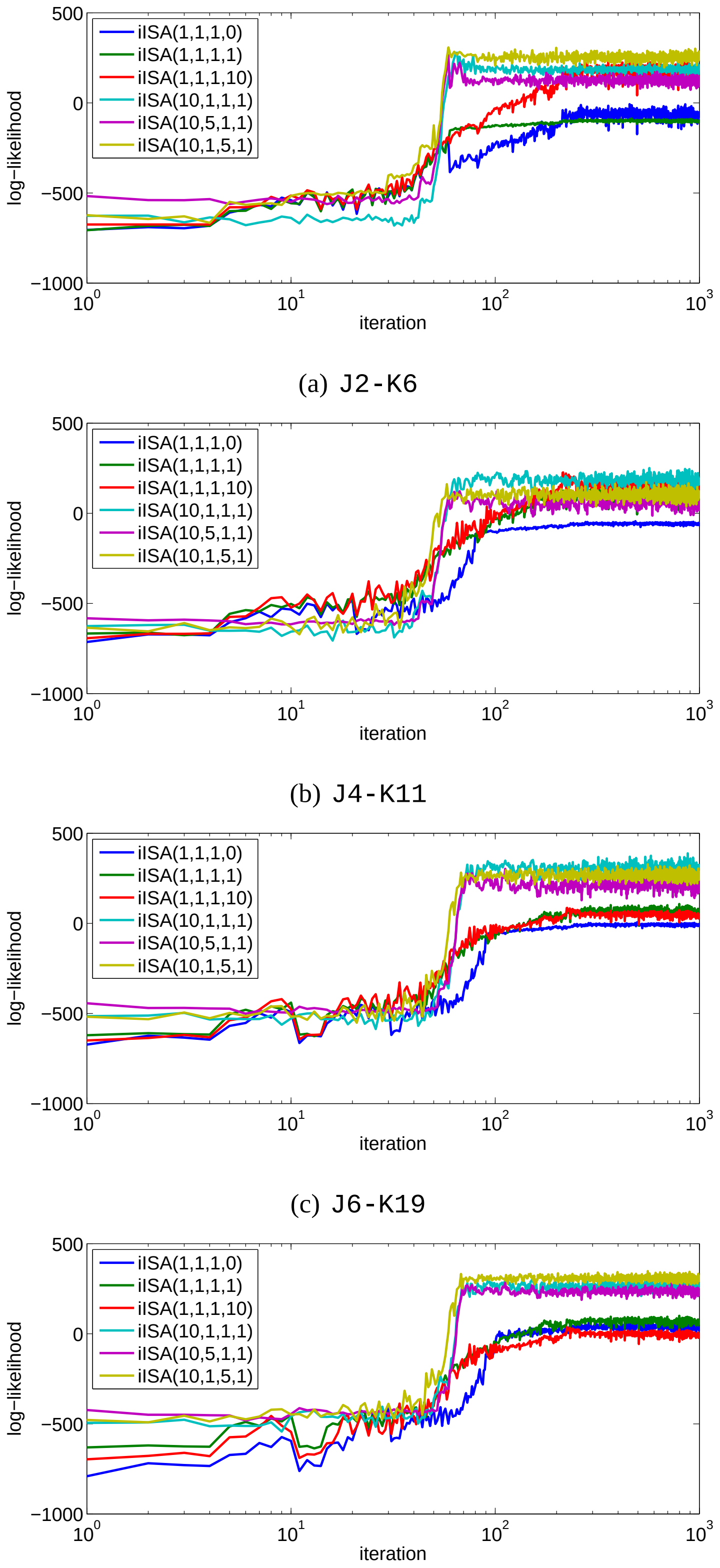
Table 5.1 – The time per iteration in seconds for each algorithm

Algorithm	J2-K6	J4-K11	J6-K19	J8-K26
iISA(1,1,1,0)	1.227	1.732	2.485	2.369
iISA(1,1,1,1)	1.396	1.895	2.422	1.961
iISA(1,1,1,10)	3.101	1.973	3.058	2.570
iISA(10,1,1,1)	2.557	3.045	5.864	4.787
iISA(10,5,1,1)	3.584	3.109	6.558	7.422
iISA(10,1,5,1)	3.320	5.806	5.101	5.475

variant, which illustrates how each parameter affects the running time. As the MCMC sampling method used for iISA model is computationally expensive, choosing appropriate values for these parameters will improve the mixing time of the MCMC samplers, especially for large-scale data sets. The results indicate that the MCMC sampler with a larger number of restricted Gibbs sampling scans and the number of full Gibbs sampling scans spend a longer amount of time. A small increase of the parameter values can incur a considerable amount of computational time. Therefore, it is crucial to look at how much each parameter affects the performance of the algorithm, leading to the appropriate parameter setting.

The trace plots of each iISA variants are shown in Figure 5.7. The convergence of each algorithm is observed by looking at the log-likelihood trace for 1000 iterations. The plots show the log-likelihood of the model using a different parameter setting of the MCMC sampler until it mixes. It is clear from Figure 5.7(a) and 5.7(b) that the 4th-6th variants of iISA mix equally well and relatively faster than the 1st-3rd variants. Although converging more slowly, the 3rd variant also converges to the likelihood values closer to those of the 4th-6th variants, which is contrary to the 1st and 2nd variants that achieve lower likelihood values. According to the likelihood values, the 2nd and 3rd variants yield better results on the `J4-K11` data set, whereas the 1st variant still achieves the lowest likelihood values.

The differences between the 1st-3rd variants and 4th-6th variants are clearly illustrated in Figures 5.7(c) and 5.7(d). The convergence patterns of the 4th-6th variants are similar to the first two cases. However, the 1st-3rd variants mix poorly and achieve the



(a) J2-K6

(b) J4-K11

(c) J6-K19

(d) J8-K26

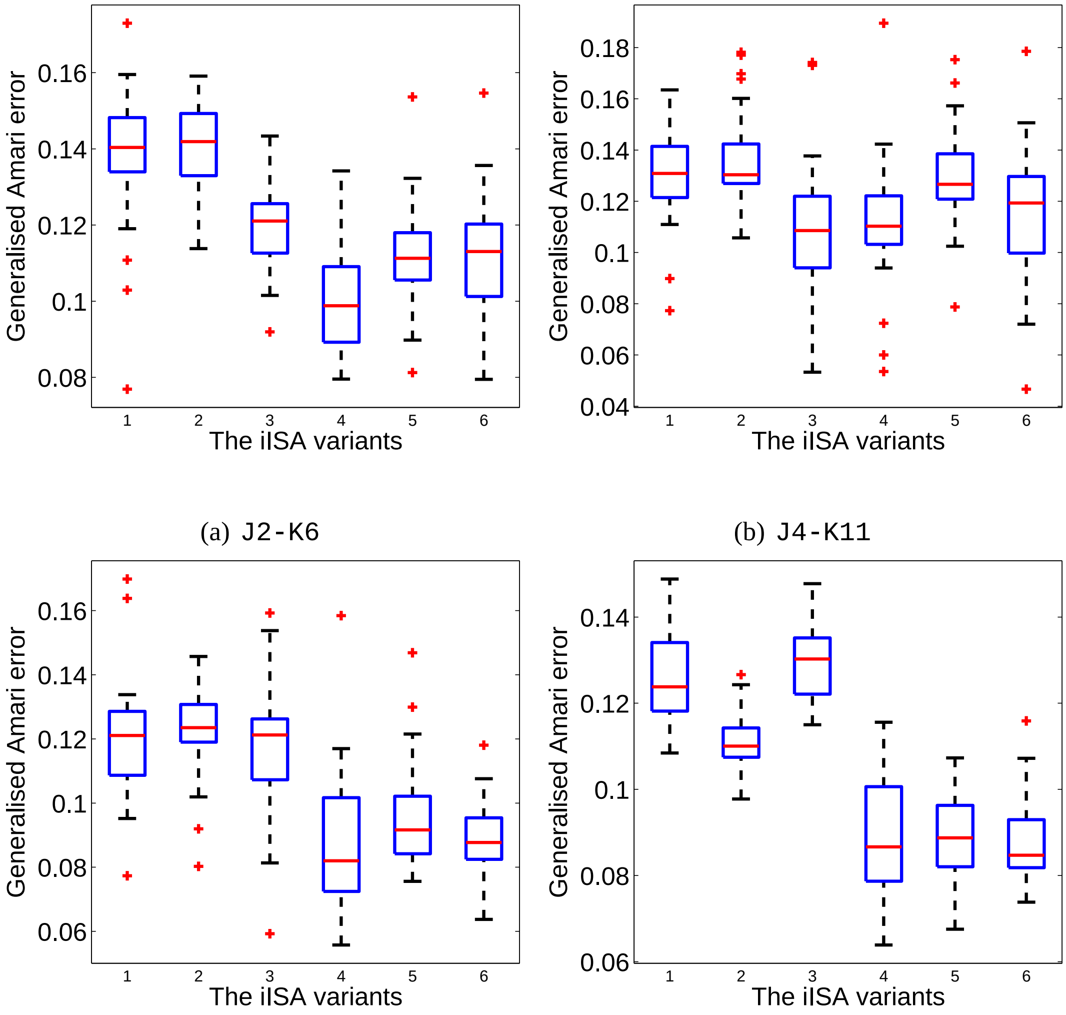
Figure 5.7 – The trace plots of iISA variants on four different synthetic data for 1000 iterations

low likelihood values on these two data sets. Furthermore, the results obtained from the 4th-6th variants are approximately the same in term of likelihood values and mixing time. Intuitively, the MCMC samplers on `J6-K19` and `J8-K26` data sets take slightly longer time before they converge.

The results shown in Figure 5.7 indicate the effects of parameters in the iISA model. The most influential parameter in the MCMC sampler is the number of restricted Gibbs sampling scans before reaching the launch state in the upper-level IBP. The large number of scans ensures that the restricted Gibbs sampler in the lower-level IBPs takes steps toward the optimal region such that the lower-level IBPs improve sufficiently at each iteration. Increasing the number of restricted Gibbs sampling scans also improve the mixing time as the acceptance rate of Metropolis-Hasting update is increased accordingly. Although playing a similar role as the first parameter, the number of restricted Gibbs sampling scans used to update the number of sources and subspaces seems to be less important compared to the first parameter. This is the case because these events occur quite rarely as a result from a relatively low acceptance rate of Metropolis-Hasting update. Lastly, the results also suggest that increasing the number of full Gibbs sampling scans after the Metropolis-Hasting update does not significantly improve the mixing time. Furthermore, some of these issues are discussed later for the comparison of different iISA variants using the generalised Amari error.

To evaluate the results obtained from each iISA variant, 30 synthetic data are generated with the same number of subspaces and their dimensions as shown in Figure 5.6. These six algorithms are then applied on these data sets. The boxplots of generalised Amari errors are shown in Figure 5.8. These plots illustrate the effects of each parameters in the MCMC sampler to the source recovery.

The Amari errors of iISA(1,1,1,0), iISA(1,1,1,1), and iISA(1,1,1,10) are used to examine the affect of number of Gibbs sampling scans after the Metropolis-Hasting step in the upper-level IBP. The results in Figure 5.8 indicate that the full Gibbs sampling scan for a lower-level IBP after Metropolis-Hasting update may be a waste of computational time. Although the performance improves in the first two simple data sets as depicted in Figure 5.8(a) and 5.8(b), there is no significant improvement for more complicated data sets as depicted in Figure 5.8(c) and 5.8(d). This result indeed closely resembles what was observed in the split-merge algorithm (Jain and Neal, 2000) although



(a) J2-K6

(b) J4-K11

(c) J6-K19

(d) J8-K26

Figure 5.8 – the generalised Amari errors of different iISA variants

its full Gibbs sampling scan refers to a different operation. In iISA, the restricted Gibbs sampling is performed on a single lower-level IBP, whereas it is performed for a subset of data points in split-merge algorithm. Thus the full Gibbs sampling scan for an iISA model is indeed similar to restricted Gibbs sampling. Note that this step is important in the case that the acceptance probability of Metropolis-Hasting update is very low and the upper-level IBP is rarely updated. As observed from the experiments, a single step of full Gibbs sampling scans for iISA is more than sufficient.

The results of iISA(10,1,1,1), iISA(10,5,1,1), and iISA(10,1,5,1) are used to examine the affects of the number of restricted Gibbs sampling scans before reaching the launch state for indicating active subspace, updating the number of sources, and

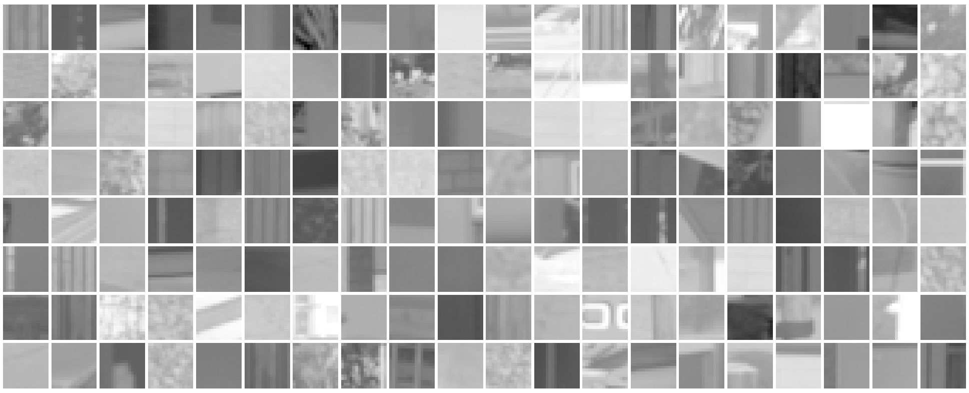


Figure 5.9 – The examples of 160 images patches extracted from the natural images

updating the number of subspaces, respectively. The results indicate the importance of these three parameters as the generalised Amari errors improve considerably in almost all cases. Moreover, the improvement seems to a result of the first parameter, that is, the number of restricted Gibbs sampling scans for indicating the active subspace. A higher number of retricted Gibbs sampling scans increases the acceptance probability of the Metropolis-Hasting update, but also results in longer computational time. Since the last two parameters are relatively less important than the first one, the values of first parameter should be higher than the last two parameters.

5.2.2 Natural images

To evaluate the iISA model on the real data, experiments on the natural images were conducted. The input data matrix $\mathbf{Y}$ is formed by taking the 16×16 pixel image patches at random locations from monochrome photographs depicting wild-life scenes (animals, meadows, and forests). The original images are available on the World Wide Web[1]. Figure 5.9 illustrates some of these image patches. The image patches are then converted into vectors of length 256. The mean gray-scale value of each image patch is subtracted. The dimension of the data is reduced to 160 by PCA, which corresponds to low-pass filtering.

Using image patches, the FastICA, ISA, iICA, and iISA algorithms are compared by observing a set of filters recovered by each algorithm. Each algorithm is executed for 1000 iterations. For FastICA, the number of independent components to be estimated is 160. For ISA, the number of independent subspaces to be estimated and the dimen-

[1] Image available at http://www.cis.hut.fi/projects/ica/data/images/

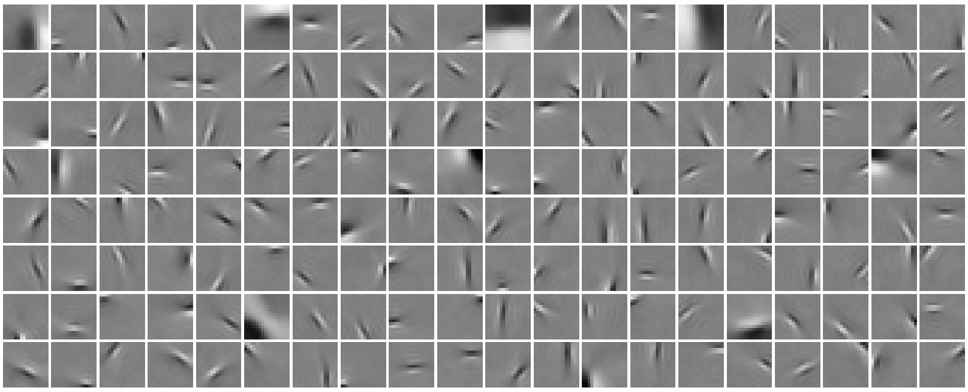


Figure 5.10 – The 160 filters obtained from the ICA algorithm

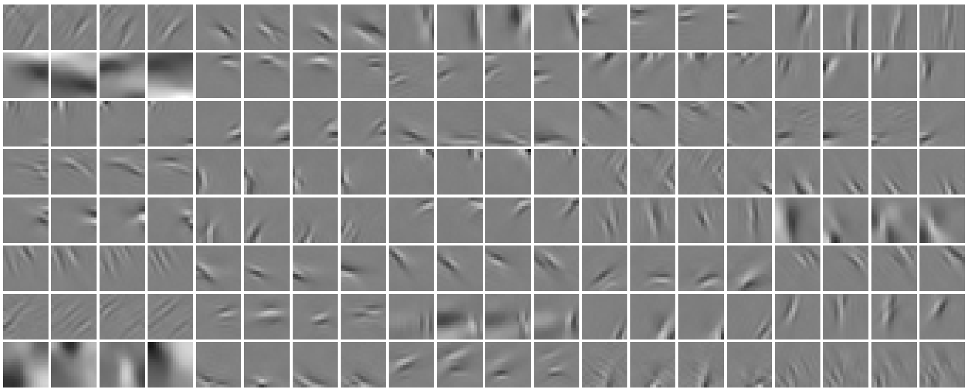


Figure 5.11 – The 160 filters obtained from the ISA algorithm

sionality of subspaces are 40 and 4, respectively. For both iICA and iISA, there is no need to specify the number of independent components nor the number of independent subspaces as they can be determined by the model.

The independent components extracted by ICA algorithm are shown in Figure 5.10. It is well-known that the independent components obtained from ICA are localized both in space and in frequency, resembling the Gabor filters. The results also show that the components produced by ICA are not completely independent. Some components are similar in their orientation. The ISA algorithm groups these components together as illustrated in Figure 5.11. For the iICA and iISA models, similar results are expected, as shown in Figure 5.10 and 5.11, respectively. The components obtained from iICA and iISA models, however, do not resemble the Gabor filters, but have random patterns, so they will not be considered further here. Although the results of iICA and iISA models are not successful, they provide useful insight into how to improve the model, which will be discussed later.

5.3 Analysis

Several issues of iISA model have been identified in the experiments. First, the computational time of iISA seems to be problematic, although the split-merge algorithm improves the mixing time of the sampler. The parameters of the iISA algorithm thus play an important role in the computational time of the algorithm, especially on the real-world data. Other than adjusting the parameters, some other techniques to improve the computational time of the algorithm can also be considered.

The second issue relate to both iICA and iISA models. Sometimes, the algorithms fail to converge on synthetic data. That is, the amount of noise added to the data has to be small such that it does not disturb the true underlying structure. From this observation, the noise variance of approximately 0.1 should be sufficient.

Thirdly, the initialisation of the model is also important, which is more obvious for the case of the iISA model. In the experiment, the number of subspaces is initialised by drawing from the Poisson distribution and the dimensions of all subspaces are initialised to be 1 to simplify the algorithm. Another approach is to draw the dimension of subspace from Poisson distribution parameterised by the parameters associated with each subspace, which yields the subspaces with different dimensions. These two approaches yield similar results on the synthetic data. It is also of interest to consider other initialisation methods that can make the algorithm converge faster.

Lastly, many issues of the iISA model on the natural images have been identified. Since the iICA model in this experiment also failed to recover the filters from the natural images, it is suspected that the structure underlying the natural images is significantly different from the model as defined by iICA. As the iISA model is based on the iICA model, the results obtained are quite similar. Possible improvement of the iISA model include considering another heavy-tail distribution other than Laplacian. More experiments need to be done to gain more insight into how to improve the iISA model. Due to time constraints, it is not possible to evaluate the iISA model on other real world data sets, but it is worth experimenting on various data sets. All issues discussed here will be considered for further improvement of the model.

Chapter 6

Conclusions and Future Works

In this thesis, the Bayesian nonparametrics have been studied, providing a framework for more expressive probabilistic representations. They provide a richer class of distributions over objects such as functions, partitions, trees with infinite branching factors and depths by replacing the finite-dimensional prior distribution with stochastic processes. In addition to the expressiveness, various properties of nonparametric Bayesian models such as exchangeability, permit the design of efficient inference algorithms. The flexibility of Bayesian nonparametrics has attracted researchers from different areas, in which many successful applications have been developed. Some of these recent developments have also been reviewed in this work.

This thesis has focused on one of the Bayesian nonparametric approaches called the Indian buffet process (IBP), which defines a class of infinite latent feature models. The basic premise of the IBP is that objects are modelled as arising from infinitely many latent features, with any particular object having finite features. In this work, the general definition of IBP has been discussed and explored some important properties. Moreover, some recent developments of the IBP and its applications have been reviewed. A theoretical background of IBP in terms of beta process is laid out to aid the understanding of its extensions proposed in this work.

Motivated by the successful applications of nCRP, I discuss a general framework for the nested IBP through beta processes. Although a nesting strategy for IBP is straightforward, theoretical results of a nesting strategy in Bayesian nonparametrics has not yet been seen. Additionally, there are no concrete applications of nested IBP. To gain some insights into the nested models, I have reviewed recent developments of nDP and nCRP, with the connections to the nBP and nIBP, respectively. Some important properties of nIBP have been illustrated through examples which provide a motivation for further developments of applications toward this direction.

To give a concrete application of nIBP, I propose the infinite Independent Subspace Analysis (iISA) which provides the open-ended complexity to the classical independent subspace analysis models. The ISA models generalise the independent component analysis by allowing dependencies between source signals. The independence assumption is instead imposed on the groups of dependent source signals. One drawback of most ISA models is that the number of groups, as well as the group size, need to be specified prior to learning. Moreover, it is not clear how to deal with the case of different group sizes. Although some recent developments in ISA address these problems explicitly, there are still some limitations. To fully handle these problems, the iISA model eliminates the restrictions on the number of groups and a groups size by allowing them to be inferred from the data. However, by increasing degrees of freedom, the model becomes more complex. To handle this complexity, we propose a specialised inference algorithm based on the Metropolis-Hasting method. The algorithm is similar to the split-merge algorithm used for the Dirichlet process mixture model to improve the mixing time of the MCMC sampler.

The experimental results have not only demonstrated the performance of the iISA model, but also led to some conceptual insights that could be used to improve the model. Even though the results on real data reveal a considerable number of issues of the iISA model, they also leave some interesting questions involving both conceptual understanding and practical uses. Indeed, this thesis provides a basis for several significant improvements that can be done in the future. Additionally, this study has introduced several ideas that can be useful for a wider range of applications.

There are several issues of iISA model that need to be discussed further. One of them is the computational complexity of the inference algorithm. An efficient inference algorithm plays an important role in the practical use of the iISA model. This can also aid in the development of applications utilising the nIBP. In addition to nCRP, it is also of interest to make connections between nIBP and other Bayesian nonparametric models such as the branching process, which may establish the nIBP in relation to some well-known models in this area.

References

R. Abel, D. B. Dunson, and A. E. Gelfand. The nested dirichlet process. *Journal of the American Statistical Association*, 103(483):1131–1154, 2008.

D. Aldous. Exchangeability and related topics. In *Ecole d'Ete de Probabilities de Saint-Flour XIII 1983*, pages 1–198. Springer, 1985.

S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 757–763, 1996.

J. Austerweil and T. L. Griffiths. Analyzing human feature learning as nonparametric bayesian inference. In *Advances in Neural Information Processing Systems (NIPS)*, pages 97–104, 2008.

F. R. Bach and M. I. Jordan. Finding clusters in independent component analysis. In *Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA)*, pages 891–896, 2003.

A. J. Bell. and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

A. Belouchrani and J.-F. Cardoso. Maximum likelihood source separation by the expectation-maximization technique: Deterministic and stochastic implementation. In *Proc. NOLTA*, pages 49–53, 1995.

E. Bingham and A. Hyvärinen. A fast fixed-point algorithm for independent component analysis of complex valued signals. *Int. J. Neural Syst.*, 10(1):1–8, 2000.

D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *J. ACM*, 57(2), 2010.

J.-F. Cardoso. Infomax and maximum likelihood for blind source separation, 1997.

J.-F. Cardoso. Multidimensional independent component analysis. In *In Proc. Int. Workshop on Higher-Order Stat*, pages 111–120, 1998.

W. Chu, Z. Ghahramani, R. Krause, and D. L. Wild. Identifying protein complexes in high-throughput protein interaction screens using an infinite latent feature model. In *Proceedings of the Pacific Symposium (BIOCOMPUTING)*, pages 231–242, 2006.

P. Comon. Independent component analysis, a new concept? *Signal Process.*, 36(3): 287–314, 1994.

A. Courville, D. Eck, and Y. Bengio. An infinite factor model hierarchy via a noisy-or mechanism. In *Advances in Neural Information Processing Systems (NIPS)*, pages 405–413. 2009.

T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, 1991.

P. J. Cowans. Information retrieval using hierarchical dirichlet processes. In *Proceedings of Conference on Research and Development in Information Retrieval*, pages 564–565, 2004.

P. J. Cowans. *Probabilistic Document Modelling*. PhD thesis, University of Cambridge, 2006.

F. Doshi-Velez and Z. Ghahramani. Correlated non-parametric latent feature models. In *Proceedings of the Conference in Uncertainty in Artificial Intelligence (UAI)*, 2009.

Z. Ghahramani, P. Sollich, and T. L. Griffiths. Bayesian nonparametric latent feature models. In *Bayesian Statistics 8*. University Press, 2007.

D. Görür, F. Jäkel, and C. E. Rasmussen. A choice model with infinitely many latent features. In *Proceedings of International Conference on Machine Learning (ICML*, pages 361–368, 2006.

T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. In *Advances in Neural Information Processing Systems (NIPS)*, pages 475–482. MIT Press, 2005a.

T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. Technical report, Gatsby Computational Neuroscience Unit, 2005b.

H. W. Gutch and F. J. Theis. Independent subspace analysis is unique, given irreducibility. In *Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA)*, pages 49–56, 2007.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

K. A. Heller and Z. Ghahramani. A nonparametric bayesian approach to modeling overlapping clusters, 2007.

A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999a.

A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999b.

A. Hyvärinen and P. O. Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000.

S. Jain and R. Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182, 2000.

C. Jutten and J. Herault. Blind separation of sources, part 1: an adaptive algorithm based on neuromimetic architecture. *Signal Process.*, 24(1):1–10, 1991.

J. Karvanen, J. Eriksson, and V. Koivunen. Maximum likelihood estimation of ica model for wide class of source distributions. In *Signal Processing*, pages 445–454, 2000.

J. Kingman. Completely random measures. *Pacific J. Math*, 21(1):59–78, 1967.

D. Knowles and Z. Ghahramani. Infinite sparse factor analysis and infinite independent components analysis. In *Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA)*, pages 381–388, 2007.

E. Meeds, Z. Ghahramani, R. M. Neal, and S. T. Roweis. Modeling dyadic data with binary latent factors. In *Advances in Neural Information Processing Systems (NIPS)*, pages 977–984, 2006.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

K. Miller, T. Griffiths, and M. I. Jordan. Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1276–1284, 2009.

K. T. Miller, T. L. Griffiths, and M. I. Jordan. The phylogenetic indian buffet process: A non-exchangeable nonparametric prior for latent features. In *Proceedings of the Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 403–410, 2008.

D. J. Navarro and T. L. Griffiths. Latent features in similarity judgments: A nonparametric bayesian approach. *Neural Computation*, 20(11):2597–2628, 2008.

K. Ni, L. Carin, and D. B. Dunson. Multi-task learning for sequential data via ihmms and the nested dirichlet process. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 689–696, 2007.

J. Paisley and L. Carin. Nonparametric factor analysis with beta process priors. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 777–784, 2009.

B. A. Pearlmutter and L. C. Parra. Maximum likelihood blind source separation: A context-sensitive generalization of ica. In *Advances in Neural Information Processing Systems (NIPS)*, pages 613–619, 1997.

D. T. Pham. Separation of a mixture of independent sources through a maximum likelihood approach. In *Proc. EUSIPCO*, pages 771–774, 1992.

B. Póczos and A. Lörincz. Independent subspace analysis using *k*-nearest neighborhood distances. In *Proceedings of International Conference on Artificial Neural Networks (ICANN)*, pages 163–168, 2005.

P. Rai and H. Daumé III. The infinite hierarchical factor regression model. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2008.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

Y. W. Teh and D. Görür. Indian buffet processes with power-law behavior. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

Y. W. Teh and M. I. Jordan. Hierarchical Bayesian nonparametric models with applications. In N. Hjort, C. Holmes, P. Müller, and S. Walker, editors, *To appear in Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, 2010.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Y. W. Teh, D. Görür, and Z. Ghahramani. Stick-breaking construction for the Indian buffet process. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 11, 2007.

F. J. Theis. Uniqueness of complex and multidimensional independent component analysis. *Signal Process.*, 84(5):951–956, 2004.

F. J. Theis. Towards a general independent subspace analysis. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

R. Thibaux and M. I. Jordan. Hierarchical beta processes and the indian buffet process. Technical report, In Practical Nonparametric and Semiparametric Bayesian Statistics, 2007.

F. Wood and T. L. Griffiths. Particle filtering for nonparametric bayesian matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1513–1520, 2006.

F. Wood, T. Griffiths, and Z. Ghahramani. A non-parametric bayesian method for inferring hidden causes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 536–543, 2006.

E. P. Xing, K.-A. Sohn, M. I. Jordan, and Y. W. Teh. Bayesian multi-population haplotype inference via a hierarchical Dirichlet process mixture. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 23, 2006.

Appendices

Appendix A

iISA Algorithm

Several parts of the proposed MCMC sampling technique are illustrated by the pseudocode. Section A.1-A.3 present the sub-algorithms which constitute the iISA algorithm shown in the Section A.4.

A.1 Active subspaces

The algorithm below shows the steps to perform the intermediate Gibbs sampling for the active subspace. The relevant parameter for this procedure is MaxAS (M_{AS}).

Algorithm 1 Intermediate Gibbs sampler for active subspaces

```
define the proposal state as $\boldsymbol{\kappa} = \{v_{tj}, \mathbf{z}_{tj:}, \mathbf{x}_{tj:}\}$
if $u_{tj} = 0$ then
    initialise $\boldsymbol{\kappa}$ from the conditional distributions
    for $m = 1$ to $\mathrm{M}_{\mathrm{AS}}$ do
        perform restricted Gibbs sampling scan on $\boldsymbol{\kappa}$
    end for
    perform one final restricted Gibbs sampling on $\boldsymbol{\kappa}$ to get $\boldsymbol{\kappa}^*$
    compute the Metropolis-Hasting acceptance probability $r_{\boldsymbol{\kappa} \to \boldsymbol{\kappa}^*}$
else
    store the original state $\boldsymbol{\kappa}^* \leftarrow \boldsymbol{\kappa}$
    initialise $\boldsymbol{\kappa}$ from the conditional distributions
    for $m = 1$ to $\mathrm{M}_{\mathrm{AS}}$ do
        perform restricted Gibbs sampling scan on $\boldsymbol{\kappa}$
    end for
    compute the Metropolis-Hasting acceptance probability $r_{\boldsymbol{\kappa} \to \boldsymbol{\kappa}^*}$
    store the new state $\boldsymbol{\kappa}^* \leftarrow \boldsymbol{\kappa}$
end if
return the new state $\boldsymbol{\kappa}^*$ and the acceptance probability $r_{\boldsymbol{\kappa} \to \boldsymbol{\kappa}^*}$
```

A.2 Update number of sources

The procedure used to perform the intermediate Gibbs sampling to update the number of sources in each subspaces is shown below. The relevant parameter for this procedure is MAXSRC ($\mathrm{M}_{\mathrm{SRC}}$) that specifies the number of restricted Gibbs sampling scans from launch state to the final state.

Algorithm 2 Intermediate Gibbs sampler for updating number of sources

```
sample $\Delta_t$ from $\{-2, -1, 1, 2\}$
define the proposal state as $\boldsymbol{\kappa}_{\mathrm{src}} = \{\Delta_t, \mathbf{x}_t^*, \mathbf{A}^*\}$
if $\Delta_t > 0$ then
    add new elements of $\mathbf{x}_t$ and rows of $\mathbf{A}$ and store in $\mathbf{x}_t^*$ and $\mathbf{A}^*$
    initialise $\boldsymbol{\kappa}_{\mathrm{src}}$ from the conditional distributions
    for $m = 1$ to $\mathrm{M}_{\mathrm{SRC}}$ do
        perform restricted Gibbs sampling scan on $\boldsymbol{\kappa}_{\mathrm{src}}$
    end for
    perform one final restricted Gibbs sampling on $\boldsymbol{\kappa}_{\mathrm{src}}$ to get $\boldsymbol{\kappa}^*_{\mathrm{src}}$
    compute the Metropolis-Hasting acceptance probability $r_{\boldsymbol{\kappa}_{\mathrm{src}} \to \boldsymbol{\kappa}^*_{\mathrm{src}}}$
else
    select $|\Delta_t|$ elements of $\mathbf{x}_t$ and rows of $\mathbf{A}$ corresponding to the features in set $D$ of binary matrix
    $\mathbf{Z}_j$ and store in $\mathbf{x}_t^*$ and $\mathbf{A}^*$
    store the original state $\boldsymbol{\kappa}^*_{\mathrm{src}} \leftarrow \boldsymbol{\kappa}_{\mathrm{src}}$
    initialise $\boldsymbol{\kappa}_{\mathrm{src}}$ from the conditional distributions
    for $m = 1$ to $\mathrm{M}_{\mathrm{SRC}}$ do
        perform restricted Gibbs sampling scan on $\boldsymbol{\kappa}_{\mathrm{src}}$
    end for
    compute the Metropolis-Hasting acceptance probability $r_{\boldsymbol{\kappa}_{\mathrm{src}} \to \boldsymbol{\kappa}^*_{\mathrm{src}}}$
    store the new state $\boldsymbol{\kappa}^*_{\mathrm{src}} \leftarrow \boldsymbol{\kappa}_{\mathrm{src}}$
end if
return the new state $\boldsymbol{\kappa}^*_{\mathrm{src}}$ and the acceptance probability $r_{\boldsymbol{\kappa}_{\mathrm{src}} \to \boldsymbol{\kappa}^*_{\mathrm{src}}}$
```

A.3 Update number of subspaces

The procedure used to perform the intermediate Gibbs sampling to update the number of subspaces is shown below. The relevant parameter for this procedure is MaxSubs (M_{SUBS}) that specifies the number of restricted Gibbs sampling scans from launch state to the final state. The algorithm is basically the same as in updating the number of sources.

Algorithm 3 Intermediate Gibbs sampler for updating number of subspaces

```
sample $\Delta_t$ from $\{-1, 1\}$
define the proposal state as $\boldsymbol{\kappa}_{\text{subs}} = \{\Delta_t, v^*_{tj}, \mathbf{z}^*_{tj:}, \mathbf{x}^*_{tj:}, \mathbf{A}^*_j\}$ for subspace $j$
if $\Delta_t > 0$ then
    add new elements of $\mathbf{z}_t$, $\mathbf{x}_t$ and rows of $\mathbf{A}$ and store in $bz^*_{tj:}$, $\mathbf{x}^*_{tj:}$ and $\mathbf{A}^*_j$
    initialise $\boldsymbol{\kappa}_{\text{subs}}$ from the conditional distributions
    for $m = 1$ to $\text{M}_{\text{SUBS}}$ do
        perform restricted Gibbs sampling scan on $\boldsymbol{\kappa}_{\text{subs}}$
    end for
    perform one final restricted Gibbs sampling on $\boldsymbol{\kappa}_{\text{subs}}$ to get $\boldsymbol{\kappa}^*_{\text{subs}}$
    compute the Metropolis-Hasting acceptance probability $r_{\boldsymbol{\kappa}_{\text{subs}} \rightarrow \boldsymbol{\kappa}^*_{\text{subs}}}$
else
    select $|\Delta_t|$ features from set $D$ of matrix $\mathbf{U}$ randomly. Store the elements of $\mathbf{z}_t$, $\mathbf{x}_t$ and rows of
    $\mathbf{A}_j$ corresponding to the selected features in $\mathbf{z}^*_t$, $\mathbf{x}^*_t$, and $\mathbf{A}^*$
    store the original state $\boldsymbol{\kappa}^*_{\text{subs}} \leftarrow \boldsymbol{\kappa}_{\text{subs}}$
    initialise $\boldsymbol{\kappa}_{\text{subs}}$ from the conditional distributions
    for $m = 1$ to $\text{M}_{\text{SUBS}}$ do
        perform restricted Gibbs sampling scan on $\boldsymbol{\kappa}_{\text{subs}}$
    end for
    compute the Metropolis-Hasting acceptance probability $r_{\boldsymbol{\kappa}_{\text{subs}} \rightarrow \boldsymbol{\kappa}^*_{\text{subs}}}$
    store the new state $\boldsymbol{\kappa}^*_{\text{subs}} \leftarrow \boldsymbol{\kappa}_{\text{subs}}$
end if
return the new state $\boldsymbol{\kappa}^*_{\text{subs}}$ and the acceptance probability $r_{\boldsymbol{\kappa}_{\text{src}} \rightarrow \boldsymbol{\kappa}^*_{\text{subs}}}$
```

A.4 MCMC sampler for iISA

Algorithm 4 shows the pseudocode for the inference algorithm of the iISA model.

Algorithm 4 MCMC sampler for iISA model

```
Initialise $\mathbf{U}$, $\mathbf{V}$, $\mathbf{X}$, $\mathbf{A}$, $\mathbf{V}$, and $\mathbf{Z}_j$, $\forall j$ from their priors.
for $m = 1$ to $\mathrm{M}_{\mathrm{GIBBS}}$ do
    for $t = 1$ to $N$ do
        for $j = 1$ to $J$ do
            if $m_{\neg tj} > 0$ then
                compute the new state $\boldsymbol{\kappa}^*$ using Algorithm 1
                if the new state $\boldsymbol{\kappa}^*$ is accepted then
                    change the value of $u_{tj}$ and set the corresponding variables to new state $\boldsymbol{\kappa}^*$
                else
                    if $u_{tj} = 1$ then
                        for $k = 1$ to $\mathrm{M}_{\mathrm{FULL}}$ do
                            sample $v_{tj}$, $\mathbf{z}_{tj:}$, and $\mathbf{x}_{tj:}$ from their conditional probabilities
                        end for
                    end if
                end if
            end if
        end for
        for each active subspace $j$ do
            compute the new state $\boldsymbol{\kappa}^*_{\mathrm{src}}$ using Algorithm 2
            if the new state $\boldsymbol{\kappa}^*_{\mathrm{src}}$ is accepted then
                either add or remove the source in subspace $j$ according to $\boldsymbol{\kappa}^*_{\mathrm{src}}$
            end if
        end for
        compute the new state $\boldsymbol{\kappa}^*_{\mathrm{subs}}$ using Algorithm 3
        if the new state $\boldsymbol{\kappa}^*_{\mathrm{subs}}$ is accepted then
            either add or remove the subspace according to $\boldsymbol{\kappa}^*_{\mathrm{subs}}$
        end if
    end for
    for $k = 1$ to $K$ do
        sample each row $\mathbf{a}_k$ of mixing matrix $\mathbf{A}$
    end for
    remove columns with $m_j = 0$ from $\mathbf{U}$ and $m_{jk} = 0$ from $\mathbf{Z}_j$
    remove corresponding elements from $\mathbf{V}$, $\mathbf{X}$, and $\mathbf{A}$
    sample $\sigma^2_\epsilon$, $\sigma^2_A$, $\alpha$, and $\alpha_j$ from their conditional distributions
end for
```

Appendix B

Conditional Probabilities

This section contains the detail derivation of conditional probability distribution for some variables mentioned in Chapter 4.

B.1 The hidden source $\mathbf{x}_t$

Let $\boldsymbol{\epsilon}_{tj\neg k}$ be the error vector $\boldsymbol{\epsilon}_t$ evaluated when $z_{tjk} = 0$. From Bayes' rule, we have

$$P(x_{tjk}|\mathbf{A}, \mathbf{s}_t, \mathbf{q}_t, \mathbf{z}_t, \mathbf{x}_{tj\neg k}) \propto P(\mathbf{y}_t|\mathbf{A}, \mathbf{s}_t, \mathbf{q}_t, \mathbf{z}_t, \mathbf{x}_t, \sigma_\epsilon^2)P(x_{tjk}) \tag{B.1}$$

Taking log of (B.1) gives

$$\begin{aligned}
\log P(x_{tjk}|\text{rest}) &= -|x_{tjk}| - \frac{1}{2\sigma_\epsilon^2}(\boldsymbol{\epsilon}_{tj\neg k} - \mathbf{a}_{jk}x_{tjk})(\boldsymbol{\epsilon}_{tj\neg k} - \mathbf{a}_{jk}x_{tjk})^\mathsf{T} + \texttt{const} \\
&= -|x_{tjk}| - \frac{1}{2\sigma_\epsilon^2}(x_{tjk}^2\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T} - 2\mathbf{a}_{jk}\boldsymbol{\epsilon}_{tj\neg k}^\mathsf{T}x_{tjk} + \boldsymbol{\epsilon}_{tj\neg k}\boldsymbol{\epsilon}_{tj\neg k}^\mathsf{T}) + \texttt{const} \\
&= -\frac{1}{2\sigma_\epsilon^2}\left(x_{tjk}^2\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T} - 2\mathbf{a}_{jk}\boldsymbol{\epsilon}_{tj\neg k}^\mathsf{T}x_{tjk} + 2\sigma_\epsilon^2|x_{tjk}| + \boldsymbol{\epsilon}_{tj\neg k}\boldsymbol{\epsilon}_{tj\neg k}^\mathsf{T}\right) + \texttt{const} \\
&= -\frac{\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T}}{2\sigma_\epsilon^2}\left(x_{tjk}^2 - 2x_{tjk}\frac{\mathbf{a}_{jk}\boldsymbol{\epsilon}_{tj\neg k}^\mathsf{T} \mp \sigma_\epsilon^2}{\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T}} + \frac{\boldsymbol{\epsilon}_{tj\neg k}\boldsymbol{\epsilon}_{tj\neg k}^\mathsf{T}}{\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T}}\right) + \texttt{const} \qquad \text{(B.2)}
\end{aligned}$$

which is the piecewise Gaussian distribution given as

$$P(x_{tjk}|\mathbf{A}, \mathbf{s}_t, \mathbf{q}_t, \mathbf{z}_t, \mathbf{x}_{tj\neg k}) = \begin{cases} \frac{\Upsilon_+}{\Omega}\mathcal{N}(x_{tjk}; \mu_+, \sigma) & \text{if } x_{tjk} > 0 \\ \frac{\Upsilon_-}{\Omega}\mathcal{N}(x_{tjk}; \mu_-, \sigma) & \text{if } x_{tjk} < 0 \end{cases} \tag{B.3}$$

where the means μ_+, μ_- and variance σ^2 can be defined as follows:

$$\mu_+ = \frac{\mathbf{a}_{jk}\boldsymbol{\epsilon}_{tj\neg k}^\mathsf{T} - \sigma_\epsilon^2}{\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T}}, \ \ \mu_- = \frac{\mathbf{a}_{jk}\boldsymbol{\epsilon}_{tj\neg k}^\mathsf{T} + \sigma_\epsilon^2}{\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T}}, \ \ \sigma^2 = \frac{\sigma_\epsilon^2}{\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T}} \tag{B.4}$$

This distribution is guaranteed to be continuous by the choice of Υ_+ and Υ_- and to be correctly normalized by the choice of Ω.

$$\begin{aligned}\Upsilon_+ &= \mathcal{N}(0;\mu_-,\sigma) && \text{(B.5)}\\ \Upsilon_- &= \mathcal{N}(0;\mu_+,\sigma) && \text{(B.6)}\\ \Omega &= \Omega_-\Upsilon_- + \Omega_+\Upsilon_+ && \text{(B.7)}\end{aligned}$$

where $\Omega_- = F(0;\mu_-,\sigma)$ and $\Omega_+ = 1 - F(0;\mu_+,\sigma)$.

B.2 The source dependency $\mathbf{v}_t$

The conditional probability of the source dependency $\mathbf{q}_{tj:}$, i.e., v_{tj}, can be rewritten as:

$$\begin{aligned}P(\mathbf{q}_{tj:}|\mathbf{A},\mathbf{x}_t,\mathbf{z}_t,\mathbf{s}_t,\mathbf{q}_{t\neg(j:)}) &\propto P(\mathbf{y}_t|\mathbf{A},\mathbf{x}_t,\mathbf{z}_t,\mathbf{s}_t,\mathbf{q}_t,\sigma_\epsilon^2)P(\mathbf{q}_{tj:}|\mathbf{q}_{t\neg(j:)})\\ &\propto P(\mathbf{y}_t|\mathbf{A},\mathbf{x}_t,\mathbf{z}_t,\mathbf{s}_t,\mathbf{q}_t,\sigma_\epsilon^2)\mathbb{I}_{\{0<\mathbf{q}_{tj:}<1\}} && \text{(B.8)}\end{aligned}$$

Replacing $\mathbf{q}_{tj:}$ with v_{tj}, we can derive the likelihood term in (4.8) as follows:

$$\begin{aligned}\mathcal{L} &= \frac{1}{(2\pi\sigma_\epsilon^2)^{\frac{D}{2}}}\exp\left\{-\frac{1}{2\sigma_\epsilon^2}(\mathbf{y}_t-(\mathbf{s}_t\circ\mathbf{q}_t\circ\mathbf{z}_t\circ\mathbf{x}_t)\mathbf{A})(\mathbf{y}_t-(\mathbf{s}_t\circ\mathbf{q}_t\circ\mathbf{z}_t\circ\mathbf{x}_t)\mathbf{A})^\mathsf{T}\right\}\\ &= \frac{1}{(2\pi\sigma_\epsilon^2)^{\frac{D}{2}}}\exp\left\{-\frac{1}{2\sigma_\epsilon^2}(\boldsymbol{\epsilon}_{t\neg(j:)}-v_{tj}(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j)(\boldsymbol{\epsilon}_{t\neg(j:)}-v_{tj}(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j)^\mathsf{T}\right\}\\ &= \frac{1}{(2\pi\sigma_\epsilon^2)^{\frac{D}{2}}}\exp\left\{-\frac{1}{2\sigma_\epsilon^2}(\boldsymbol{\epsilon}_{t\neg(j:)}\boldsymbol{\epsilon}_{t\neg(j:)}^\mathsf{T}-2v_{tj}(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j\boldsymbol{\epsilon}_{t\neg(j:)}^\mathsf{T}\right.\\ &\quad\left.+v_{tj}^2(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j\mathbf{A}_j^\mathsf{T}(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})^\mathsf{T})\right\}\\ &= \frac{1}{(2\pi\sigma_\epsilon^2)^{\frac{D}{2}}}\exp\left\{-\frac{(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j\mathbf{A}_j^\mathsf{T}(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})^\mathsf{T}}{2\sigma_\epsilon^2}\left(v_{tj}^2-\frac{2v_{tj}(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j\boldsymbol{\epsilon}_{t\neg(j:)}^\mathsf{T}}{(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j\mathbf{A}_j^\mathsf{T}(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})^\mathsf{T}}\right.\right.\\ &\quad\left.\left.+\frac{\boldsymbol{\epsilon}_{t\neg(j:)}\boldsymbol{\epsilon}_{t\neg(j:)}^\mathsf{T}}{(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j\mathbf{A}_j^\mathsf{T}(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})^\mathsf{T}}\right)\right\} && \text{(B.9)}\end{aligned}$$

After completing the square in (B.9), we obtain the conditional probability of v_{tj} written as

$$P(v_{tj}|\mathbf{A},\mathbf{x}_t,\mathbf{z}_t,\mathbf{s}_t,\mathbf{v}_{t\neg j}) \propto \mathcal{N}(v_{tj};\mu_v,\sigma_v^2)\mathbb{I}_{\{0<v_{tj}<1\}} \qquad \text{(B.10)}$$

which is two-sided truncated normal distribution with mean μ_v and variance σ_v^2 defined as follows.

$$\mu_v = \frac{(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j\boldsymbol{\epsilon}_{t\neg(j:)}^\mathsf{T}}{(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j\mathbf{A}_j^\mathsf{T}(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})^\mathsf{T}},\quad \sigma_v^2 = \frac{\sigma_\epsilon^2}{(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})\mathbf{A}_j\mathbf{A}_j^\mathsf{T}(\mathbf{z}_{tj:}\circ\mathbf{x}_{tj:})^\mathsf{T}} \qquad \text{(B.11)}$$

To generate samples from two-sided truncated normal distribution, we use the following method given that we know the cumulative distribution function $F(\cdot)$ and the inverse error function $\mathrm{erf}^{-1}(\cdot)$.

$$\phi_l = F\left(\frac{(a-\mu_v)}{\sigma_v};0,1\right),\ \ \phi_r = F\left(\frac{(b-\mu_v)}{\sigma_v};0,1\right) \tag{B.12}$$

$$v = \mu_v + \sigma_v\left(\sqrt{2}\mathrm{erf}^{-1}(2(\phi_l + (\phi_r - \phi_l)w) - 1)\right) \tag{B.13}$$

where a and b are the lower and upper truncation points, respectively, and w is a sample from $\mathrm{Uniform}(a, b)$.

B.3 The mixing matrix A

the conditional probability distribution of the rows $\mathbf{a}_{jk}$ of mixing matrix $\mathbf{A}_j$ can be written as

$$P(\mathbf{a}_{jk}|\mathbf{A}_{\neg(jk)}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{S}, \mathbf{Q}, \sigma_\epsilon^2, \sigma_A^2) \propto P(\mathbf{Y}|\mathbf{A}, \mathbf{X}, \mathbf{Z}, \mathbf{S}, \mathbf{Q}, \sigma_\epsilon^2)P(\mathbf{a}_{jk}|\sigma_A^2) \tag{B.14}$$

Let $\mathbf{E}_{\neg(jk)}$ be the noise matrix $\mathbf{E} = \mathbf{Y} - (\mathbf{S} \circ \mathbf{Q} \circ \mathbf{Z} \circ \mathbf{X})\mathbf{A}$ evaluated when $\mathbf{a}_{jk} = 0$ and denote the column of $(\mathbf{S} \circ \mathbf{Q} \circ \mathbf{Z} \circ \mathbf{X})$ by $\mathbf{w}_{jk}^\mathsf{T}$. Consider the log conditional of $\mathbf{a}_{jk}$ in (B.14).

$$\begin{aligned}
\log P(\mathbf{a}_{jk}|\mathrm{rest}) &= -\frac{1}{2\sigma_\epsilon^2}\mathrm{tr}(\mathbf{Y} - \mathbf{G}\mathbf{A})(\mathbf{Y} - \mathbf{G}\mathbf{A})^\mathsf{T} - \frac{1}{2\sigma_A^2}\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T} + \mathtt{const} \\
&= -\frac{1}{2\sigma_\epsilon^2}\mathrm{tr}(\mathbf{E}_{\neg(jk)} - \mathbf{w}_{jk}\mathbf{a}_{jk})(\mathbf{E}_{\neg(jk)} - \mathbf{w}_{jk}\mathbf{a}_{jk})^\mathsf{T} - \frac{1}{2\sigma_A^2}\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T} + \mathtt{const} \\
&= -\frac{1}{2\sigma_\epsilon^2}\mathrm{tr}(\mathbf{w}_{jk}^\mathsf{T}\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T}\mathbf{w}_{jk} - 2\mathbf{w}_{jk}^\mathsf{T}\mathbf{a}_{jk}\mathbf{E}_{\neg(jk)}^\mathsf{T}) - \frac{1}{2\sigma_A^2}\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T} + \mathtt{const} \\
&= -\frac{1}{2\sigma_\epsilon^2}((\mathbf{w}_{jk}\mathbf{w}_{jk}^\mathsf{T})(\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T}) - 2\mathbf{a}_{jk}\mathbf{E}_{\neg(jk)}^\mathsf{T}\mathbf{w}_{jk}^\mathsf{T}) - \frac{1}{2\sigma_A^2}\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T} + \mathtt{const} \\
&= -\frac{1}{2\sigma_\epsilon^2}\left((\mathbf{w}_{jk}\mathbf{w}_{jk}^\mathsf{T})(\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T}) + \frac{\sigma_\epsilon^2}{\sigma_A^2}\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T} - 2\mathbf{a}_{jk}\mathbf{E}_{\neg(jk)}^\mathsf{T}\mathbf{w}_{jk}^\mathsf{T}\right) + \mathtt{const} \\
&= -\frac{\mathbf{w}_{jk}\mathbf{w}_{jk}^\mathsf{T} + \frac{\sigma_\epsilon^2}{\sigma_A^2}}{2\sigma_\epsilon^2}\left(\mathbf{a}_{jk}\mathbf{a}_{jk}^\mathsf{T} - \frac{2\mathbf{a}_{jk}\mathbf{E}_{\neg(jk)}^\mathsf{T}\mathbf{w}_{jk}^\mathsf{T}}{\mathbf{w}_{jk}\mathbf{w}_{jk}^\mathsf{T} + \frac{\sigma_\epsilon^2}{\sigma_A^2}}\right) + \mathtt{const} \qquad \text{(B.15)}
\end{aligned}$$

Thus the conditional distribution of each row of the mixing matrix given other variables is Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ whose mean and variance are defined as

$$\boldsymbol{\mu} = \frac{\sigma_A^2}{\mathbf{w}_{jk}\mathbf{w}_{jk}^\mathsf{T}\sigma_A^2 + \sigma_\epsilon^2}\mathbf{E}_{\neg(jk)}^\mathsf{T}\mathbf{w}_{jk}^\mathsf{T} \tag{B.16}$$

$$\boldsymbol{\Lambda} = \left(\frac{\mathbf{w}_{jk}\mathbf{w}_{jk}^\mathsf{T}}{\sigma_\epsilon^2} + \frac{1}{\sigma_A^2}\right)\mathbf{I}_{D\times D} \tag{B.17}$$

B.4 The noise variance σ_ϵ^2

The conditional distribution of σ_ϵ^2 can be obtained using Bayes's rule as follow.

$$P(\sigma_\epsilon^2|\mathbf{E},a,b) \propto P(\mathbf{E}|\sigma_\epsilon^2)P(\sigma_\epsilon^2|a,b) \tag{B.18}$$

where the log of the total likelihood (4.2) and the prior are given as

$$\begin{aligned} \log P(\mathbf{E}|\sigma_\epsilon^2) &= -\frac{1}{2\sigma_\epsilon^2}\mathrm{tr}(\mathbf{E}\mathbf{E}^\mathsf{T}) - \frac{ND}{2}\log(2\pi\sigma_\epsilon^2) && \text{(B.19)}\\ \log P(\sigma_\epsilon^2|a,b) &= -(a+1)\log\sigma_\epsilon^2 - \frac{1}{b\sigma_\epsilon^2} - \log\Gamma(a) - a\log b && \text{(B.20)} \end{aligned}$$

Taking log of (B.18) results in the following equation.

$$\begin{aligned} \log P(\sigma_\epsilon^2|\mathbf{E}) &= \log P(\mathbf{E}|\sigma_\epsilon^2) + \log P(\sigma_\epsilon^2|a,b)\\ &= -\frac{1}{2\sigma_\epsilon^2}\mathrm{tr}(\mathbf{E}\mathbf{E}^\mathsf{T}) - \frac{ND}{2}\log(2\pi\sigma_\epsilon^2) - (a+1)\log\sigma_\epsilon^2\\ &\quad -\frac{1}{b\sigma_\epsilon^2} - \log\Gamma(a) - a\log b\\ &= -\left((a+1) + \frac{ND}{2}\right)\log\sigma_\epsilon^2\\ &\quad -\left(\frac{1}{b} + \frac{1}{2}\mathrm{tr}(\mathbf{E}\mathbf{E}^\mathsf{T})\right)\frac{1}{\sigma_\epsilon^2} + \mathtt{const} && \text{(B.21)} \end{aligned}$$

As a result, we have the conditional probability

$$P(\sigma_\epsilon^2|\mathbf{E},a,b) = \mathcal{IG}(\sigma_\epsilon^2; a + \frac{ND}{2}, \frac{b}{1+\frac{b}{2}\mathrm{tr}(\mathbf{E}\mathbf{E}^\mathsf{T})}) \tag{B.22}$$

B.5 The variance of mixing matrix elements

The conditional probability distribution of σ_A^2 can be obtained using Bayes' rule.

$$P(\sigma_A^2|\mathbf{A},c,d) \propto P(\mathbf{A}|\sigma_A^2)P(\sigma_A^2|c,d) \tag{B.23}$$

Similarly to the previous section, the log of conditional can be written as

$$\begin{aligned} \log P(\sigma_A^2|\mathbf{A},c,d) &= \frac{1}{2\sigma_A^2}\mathrm{tr}(\mathbf{A}\mathbf{A}^\mathsf{T}) - \frac{DK}{2}\log(2\pi\sigma_A^2)\\ &\quad -(c+1)\log\sigma_A^2 - \frac{1}{\sigma_A^2 d} && \text{(B.24)} \end{aligned}$$

As a result, we have the conditional probability distribution of σ_A^2 as below.

$$P(\sigma_A^2|\mathbf{A},c,d) = \mathcal{IG}(\sigma_A^2; c + \frac{DK}{2}, \frac{d}{1+\frac{d}{2}\mathrm{tr}(\mathbf{A}\mathbf{A}^\mathsf{T})}) \tag{B.25}$$