Bayesian computation of Markov random fields and its application in medical imaging

Wanchuang Zhu

A thesis in fulfilment of the requirements the degree of
Doctor of Philosophy

UNSW
AUSTRALIA

School of Mathematics and Statistics
Faculty of Science, University of New South of Wales, Australia

February 2017

PLEASE TYPE

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: **Zhu**

First name: **Wanchuang** Other name/s:

Abbreviation for degree as given in the University calendar: **PhD**

School: **School of Mathematics and Statistics** Faculty: **Faculty of Science**

Title: Bayesian computation of Markov random fields and its application in medical imaging

Abstract 350 words maximum: (PLEASE TYPE)

PET imaging has been an active area of research over the recent years. In order to obtain uncertainty estimations of parameters and incorporate spatial dependence of the voxels in the PET images, a Bayesian spatial mixture model was employed to estimate kinetic parameters in compartmental modelling of the myocardium. Potts models were utilized to model the spatial dependence of the voxels. Motivated by the need to develop computationally efficient and accurate inferential methods for the parameter in the Potts model, particularly for large Potts models, a novel method was proposed to overcome the well known normalizing constant problem in the Potts models. The suggested method took advantage of conditional independence of the Markov random field (MRF) and the original lattice of Potts model was recursively split into a series of sublattices. Consequently, the original density function can be calculated by multiplying all the conditional density functions of the sublattices. The procedure avoids the calculation of the normalizing constant entirely. In this sense, normalizing constant problem was overcome.

An alternative method was proposed to overcome the normalizing constant problem. In the suggested method, the intractable density function was decomposed into a series of conditional density functions. Subject to some assumptions, each conditional density can be approximated by a Monte Carlo approximation of conditional distribution of the corresponding summary statistics. The method has been demonstrated to be faster than most of the other methods in empirical studies. In addition, this method is extendable to irregular lattices.

Label switching in MCMC is a well known problem in mixture models. Many methods have been proposed to solve the problem. However, the algorithms were difficult to scale up with the size of observations very well. A new method that can scale up well with the size of spatial field was suggested to solve the label switching problem. The newly developed method achieved the optimized permutation by minimizing a loss function.

FOR OFFICE USE ONLY Date of completion of requirements for Award:

THIS SHEET IS TO BE GLUED TO THE INSIDE FRONT COVER OF THE THESIS

ORIGINALITY STATEMENT

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Signed ……..

Date ……..

Acknowledgement

I would like to express my sincere gratitude to my supervisor Yanan Fan for her continuous support of my PhD study, for her patience and immense knowledge. Her guidance helped me to keep on the right track of my study in all the time of research and writing of this thesis.

My sincere thanks also goes to my family. Words cannot express how grateful I am to my father, my mother and my younger brother for all of the sacrifices that you have made on my behalf. I am so grateful for your unconditional support of my study. I would also like to thank all of my friends who supported me in writing and incented me to strive towards my goal. All your support and encouragement were greatly appreciated.

At the end, I would like to express appreciation to my beloved wife Julia Feng who spent numerous hard times with me and has been always my support in the moments when the pressure of research was unbearable. Your sincere prayers and your sacrifices for our family were much appreciated.

Abstract

PET imaging has been an active area of research over the recent years and has been used to facilitate disease diagnosis, for instance, in cancer/tummor detection. Given the fact that PET images are extremely noisy, researchers have encountered difficulties in the analyses of PET images. Mixture models have been widely utilized in PET images analyses due to their flexibility and capabilities of modelling heterogeneous data. When spatial dependence has to be considered in the modelling, challenges arise in the subsequent parameter estimations. In particular, the large sizes of the images often lead to computational intractability. This thesis has been largely focused on the inferential problems resulting from the intractable normalizing constants in the spatial mixture models involving the Potts/Ising models.

In Chapter 2, a Bayesian spatial mixture model was employed to estimate kinetic parameters in compartmental model of the myocardium. Our results suggested that Bayesian inference can provide more robust estimations than the conventional methods. In addition, Bayesian inference naturally provided uncertainty estimations for the parameters. The uncertainty estimations are particularly important due to the extremely noisy nature of the data. The spatial dependence between voxels was incorporated by employing the Potts model as the prior in the spatial mixture model where Thermodynamic integration (Green and Richardson (2002)) was utilized to solve the inferential problems related to the spatial correlation.

Motivated by the need to develop computationally efficient and accurate infer-

ential methods for the spatial mixture models, in Chapter 3, a novel method was proposed to overcome intractable normalizing constant problem in the Potts model. The proposed method took advantage of conditional independence of the Markov random field (MRF) and the original lattice of Potts model was recursively split into sublattices. Two sublattices were generated at each split. The first sublattice consisted of pixels which were mutually independent given the second sublattice, and vice versa. Therefore, it became tractable to calculate the conditional density function of the first sublattice given the second one according to the property of the MRF. The second sublattice was then approximated by a new Potts model. The second sublattice was split again and two new smaller sublattices were generated. The decomposition procedure was repeated until some preset criterion was satisfied. The original lattice of Potts model was eventually decomposed into many sublattices of different sizes. The original density function can be calculated by multiplying all the conditional density functions of the sublattices. The procedure avoids the calculation of the normalizing constant entirely. It has been shown that the new method is able to deal with Potts models of large dimensions which cause problems in many existing methods. The ability of dealing with large lattices becomes more and more useful as the size of available data nowadays has increased exponentially. The algorithms which can handle large dataset are needed more urgently.

In Chapter 4, an alternative method was proposed to overcome the normalizing constant problem. In the suggested method, the intractable density function was decomposed into a series of conditional density functions. Subject to some assumptions, each conditional density can be approximated by a Monte Carlo approximation of conditional distribution of the corresponding summary statistics. The method has been demonstrated to be faster than most of the competitors in the empirical studies. In addition, this method is extendable to irregular lattices.

Finally, when a mixture model is used in conjunction with Markov random field,

label switching arises since posterior distributions are invariant with respect to the permutation of MCMC samples. Various methods were developed to solve the label switching problem. However, it was difficult to find an algorithm which is suitable for the spatial mixture models involving large sized Potts models. In Chapter 5, a new method was suggested to solve the label switching problem for the spatial mixture models. We concluded with some dicussions in Chapter 6.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Introduction

Positron emission tomography (PET) is a powerful medical imaging modality which exploits point sources of radioactivity to produce voxel-wise images. This medical imaging technique is widely used to study biological processes in-vivo (Boudraa et al. (1996), Liew et al. (2000), Jiang (2004), Martinez-Möller et al. (2009), Belhassen and Zaidi (2010)). A radioactive tracer is administered into blood stream of an object, usually a human body, and is delivered to the whole body by the flowing blood. Radiation is created when the nuclei of the tracer decay and produce photons which are then captured by radiation sensitive detectors external to the body.

A voxel is an equal sized and non-overlapping volume in the imaged volume. The goal of PET imaging is to obtain a map of radioactivities with respect to the location of voxels. The resulting map shows the tissues in which the molecular tracer has become concentrated and it can be interpreted by a radiologist in the context of the patient's diagnosis and treatment plan.

Dynamic PET imaging, i.e, PET images taken over time, collects a series of frames of sinogram data over contiguous time intervals. This enables dynamic PET images

to measure changes of radioactive concentrations in a quantifiable way over time. This, in turn, offers very useful information about the underlying physiological or metabolic processes and thus makes dynamic PET images helpful in diagnosis of certain cancers (Toga and Mazziotta (2002, Chapter 18)).

The quantitative accuracy of PET measurements is limited by its weak capability to resolve small objects, leading to poor resolution of PET images. Thus, inaccurate estimations of radioactivity concentration and related metabolic mechanism are quantified. These biased estimations also result from blurring of counts out of and into the structure from surrounding radioactivity, which are referred to "spill-out" and "spill-in" respectively. Collectively, they are referred to as partial-volume effects.

Millions of voxels are generated as a 3D volume of data in a typical PET image. These 3D volumes are tomographic reconstructions, see Leahy and Qi (2000). The partial-volume effects, combined with errors generated during image reconstruction (Alessio and Kinahan (2006)), result in noisy PET images. To reduce the effects of noise and recover the true radioactivity concentration, attention has been directed towards the development of algorithms to improve PET image reconstruction and the subsequent parameter estimation.

Image reconstruction algorithms are not main focus of our work. Parameter estimations of given reconstructed images are discussed in this work. Parameter estimation procedures involve fitting appropriate models for time activity curves (TAC) which are the time series collected at each voxel. Gunn et al. (1997), Zhou et al. (2013), Mohy-ud Din et al. (2014) consisted of a small proportions of the whole literature. The current methods for parameter estimation either utilize a voxel-wise fashion (Gunn et al. (1997)) or incorporate spatial dependence by adding a penalty term controlled by a control parameter in likelihood function (Mohy-ud Din et al. (2014)). However, the estimation of control parameter for spatial dependence was either ignored or remains challenging.

The work described herein will focus on the development of Bayesian inferential methods for spatial mixture models motivated by the estimation of the above control parameters for noisy PET images. In particular, we study in detail the estimation problems related to the spatial mixture models involving Potts models for extremely large datasets. In the rest of the Chapter, spatial mixture models and Potts models will be introduced, as well as the existing solutions for related inferential difficulties.

1.2 Spatial mixture model

1.2.1 Mixture models

Mixture models can be dated back to Pearson (1894) that is deemed as the first paper to advocate statistical method in biological studies, according to Stigler (1986). Various applications were outlined in Titterington et al. (1985) and Titterington (1997), such as agriculture, fishery, medicine,economics and so forth . In recent times, applications have been extended to a wide range of areas, including micro-array analysis (McLachlan et al. 2002), disease mapping (Green and Richardson 2002), finance analysis (Brigo and Mercurio 2002; Alexander 2004; Xu and Knight 2013), texture models (Permuter et al. 2003; Sujaritha and Annadurai 2011), ecology (Ullah et al. 2015), image analysis (Brazey and Portier 2014), density estimation (Zhu 2016) and so on. Mixture models have been extensively utilized in cluster analysis since it was first proposed, especially for heterogeneous data. A systematical review on mixture model as a tool for clustering analysis can be found in McLachlan and Basford (1988).

Let $\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_n$ denote observations of size n, where each $\mathbf{Y}_i$ denotes a p-dimensional random vector with probability density function $f(\mathbf{Y}_i, \boldsymbol{\theta})$ which denotes $f(\mathbf{Y}_i, \boldsymbol{\theta})$ the parametric density function of $\mathbf{Y}_i$ given unknown parameters $\boldsymbol{\theta}$. Suppose there are G components in a mixture model. Each observation is assumed to be drawn from the mixture of G Gaussian distributions. Then the likelihood func-

tion of each observation can be written as,

$$f(\mathbf{Y}_i) = \sum_{j=1}^{G} \pi_j f_j(\mathbf{Y}_i, \boldsymbol{\theta}_j), \tag{1.1}$$

where $\pi_j, j = 1, 2, \cdots, G$ are nonnegative values denoting mixing proportions or weights. The weights should sum to one, that is, $\sum_{j=1}^{G} \pi_j = 1$. For example, if each density function is distributed as Gaussian distribution, the mixture model is known as a Gaussian mixture model. Then $\boldsymbol{\theta}_j$ in Equation 1.1 denotes the parameters in the Gaussian distribution, which are the mean and variance parameters.

Parameter estimations are obtained for G groups in the mixture model as a tool of clustering analysis. Thus, the probabilities of each observation belonging to each group can be calculated accordingly.

1.2.2 Spatial mixture model

Spatial mixture models extend the usual mixture models by incorporating spatial dependence between observations. Spatial dependence can be found in many applied sciences, such as epidemiology (Lawson and Clark 2002), medicine (Hartvig and Jensen 2000; Woolrich et al. 2005; Woolrich and Behrens 2006), genetics (Guillot et al. 2005), ecology (Royle 2004; Lichstein et al. 2002) and many others (KaewTraKulPong and Bowden 2002; Weiss and Adelson 1996; Huang et al. 2005). To incorporate spatial dependence, a spatial penalty term can be added to the likelihood function. The penalty term can take different forms. In Lichstein et al. (2002), the conditional autoregressive (CAR) model was adopted to fit spatial correlation between individuals where the penalty term takes continuous form. Woolrich et al. (2005) utilized a discrete form of penalty term which was also employed by Geman and Geman (1984), Besag (1986). Control parameters were used to control the strength of spatial dependence. As demonstrated in Woolrich et al. (2005), inference of control parameters

was problematic, leading to some studies using fixed values for control parameters. These methods shared a distinct disadvantage leading to inflexibility of control parameters.

The Potts model is one of the most ubiquitous models which were utilized to incorporate spatial dependence, in particular in image analysis. Potts models were usually utilized as prior distributions in the Bayesian framework. We refer to mixture models with a Potts model prior as spatial mixture model in this thesis.

A latent variable z_i is introduced for each observed data $\mathbf{Y}_i, i = 1, \ldots, n$, where each pair $(\mathbf{Y}_i, z_i)$ has a corresponding spatial location. For instance, the posterior distribution of a q-component spatial mixture model takes the form

$$\pi(\mathbf{z}, \beta, \boldsymbol{\theta}|\mathbf{Y}) \propto \prod_{i=1}^{n} \pi(\mathbf{Y}_i|\theta, z_i)\pi(\mathbf{z}|\beta)\pi(\beta)\pi(\boldsymbol{\theta}), \tag{1.2}$$

where $\pi(\mathbf{Y}_i|\boldsymbol{\theta}, z_i)$ denotes the component distribution for $\mathbf{Y}_i$ conditional on the model parameters $\boldsymbol{\theta}$ and z_i. $\pi(\boldsymbol{\theta})$ and $\pi(\beta)$ denote the prior and hyper prior for the unknown parameters accordingly. $\pi(\mathbf{z}|\beta)$ denotes the density function of the Potts model.

1.3 Potts model

A Potts model, consisting of n discrete random variables $\mathbf{z} = (z_1, \ldots, z_n)$, can be defined on a rectangular lattice $\mathcal{L}$. The sample space for each z_i is $\{1, 2, \cdots, q\}$, and the corresponding model is referred to as the q-state Potts model. When $q = 2$, the Potts model is known as the Ising model. If $\mathbf{z}$ is distributed as Potts model, the corresponding density function is given as,

$$\pi(\mathbf{z}|\beta) = \frac{1}{\mathcal{C}(\beta)} \exp\{\beta \sum_{i \sim j} I(z_i = z_j)\}, \tag{1.3}$$

where $i \sim j$ indicates that i and j are neighbours, and $\mathcal{C}(\beta) = \sum_{\mathbf{z}} \exp\{\beta \sum_{i \sim j} I(z_i = z_j)\}$ is the normalizing constant. $I(\cdot)$ is the indicator function, $I(z_i = z_j) = 1$ if $z_i = z_j$ is true, otherwise $I(z_i = z_j) = 0$.

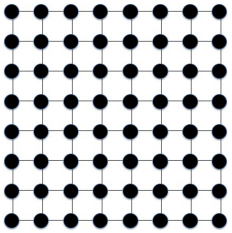


Figure 1.1: The Potts model on a 8×8 lattice.

Figure 1.1 gives a pictorial illustration of Potts model on a 2D lattice. The first order neighbourhood structure defines the nearest four pixels as neighbours of each pixel. The structure in 3D MRF is similarly defined with each site dependent on its neighbours on the left, right, front, back, above and below. The parameter β in Equation 1.3 controls the degree of spatial dependence. See Wu (1982), Chang and Shrock (2015) for more illustrations on the Potts model.

1.3.1 Generation of Potts model

The Markov chain Monte Carlo (MCMC) can be employed to sample from a Potts model. We describe three methods to generate a Potts model: The Swendsen-Wang Algorithm (Swendsen and Wang (1987)), Wolff's Algorithm (Wolff (1989)) and Gibbs sampling using conditional independence (Feng (2008)).

Swendsen-Wang algorithm

For the Swendsen-Wang algorithm, an auxiliary random variable u is proposed to assist the generation of a Potts model. This auxiliary variable will be repeated for a number of times in each update of the configuration of the Potts model.

The generation of Potts model begins with an arbitrary configuration of a Potts model $\mathbf{z_0}$. $\mathbf{z_0}$ is updated from iteration to iteration until the Potts model converges. Suppose the current iteration is t. Given $\mathbf{z_t}$, an assistant network is required to be constructed. Each pair of neighbours in the Potts model $\mathbf{z_t}$, such as z_i and z_j, is considered as an interaction, which is denoted by $i \sim j$. If current $z_i \neq z_j$, no bond is created between them. If current $z_i = z_j$, then a bond between z_i and z_j is created with a probability of $1-\exp(-\beta I(z_i = z_j))$. This is achieved by simulating a binomial random variable u with success probability equal to $1 - \exp(-\beta I(z_i = z_j))$. If $u = 1$, the bond is created. Otherwise, no bond will be created. According to the bonds created in above sweeps, the current Potts model $\mathbf{z_t}$ can be divided into several patches (clusters) with all the sites in each patch holding the same value. The next step is to assign a random value from the sample space to each patch where all the sites will be assigned to the same value. A new configuration of the Potts model is generated and is denoted as $\mathbf{z_{t+1}}$. At the same time, one iteration is completed. Long enough iterations will be implemented to generate one sample of Potts model of interest. A summary of the procedures above is illustrated in Algorithm 1, where ii denotes the number of iterations.

Algorithm 1: Swendsen-Wang algorithm

Input: Current $\mathbf{z}$

for $ii = 1, \cdots, n$ **do**
- Create bond between z_i and z_j with probability of $1 - \exp(-\beta I(z_i = z_j))$.
- Divide Potts model into patches according to the bonds.
- Change each patch to a random value including the current value.

end

Wolff's algorithm

A modification was suggested by Wolff (1989) to improve upon the Swendsen-Wang algorithm. The difference between these two algorithms lies in how the patches are

updated. As previously described, the Swendsen-Wang updates each patch to a random value from the sample space. Consequently, there is a chance that the value remains unchanged after one iteration. In contrast, the Wolff's algorithm enforces that each patch has to be updated to a different value. The details are shown in Algorithm 2. A summary of the procedures above is illustrated in Algorithm 2, where ii denotes the number of iterations.

Algorithm 2: Wolff's algorithm

Input: Current **z**

for $ii = 1, \cdots, n$ **do**
  Randomly choose a site z_i. Create bond between z_i and its neighbours with probability of $1 - \exp(-\beta I(z_i = z_j))$, where $i \sim j$.
  Keep creating bonds until no more site can be linked together.
  Change the patch to another value.
end

From empirical studies, $n = 50$ is long enough. More details can be found in Liang and Jin (2013). Statistical efficiency was compared between the Wolff's algorithm and the Swendsen-Wang algorithm in Feng (2008) where it was concluded that Wolff's algorithm is more preferable in most of cases in terms of computational efficiency.

Gibbs sampling using conditional independence

Prior to the introduction of Gibbs sampling using conditional independence, single site updating scheme is needed to be reviewed. Literally, single site updating scheme updates a single site in the configuration at each iteration. It is an example of Gibbs samplers where the full conditional distributions of each parameter are required. In the scenario of Potts model, the full conditional distribution of each site z_i is known as multinomial distribution. Mathematically, it is given as following:

$$\pi(z_i|z_{\partial i}) = \frac{\exp\{\beta \sum_{i\sim j} I(z_i = z_j)\}}{\sum_{k=1}^{q} \exp\{\beta \sum_{i\sim j} I(k = z_j)\}}, \tag{1.4}$$

where ∂i denotes all the neighbours of z_i. The full conditional distribution of z_i is simple in terms of computation. Therefore, single site updating scheme is easy to implement but may mix slowly as a result of numerous sites. The Gibbs sampling introduced herein is essentially adopting the same full conditional distribution but in a parallel way.

Given the neighbourhood structure of Potts model, the lattice can always be divided into non-overlapping sublattices. The sublattices own special properties: the sites in any sublattice are mutually independent given the other sublattices. The "coding method" approach were employed to obtain the sublattices (see Besag (1974), Winkler (2003) and Wilkinson (2005)). The minimum number of sublattices for the first order structure is 2 in both 2D and 3D lattices. Subsequently, 4, 4 and 8 are the minimum number of sublattices in the second order neighbourhoods structure with 8 neighbours in 2D, 18 neighbours in 3D and 26 neighbours in 3D respectively. These numbers are the so-called "chromatic number", whose more details can be found in Feng (2008) and Feng et al. (2012).

The first order neighbourhood structure is demonstrated in Figure 1.2. The first order neighbourhood structure means that given other pixels, each pixel is only dependent on its four nearest pixels. In this case, chromatic number is equal to 2, which corresponds to the gray pixels and the black pixels. It is straightforward to conclude that given the gray pixels, the black pixels are mutually independent, and vice versa.

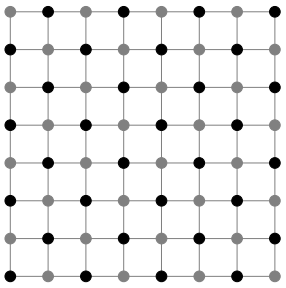


Figure 1.2: Left panel: a first order neighbourhood MRF, with black and grey points depicting $\mathbf{z}$. Each site only depends on the nearest four neighbours of the other color.

Once such sublattices are found, the updating can be implemented in block-wise level instead of the site-wise level. To a considerable extent, this improves the efficiency of single site updating scheme. The details of this generation method are shown in Algorithm 3.

Algorithm 3: Gibbs sampling using conditional independence

Input: Current **z**, chromatic number C

for $ii = 1, \cdots, n$ **do**
  for $j = 1, \cdots, C$ **do**
    Choose sublattice j.
    Given other sublattices, simultaneously update all the sites in the sublattice j according to the distribution in Equation 1.4.
  end
end

1.4 Normalizing constant problem in Potts models

1.4.1 Bayesian inference of Potts model

Bayesian inference (O'Hagan and Forster (2004)) treats parameters in likelihood function as random variables. Inference is based on posterior distributions of parameters. Posterior distribution can be derived from Bayes theorem which is given below:

$$\pi(\theta|x) = \frac{\pi(x|\theta)\pi(\theta)}{\int_\theta \pi(x|\theta)\pi(\theta)d\theta},$$

where, in general, θ is a unknown parameter, x denotes observed data. When it comes to inference about Potts model, β is always the parameter of interest. By applying Bayes theorem to Potts model, the posterior distribution of β is given below:

$$\pi(\beta|\mathbf{z}) \propto \pi(\mathbf{z}|\beta)\pi(\beta), \tag{1.5}$$

where $\pi(\beta)$ is prior distribution of β.

1.4.2 Computation of normalizing constant

The analytic form of $\mathcal{C}(\beta)$ is given as,

$$\mathcal{C}(\beta) = \sum_{\mathbf{z}} \exp\{\beta \sum_{i \sim j} I(z_i = z_j)\}. \tag{1.6}$$

Literally, $\mathcal{C}(\beta)$ is the summation over all possible realizations of $\mathbf{z}$. For a q-state Potts model with size of $n \times n$, the number of possible realizations is q^{n^2}. Even in a moderate sized Potts model, the number tends to be too large to compute. Since an intractable term $\mathcal{C}(\beta)$ is in $\pi(\mathbf{z}|\beta)$, $\pi(\beta|\mathbf{z})$ becomes intractable. The intractability of normalizing constant causes problem of inference about β. This is referred to normalizing constant problem that is illustrated by MCMC algorithm below.

Algorithm 4: Metropolis-Hastings algorithm for Potts model

Input: Current β
for $i = 1, \cdots, n$ **do**
  Propose new state $\beta^{'} \sim \pi(\beta^{'}|\beta)$
  Compute acceptance ratio:
    $a = \frac{\pi(\beta^{'})\pi(\beta|\beta^{'})\pi(\mathbf{z}|\beta^{'})}{\pi(\beta)\pi(\beta^{'}|\beta)\pi(\mathbf{z}|\beta)}$
  Draw random number $u \sim U[0, 1]$
  If $(a > u)$ set the new state to be $\beta^{'}$, otherwise keep β
end

Algorithm 4 shows how MCMC algorithm is used to sample β from the posterior distribution. In computation of a, $\frac{\pi(\mathbf{z}|\beta^{'})}{\pi(\mathbf{z}|\beta)} = \frac{\exp\{\beta^{'} \sum_{i \sim j} I(z_i = z_j)\}\mathcal{C}(\beta)}{\exp\{\beta \sum_{i \sim j} I(z_i = z_j)\}\mathcal{C}(\beta^{'})}$ is intractable, since both $\mathcal{C}(\beta)$ and $\mathcal{C}(\beta^{'})$ are unknown.

For relatively small random fields (less than 10×10), the normalizing constant $\mathcal{C}(\beta)$ can be computed by summing exhaustively over all possible realizations of $\mathbf{z}$ for any given value of β. However, the calculation of $\mathcal{C}(\beta)$ becomes computationally intractable for large spatial fields. This problem is well known in the statistical community, and has received considerable amount of attention in the literature, see Lyne et al. (2015) for a recent review.

1.5 Existing methods for normalizing constant problem

Statisticians have developed various methods to solve the problem of normalizing constant over the past few decades. Gelman and Meng (1998) reported that numerical integration (Evans and Swartz (1995)), analytic approximation (DiCiccio et al. (1997)) and Monte Carlo simulation were common approaches to tackle the normalizing constant problem. Gelman and Meng (1998) developed path sampling method which is one of Monte Carlo methods to reduce Monte Carlo error comparing with previous methods. A multitude of approaches have been proposed since then. Four categories were reviewed in this thesis: Monte Carlo methods, numerical integration methods, approximation methods and exact sampling methods.

1.5.1 Monte Carlo methods

In a nutshell, Monte Carlo methods aim to calculate the normalizing constant directly by using Monte Carlo simulation. Monte Carlo methods were widely used because of their flexibility and applicability to high-dimensional problems. Path sampling is one of Monte Carlo methods. It was developed on the foundation of thermodynamic integration which is well known in Physics. Gelman and Meng (1998) reported that thermodynamic integration has close connection with importance sampling. In addition, path sampling offered more flexibility and thus potential efficiency to thermodynamic integration. Path sampling was further discussed in Green and Richardson (2002) where a concise way of implementation was demonstrated. Other simulation-based methods can be found in Geyer and Thompson (1992), Gu and Zhu (2001), Liang (2007) and references therein.

Thermodynamic integration

Thermodynamic integration was derived in Gelman and Meng (1998) by using path sampling to approximate the log ratio of normalizing constants. In the original paper, the log ratio of normalizing constants is of interest. The identity is shown below:

$$\log\left\{\frac{\mathcal{C}(\beta^{'})}{\mathcal{C}(\beta^{*})}\right\} = \int_{\beta^{*}}^{\beta^{'}} E(U(\mathbf{z}))d\beta.$$

Thermodynamic integration in Green and Richardson (2002) is one of the further development of the identity. The identity can be derived by the following procedures. The Potts model can be rewritten in the following form:

$$\pi(\mathbf{z}|\beta) = \exp\{\beta U(\mathbf{z}) - \theta_q(\beta)\},$$

where $U(\mathbf{z}) = \sum_{i\sim i'} I(z_i = z_{i'})$ and $\theta_q(\beta) = \log(\sum_{\mathbf{z}\in Z} e^{\beta U(\mathbf{z})})$, q is the number of states in the Potts model. We take differentiation of $\theta_q(\beta)$ with respect to β, the following was obtained:

$$\frac{\partial}{\partial\beta}\theta_q(\beta) = \frac{\partial}{\partial\beta}\log\sum_{\mathbf{z}\in Z} e^{\beta U(\mathbf{z})} = \sum_{\mathbf{z}\in Z} U(\mathbf{z})p(\mathbf{z}|\beta) = E(U(\mathbf{z})|\beta, q).$$

If $\beta = 0$, then $\theta_q(\beta) = \theta_q(0) = \log\sum_{\mathbf{z}\in Z} 1 = n\log q$. Therefore,

$$\theta_q(\beta) = n\log q + \int_0^{\beta} E(U|\beta^{'}, q)d\beta^{'}. \tag{1.7}$$

$E(U|\beta^{'}, q)$ could be obtained by Monte Carlo simulation given any value of $\beta^{'}$. The simulations can be implemented over a grid of β, resulting in a look-up table for further calculation. Then $\int_0^{\beta} E(U|\beta^{'}, q)d\beta^{'}$ could be estimated by a spline approximation (Green and Richardson (2002)). Consequently, the normalizing constant $\mathcal{C}(\beta)$ can be calculated by $\exp\{\hat{\theta}_q(\beta)\}$.

1.5.2 Numerical integration

General factorization model

Let $p(\mathbf{z}|\beta)$ denote the unnormalized likelihood function of Potts model $\mathbf{z}$. Reeves and Pettitt (2004) proposed a lag-r model to factorize the $p(\mathbf{z}|\beta)$. A valid factorization of $p(\mathbf{z}|\beta)$ was in the following form,

$$p(\mathbf{z}|\beta) = p_1(z_1, z_2, \ldots, z_{r+1})p_2(z_2, z_3, \ldots, z_{r+2}) \ldots p_k(z_k, z_{k+1}, \ldots, z_n). \tag{1.8}$$

Therefore, the normalizing constant $\mathcal{C}(\beta)$ is given by

$$\mathcal{C}(\beta) = \sum_{\mathbf{z}} p(\mathbf{z}|\beta) = \sum_{z_{k+1}^n} \sum_{z_k} p_k(z_k^n) \sum_{z_{k-1}} p_{k-1}(z_{k-1}^{n-1}) \ldots \sum_{z_1} p_1(z_1^{r+1}). \tag{1.9}$$

By rearranging the order of z_i, lag-r model can save computational time when calculating the normalizing constant. The total computational complexity is reduced from $O(q^n)$ to $O(q^{r+1})$, where q is the number of states in Potts model and n is the length of the Potts model. However, their method was limited by the size of lattice. They recommended that the number of rows should be no more than 20. The above requirement limits the application of lag-r model. Moreover, the computational time increases exponentially as q increases. Thus, the limitation restricted its applications to large random fields. A similar idea can be found in Bartolucci and Besag (2002) where conditional probabilities instead of joint probabilities were defined. Nonetheless, the full conditional distributions are not always compatible with a valid joint distribution.

1.5.3 Approximation methods

Pseudo likelihood

Besag (1975) first developed a pseudo likelihood (PL) to approximate $\pi(\mathbf{z}|\beta)$ directly. He used the following formula to approximate the normalized likelihood function,

$$\pi(\mathbf{z}|\beta) = \prod_{i=1}^{n} \pi(z_i|z_{\setminus i}, \beta), \tag{1.10}$$

where $z_{\setminus i}$ denotes all the sites in $\mathbf{z}$ except z_i. In a Markov random field, one site is only dependent on its neighbours given all other sites in the field. Therefore, $\pi(z_i|z_{\setminus i}, \beta)$ can be reduced to $\pi(z_i|z_{\partial i}, \beta)$, where $z_{\partial i}$ denotes the neighbourhoods of z_i. As a result, the approximation becomes

$$\pi(\mathbf{z}|\beta) \approx \prod_{i=1}^{n} \pi(z_i|z_{\partial i}, \beta). \tag{1.11}$$

In fact, the right side of Equation (1.11) is one type of composite likelihoods that were introduced by Lindsay (1988), and were studied in a remarkable volume of papers. As reported in Varin et al. (2011), the estimator of composite likelihood was asymptotically unbiased. However, the variance was underestimated. The approximation takes advantage of local dependence in the lattice and approximates the true likelihood function as the product of local sublattices. PL outperforms the other approaches in terms of computational time due to the simple form of the approximation. At the same time, Liang et al. (2016) reported that PL estimator is unsatisfactory when the dependence is strong in the Potts model.

Partially ordered Markov model

Cressie and Davidson (1998) suggested a similar method called Partially ordered Markov models (POMMs). Not only can POMMs generalize the Markov chain to

a directed acyclic graph (DAG), but they can also generalize Markov mesh models (MMMs), which were studied in Abend et al. (1965). In POMMs, the pixels interact with each other differently with in Potts models. As we can see, the interactions are two-ways in Potts model. Whereas, the interactions are directional. A set of parental pixels are requested to define for each z_i. The parents can effect z_i, while z_i has no impact on its parents. With POMMs, Equation (1.3) can be calculated without computing the normalizing constant. For eligible MRFs, density function is expressed as the following,

$$\pi(\mathbf{z}|\beta) = \prod_{i=1}^{n} \pi(z_i|pa(z_i), \beta),$$

where $pa(z_i)$ denotes parents point of z_i. However, POMMs cannot be applied to all MRFs, because only a subset of MRFs are expressible as POMMs. This, to a large extent, limits applications of POMMs.

Reduced dependence approximation

Friel et al. (2009) developed reduced dependence approximation (RDA) method given the initial research in Reeves and Pettitt (2004). They aimed to extend the method of Reeves and Pettitt (2004) to larger lattices. For large lattice, by relaxing the dependence in the latent model, they split the lattice into smaller sublattices. Then tractable approximations like pseudo likelihood and lag-r model can be applied to the small sublattices. By calculating the normalizing constants of the small lattices, the likelihood function can be given as the following,

$$\pi(\mathbf{z}|\beta) = \frac{p(\mathbf{z}|\beta)(z_{m_1 \times n}(\beta))^{m-m_1-1}}{z_{(m_1+1)\times n}(\beta))^{m-m_1}},$$

where m_1 is the number of rows in the small sublattices and $p(\mathbf{z}|\beta)$ is the unnormalized likelihood function of the Potts model.

RDA was proposed to solve the normalizing constant problem for large lattices.

It requires assumptions about the dependence structure. In the meantime, RDA aims to calculate normalizing constant itself and then substitute the value in Equation 1.3. This is tedious in terms of computation, see simulation study in Section 3.4.

1.5.4 Exact sampling methods

The purposes of the above methods are either to approximate the density function of the Potts model directly or to calculate the normalizing constant and thus make the density function tractable. Exact sampling methods aim to tackle the problem during MCMC sampling step.

Auxiliary variable methods

Given all these approximation methods, Møller et al. (2006) advocated a sampling scheme which aimed to obtain the full posterior distribution of β. Since drawing posterior samples of β is intractable directly from $\pi(\beta|\mathbf{z})$, they introduced an auxiliary variable v. If $\pi(v, \beta|\mathbf{z})$ is available, then $\pi(\beta|\mathbf{z})$ can be obtained by integrating out v from $\pi(v, \beta|\mathbf{z})$. They defined an auxiliary variable v on the same state space of $\mathbf{z}$ with the same distribution of $\mathbf{z}$. Suppose the current state of parameters is $\{v, \beta\}$. New state is proposed as the following. First of all, β^* is proposed given β with density function $\pi(\beta^*|\beta, \mathbf{z})$. Then v^* is proposed based on v, β^* and β with density function $\pi(v^*|v, \beta^*, \beta)$. As all the proposal densities are arbitrary, proposal density of v can be reduced to $\pi(v^*|\beta^*)$. In Metropolis-Hastings updating step of v and β, the acceptance ratio becomes,

$$a = \frac{\pi(v^*|\beta^*, \mathbf{z})\pi(\mathbf{z}|\beta^*)\pi(\beta^*)\pi(\beta|\beta^*)\pi(v|\beta)}{\pi(v|\beta, \mathbf{z})\pi(\mathbf{z}|\beta)\pi(\beta)\pi(\beta^*|\beta)\pi(v^*|\beta^*)}. \tag{1.12}$$

As a consequence, there is no normalizing constant in Equation 1.12. Nevertheless, appropriate auxiliary density $\pi(v|\beta, \mathbf{z})$ should be chosen properly. Otherwise, nor-

malizing constant will be introduced in the calculation again. If let $\pi(v|\beta, \mathbf{z}) = \pi(v|\beta)$, the normalizing constant problem occurs. One simple approximation is $\pi(v|\beta, \mathbf{z}) = \pi(v|\hat{\beta})$, where $\hat{\beta}$ is fixed. For example, set $\hat{\beta} = \hat{\beta}(\mathbf{z})$ which can be estimated from some quick algorithm, such as PL. Now the acceptance ratio can be reduced to:

$$a = \frac{U(v^*|\hat{\beta})U(\mathbf{z}|\beta^*)\pi(\beta^*)\pi(\beta|\beta^*)U(v|\beta)}{U(v|\hat{\beta})U(\mathbf{z}|\beta)\pi(\beta)\pi(\beta^*|\beta)U(v^*|\beta^*)}. \tag{1.13}$$

There is no normalizing constant in the above acceptance ratio, because the normalizing constants in the numerator and the denominator have cancelled. Sometimes the proposed method is referred to as single auxiliary variable method (SAVM). It is described in Algorithm 5. The name is adopted to differentiate from multiple auxiliary variable method (MAVM) that will be introduced in next Section.

Algorithm 5: Algorithm for auxiliary variable methods

Input: Current β, current v

for $i = 1, \cdots, n$ **do**

  Propose new state $\beta^* \sim \pi(\beta^*|\beta)$

  Generate an auxiliary variable $v^* \sim \frac{1}{\mathcal{C}(\beta^*)} \exp\{\beta^* \sum_{i\sim j} I(v_i^* = v_j^*)\}$

  Compute acceptance ratio:

    $a = \frac{U(v^*|\hat{\beta})U(\mathbf{z}|\beta^*)\pi(\beta^*)\pi(\beta|\beta^*)U(v|\beta)}{U(v|\hat{\beta})U(\mathbf{z}|\beta)\pi(\beta)\pi(\beta^*|\beta)U(v^*|\beta^*)}$

  Draw random number $u \sim U[0, 1]$

  If $(a > u)$ set the new state to be $\{\beta^*, v^*\}$, otherwise keep $\{\beta, v\}$

end

Comparing the acceptance ratios in Algorithm 4 with Equation 1.12, it is identified that $\mathcal{C}(\beta)$ was replaced by $U(v|\beta)/\pi(v|\beta, \mathbf{z})$ and $\mathcal{C}(\beta^*)$ was replaced by $U(v^*|\beta^*)/\pi(v^*|\beta^*, \mathbf{z})$.

The difficulty of SAVM lies in the generation of auxiliary variable given β. This requires the sample to be an exact sample that can be obtained by using exact sampling methods which can be coupling from the past (CFTP) Propp and Wilson (1996). However, the method is difficult to implement. In other words, exact sampling can be time consuming and make the overall computation much more complicated when using the auxiliary variable method.

Multiple auxiliary variable method

Murray (2007) advocated the multiple auxiliary variable method (MAVM) to improve the efficiency of Møller et al. (2006). In essence, SAVM was adopting an "importance sampling" estimator to approximate the ratio of normalizing constants. Usually, it suffers from high rejection rate. The high rejection rate will reduce the efficiency of sampling β. Consequently, computational time will increase. It was natural to extend importance sampling to other methods, such as annealed importance sampling (AIS Neal (2001)). Instead of generating one auxiliary variable, MAVM has to generate multiple auxiliary variables. Let $V = \{v_1, v_2, \cdots, v_{K+1}\}$ denote the set of auxiliary variables. The first auxiliary variable can be generated by using the same method in SAVM. The other auxiliary variables were defined by a sequence of Markov chain transition operators $\tilde{T}_k(v_{k+1}|v_k)$,

$$\pi(v_{k+1}|v_k, \beta, \mathbf{z}) \sim \tilde{T}_k(v_{k+1}|v_k, \beta, \hat{\beta}(\mathbf{z})), \quad k = 1, 2, ..., K. \tag{1.14}$$

All the bridging distributions were aimed at bringing $\pi(v_1|\beta, \mathbf{z})$ towards $\pi(v|\beta)$. $\tilde{T}_k$ was chosen to make the corresponding distribution p_k stationary. By default, p_k is in the following format:

$$p_k(v_k|\beta, \hat{\beta}(\mathbf{z})) \propto \pi(v_k|\hat{\beta})^{l_k} \pi(v_k|\beta)^{1-l_k}, \tag{1.15}$$

where $l_k = \frac{K-k+1}{K+1}$.

The details of MAVM are described in Algorithm 6. By incorporating AIS, MAVM has higher acceptance ratio. Although it is unnecessary for both SAVM and MAVM to calculate normalizing constant, they suffer from the need for perfect sampling of $\mathbf{z}$. Overall, perfect sampling takes excessive amount of time, consequently these two methods are not suitable for even moderate size of MRFs.

Algorithm 6: Multiple auxiliary variable algorithm

Input: Current β

for $i = 1, \cdots, n$ **do**

1. Propose new state $\beta^{'} \sim \pi(\beta^{'}|\beta)$
2. Generate an auxiliary variable $v_{K+1} \sim \pi(v_{K+1}|\beta^{'})$ using exact sampling method.
3. Propose $\{v_K, v_{K-1}, \cdots, v_1\}$ in order, transition operators are given as follow: $\pi(v_k|v_{k-1}, \beta^{'}, \mathbf{z}) \sim T_k(v_{k-1}|v_k, \beta^{'}, \hat{\beta}(\mathbf{z}))$, for $k = K, K-1, \cdots, 1$. Where T_k is the corresponding reverse transition operator $\tilde{T}_k$.
4. Compute acceptance ratio of the whole move from $\{\beta, V\}$ to $\{\beta^{'}, V^{'}\}$:

$$a = \frac{U(\mathbf{z}|\beta^*)\pi(\beta^*)\pi(\beta|\beta^*)}{U(\mathbf{z}|\beta)\pi(\beta)\pi(\beta^*|\beta)} \prod_{k=0}^{K} \frac{p_k(v_{k+1}^{'}|\beta^{'}, \hat{\beta}(\mathbf{z}))}{p_{k+1}(v_{k+1}^{'}|\beta^{'}, \hat{\beta}(\mathbf{z}))} \frac{p_{k+1}(v_{k+1}|\beta, \hat{\beta}(\mathbf{z}))}{p_k(v_{k+1}|\beta, \hat{\beta}(\mathbf{z}))}$$

5. Draw random number $u \sim U[0, 1]$

If $(a > u)$ set the new state to be $\beta^{'}$, otherwise keep β

end

Exchange algorithm

Murray (2007) suggested a simpler and more direct method to solve the normalizing constant problem. This method is usually referred to Exchange algorithm (EA) that is described in Algorithm 7. Like the above two methods, EA also has to draw

Algorithm 7: Exchange algorithm

Input: Current β

for $i = 1, \cdots, n$ **do**

Propose new state $\beta^{'} \sim \pi(\beta^{'}|\beta)$

Generate an auxiliary variable $v^* \sim \frac{1}{\mathcal{C}(\beta^{'})} \exp\{\beta^{'} \sum_{i \sim j} I(v_i^* = v_j^*)\}$

Compute acceptance ratio:

$$a = \frac{U(v^*|\beta)U(\mathbf{z}|\beta^*)\pi(\beta^*)\pi(\beta|\beta^*)}{U(v^*|\beta^*)U(z|\beta)\pi(\beta)\pi(\beta^*|\beta)}$$

Draw random number $u \sim U[0, 1]$

If $(a > u)$ set the new state to be $\beta^{'}$, otherwise keep β

end

exact sample from Potts model. Compared to Algorithm 5, EA adopts $\frac{U(v^*|\beta)}{U(v^*|\beta^*)}$ to approximate $\frac{\mathcal{C}(\beta)}{\mathcal{C}(\beta^*)}$, which is better than estimation in single auxiliary variable method. Similar to the extension from SAVM to MAVM, EA can be extended to exchange algorithm with bridging. More details can be found in Murray (2007). In recent times,

some variations of EA were proposed, see Liang (2010), Liang et al. (2016).

It is evident that current methods either requires heavy computation or compromises model flexibility by making assumptions on dependence structure. As the size of lattice increases, computational burden of Monte Carlo methods becomes more significant and their advantages tend to be less vital. Approximation methods require some assumptions about dependence structure of Potts model. The assumptions lead to the underestimation of variance of β in PL. Theoretically, exact sampling methods could be the best way to solve the normalizing constant problem, as it introduces auxiliary variable to cancel $\mathcal{C}(\beta)$ during sampling. The main drawback is that it requires exact sample from the Potts model.

Consequently, methods which satisfy the following requirements are essential to statisticians. Initially, it can scale up well with the size of Potts model. Subsequently, it should obtain good estimation of parameters of interest. In the thesis, two approaches were proposed to fulfill the above requirements wholly.

1.6 Thesis arrangement

Chapter 2. Bayesian inference of PET parametric image was implemented using spatial mixture model. The main body of this Chapter was published as "Zhu, W., J. Ouyang, Y. Rakvongthai, N. J. Guehl, D. W. Wooten, G. El Fakhri, M. D. Normandin, and Y. Fan. "A Bayesian spatial temporal mixtures approach to kinetic parametric images in dynamic positron emission tomography." Medical physics 43, no. 3 (2016): 1222-1234.".

Chapter 3. A recursive decomposition method was proposed to solve the normalizing constant problem in the Potts model. This Chapter is under revision for *Journal of Computational and Graphical Statistics* as "A novel approach for Markov Random Fields with intractable normalizing constant on large lat-

tices".

Chapter 4. An alternative approach to solve normalizing constant problem under perspective of summary statistics was proposed in this Chapter.

Chapter 5. Relabelling methods for mixture model were reviewed. A new approach for spatial mixture model was proposed. This Chapter was published as "Zhu, W., and Y. Fan. "Relabelling algorithms for mixture models with applications for large data sets." Journal of Statistical Computation and Simulation 86, no. 2 (2016): 394-413.".

Chapter 6. Summary for current work and directions for future work.

Chapter 2

A Bayesian spatial temporal mixtures approach to kinetic parametric images in dynamic positron emission tomography

2.1 Introduction to kinetic model estimation

Medical imaging plays an important role in medical diagnosis and treatment. Medical images can be generated from different modalities. Magnetic resonance imaging (MRI), computed tomography (CT), positron emission tomography (PET) are the most widely used image types in practice. Dynamic PET, i.e, PET images taken over time, can be utilized to measure tracer kinetics in-vivo, from which physiological parameters, such as tissue perfusion, ligand receptor binding potential, and metabolic rate can be determined using compartmental modelling techniques. Kinetic modelling utilizes differential equations to model the behavior of radioactive tracer which are injected into a patient's blood stream. Kinetic parameters are of

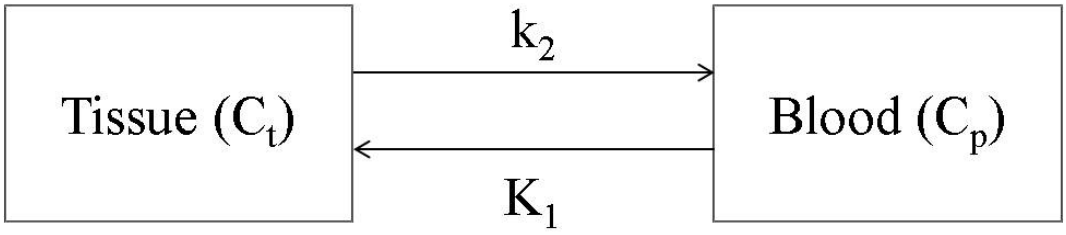


Figure 2.1: One-compartmental model with one tissue C_t and blood C_p.

interest and describe the behavior of tracer in specific areas.

Compartmental models are adopted to describe PET kinetic parameters. Each compartment can be viewed as a homogeneous region in the body or a tissue. According to the number of compartments, there are one-compartmental model and multi-compartmental model. The pictorial description of tracer behaviour for one-compartmental model is demonstrated in Figure 2.1. C_p denotes the concentration of radioactive tracer in blood and C_t denotes the concentration of radioactive tracer in specific tissue of interest.

Tracer transits between compartments and the paths can be described by a set of ordinary differential equations (ODE). The ODE for a one-compartmental model is given below,

$$\frac{dC_t}{dt} = K_1 C_p - k_2 C_t, \tag{2.1}$$

where K_1 denotes the velocity of tracer spreading from blood to the tissue. k_2 denote the velocity of tracer spreading in the reverse direction. C_ts which are collected at different time points form a TAC. The solution of Equation 2.1 is presented in Equation 2.10.

Estimation of the kinetic parametric images can be extremely challenging, since the data are often very noisy. Most conventional methods (Lammertsma and Hume (1996), Slifstein et al. (2008), Nye et al. (2008)) either defined a region of interest (ROI) and then estimated parameters based on the averages, or estimated parameters in a voxel-wise fashion. The former requires the identification of ROI which itself can be

difficult. The latter fails to utilize information from nearby voxels, resulting in more noisy estimates.

Gunn et al. (1997) employed a minimum least-squares approach to estimate the parameters for each voxel independently. To account for irregularities in the noise distribution, mixture models were utilized to fit each voxel in Lin et al. (2014). It was suggested that, it is necessary to restrict the total number of mixture components to be small and employ regularization to constrain parameter estimates. Zhou et al. (2013) reported Bayesian methods provided an alternative way of obtaining uncertainty estimates of the kinetic parameters, as well as model choice for the competing compartmental models. However, these methods yielded higher voxel to voxel variability because each voxel was processed independently. In addition, the assumption of Gaussian distribution can be inappropriate, leading to biased parameter estimates.

Given low signal-to-noise ratio (SNR), particularly in voxel-wise estimations, some external constraints were often necessary to stabilize parameter estimation. Huang and Zhou (1998), Kamasak et al. (2005) suggested smoothness regularization to constrain the parameters from nearby spatial locations to be more similar. Similarly, Tikhonov regularization was used in O'Sullivan and Saha (1999) to directly enforce parameter values to be within a certain range. Thus, estimates obtained were less sensitive to noise.

Voxel-wise estimation does not take into account spatial dependence which should be naturally considered to reduce noise. Recently, simultaneous clustering and parameter estimation methods have been proposed in (Saad, Smith, Hamarneh, and Möller 2007) using a spatially regularized K-means algorithm. The algorithm iteratively estimated the kinetic parameters in a least-squares sense between each cluster update. It was demonstrated that incorporating the physiological model in the clustering procedure performed better than their counterparts in terms of clustering.

However, the method offered no guidance on the choice of cluster numbers, or how to select the spatial regularization parameter, while both can have great influence on the results. A similar algorithm was suggested by Mohy-ud Din et al. (2014), where the clustering and parameter estimation were performed simultaneously. However, spatial correlation was ignored.

As discussed above, PET image clustering played an important role in kinetic model estimation procedures. In the rest of the Chapter, image segmentation methods were reviewed and then some estimation methods were described.

2.2 Medical image segmentation

PET image clustering is also referred to as PET image segmentation. Image segmentation is a procedure to extract the region of interest (ROI) through an automatic or semi-automatic process (see, Pham et al. (2000), Norouzi et al. (2014)). PET Image segmentation encountered difficulties in practice caused by low SNR of images due to the limitation of PET scanners and noisy environment. Numerous methods were proposed to solve image segmentation under the imperfect situation. Four categories are discussed in this work: thresholding methods, classical methods, computation-intensive methods and the mixture model-based method.

2.2.1 Thresholding methods

Thresholding methods partition image by setting several thresholds as boundaries between groups. The segmentation is achieved by grouping voxels according to the thresholds. Various methods can be employed to determine thresholds, such as standard deviation, and mean.

Thresholding methods are simple and efficient when the intensity values of the image are well separated. However, it cannot handle very noisy images where in-

tensity values are mixed. Therefore, thresholding methods were usually used as exploratory tools for further analysis, see Gordon et al. (1996), Singleton and Pohost (1997). Some other thresholding methods, for example, local thresholding and Otsu's thresholding were discussed in Norouzi et al. (2014) and references therein.

2.2.2 Classical methods

Classical methods include clustering analysis, factor analysis and principal component analysis (PCA). Clustering analysis was applied to the image detection in O'Sullivan (1993), Ashburner et al. (1996) and Kimura et al. (1999). Wong et al. (2002) aimed to classify the tissue time activity curves (TACs) according to their shapes and magnitudes, while most clustering methods only considered one of them. They employed a weighted least-squares distance as their distance metric. Wernick (2003) utilized similarity as distance metric which eliminated the effect of the magnitude of the difference of the TACs. El Fakhri et al. (2005) employed factor analysis to classify the TACs. Generalized factor analysis of dynamic sequences was adopted in this paper. This method was an extension of the factor analysis of dynamic sequences whose major drawback is the uncertainty of the unique solution. L.Wahl (1999) applied PCA to dynamic Flurodeoxyglucose (18F) (FDG) PET images for segmentation. PCA formed clusters of voxels that had similar kinetic behaviour of FDG uptake and summarized them into a component.

2.2.3 Computation-intensive methods

Computation-intensive methods include Fuzzy C-means(FCM) and machine learning methods including K-means, support vector machines (SVM), convolutional neural network (CNN) and artificial neural network (ANN). Wong et al. (2002) used a K-means like algorithm to classify the voxels. Liptrot et al. (2004) introduced a hierarchical K-means clustering method. Janssen et al. (2009) extracted the slopes of

the TACs, then used the slopes to classify the voxels. Formisano et al. (2008) focused on two machine learning methods: SVM and relevance vector machines. Machine learning models are typical data-drive models which analyze data (system), in particular finding connections between variables in the data without explicit knowledge of the intrinsic behaviour of the system. K-means type clustering algorithms have no test to check if the result is the within cost minimal.

FCM, which is also called soft K-means algorithm, is one of most widely used methods. FCM was first proposed by Dunn (1973) and developed by Bezdek et al. (1987). Every individual of interest was given a membership function with respect to each cluster. The objective of the FCM algorithms is to minimize the predefined objective function. FCM has been successfully used in the image segmentation in various image types, for example Clark et al. (1994), Pham and Prince (1999), Liew and Yan (2003), Chuang et al. (2006), Chen et al. (2006), Wang et al. (2008). Some applications of FCM to PET image segmentation can be found in Zaidi et al. (2002), Belhassen and Zaidi (2010), Onoma et al. (2012). Variants of FCM were developed. For example, Hatt et al. (2009) suggested a fuzzy locally adaptive Bayesian segmentation.

Pham (2001) considered the spatial effect by incorporating a spatial penalty in the membership functions. Ahmed et al. (2002) took into account neighbourhood by adding a second regularization to likelihood function. Liew and Yan (2003) incorporated the spatial information of the voxels by accounting for the effect of the neighbourhoods. Chen and Zhang (2004) also investigated the spatial effect in image segmentation. It is noticeable that some of them used an adaptive algorithm to improve the conventional FCM. Boudraa et al. (1996), Liew et al. (2000) and Jiang (2004) were especially for the applications of PET image segmentation. Belhassen and Zaidi (2010) aimed to overcome the heterogeneity of the voxels by modifying the objective function. However, as demonstrated in Woolrich et al. (2005), inference

of spatial control parameter is always problematic.

2.2.4 Mixture model-based methods

Mixture model-based clustering has been adopted in many research fields (Banfield and Raftery (1993), Celeux and Govaert (1995), McLachlan and Peel (2004)). Medvedovic et al. (2004) employed Bayesian mixture models to cluster microarray data. The Bayesian mixture models were utilized to incorporate the experimental variability, thus the precision of the clustering analysis can be generally improved. Gaffney and Smyth (1999), Gaffney and Smyth (2003), James and Sugar (2003) extended these models to regression mixture models and random effect regression models. Coke and Tsao (2010) advocated data compression, namely dimension reduction, in clustering. They transferred the original data space into a basis space using spline basis functions or polynomial basis functions. Biernacki et al. (2000) proposed a method by using integrated completed likelihood to determine the number of clusters in mixture model. This method was reported to be more robust than BIC when assumptions of the mixture model were violated. Samé et al. (2011) improved the model-based clustering for time series by considering changes in regime. The other applications of Mixture model clustering include Liu and Rattray (2010), Jiechang Wen (2012), Pelosi et al. (2015).

Van Leemput et al. (1999), Ashburner and Friston (2005), Aristophanous et al. (2007) suggested Gaussian mixture models to implement segmentation for PET images. These papers considered the temporal dependence, whilst ignoring the spatial effect. In order to take spatial dependence into account, spatial mixture models were developed. Markov random field (MRF) is the most popular approach to model the spatial correlation between nearby voxels in image segmentation. MRF assumes that nearby voxels are more likely to belong to the same cluster. This assumption is reasonable in most cases. MRF can also be incorporated into K-means algorithms in a

Bayesian framework, see Rajapakse et al. (1997), Pappas (1992), Held et al. (1997). Mixture models with MRF in general are difficult to estimate when the field is large. Such difficulty has been reported by Woolrich et al. (2005) andLi (2012). Specifically, Potts models which are studied in this thesis are the special forms of MRF. The difficulty of mixture models involving Potts models is related to an intractable normalizing constant which is introduced in Section 1.4. Mixture model with Potts is referred to as spatial mixture model.

2.3 Methods for kinetic parameter estimation

Curve fitting and Spatial K-means are described. Then a Bayesian spatial mixture model was proposed to cluster voxels and to estimate parameters simultaneously.

2.3.1 Curve Fitting

Kinetic analysis is performed by curve-fitting the TAC in each voxel using a nonlinear least-squares fitting,

$$\boldsymbol{k}_i = \text{argmin} \sum_{t=1}^{T} w^t (\mathbf{Y}_i^t - x_i^t(\boldsymbol{k}_i))^2, \tag{2.2}$$

where $\mathbf{Y}_i^t$ is the reconstructed activity concentration for voxel i at time frame t divided by frame duration $\Delta\tau_t = \tau_{t,e} - \tau_{t,s}$, $x_i^t(\boldsymbol{k}_i) = \frac{1}{\Delta\tau_t} \int_{\tau_{t,s}}^{\tau_{t,e}} K_{1,i}[\hat{C}_p(s) \otimes \exp(-k_{2,i}s)]ds$, is the average concentration over time frame t using the current estimates of the kinetic parameters $\boldsymbol{k}_i = (K_{1,i}, k_{2,i})$ in voxel i, and measured blood input function $\hat{C}_p(t)$, w^t is the weighting factor which herein is chosen to be the squared frame duration divided by the total counts in that frame (Gunn, Lammertsma, Hume, and Cunningham 1997). This nonlinear least-squares problem can be solved using the Levenberg-Marquardt algorithm. (Wang and Qi 2009) We denote this standard curve-fitting (SCF) approach in this paper.

2.3.2 Spatial K-means

The spatial SKMS method performs spatial K-means clustering and parameter estimation iteratively (Saad, Smith, Hamarneh, and Möller 2007). The process is as follows. (1) Initialize the cluster means $\mu_g, g = 1, \dots G$ for a predetermined number of clusters G. (2) For each g, estimate kinetic parameters $\boldsymbol{k}_g = \text{argmin}\sum_{t=1}^{T}(\mu_g(t) - C_t(t, \boldsymbol{k}_g))^2$; subject to positivity constraints on $\boldsymbol{k}_g$. (3) For each voxel indexed by $i = 1, \dots, n$, reassign cluster membership by minimizing the objective function $\sum_{i=1}^{n}(\sum_{g=1}^{G}||\mathbf{Y}_i - C_t(t, \boldsymbol{k}_g)||^2) + \beta \sum_{r=1}^{R} I(\mathbf{Y}_i, \mathbf{Y}_r)$. $\mathbf{Y}_i$ is the TAC at voxel i. β determines the influence of the spatial regularizer. $I(\cdot)$ is the indicator function returning a one if $\mathbf{Y}_i$ and $\mathbf{Y}_r$ belong to the same cluster and zero otherwise, for all $\mathbf{Y}_r$ in the neighbourhood of $\mathbf{Y}_i$. (4) Based on the new clusters, calculate μ_g as the mean for each cluster. (5) Repeat above steps until there are no significant changes in $C_t(t, \boldsymbol{k}_g)$.

There are two main issues for SKMS. First, the authors offered no theoretical guarantee of convergence of their proposed algorithm. Second, both the number of clusters and spatial regularization parameter β, need to be determined but it is not clear how this can be done. Since G and β are not considered as parameters in SKMS, which means they can't be estimated by the method itself. Therefore, we need to determine their values with extra effort. In our implementation of their method, we chose these parameters by looking at a range of β and G values, and selected the values which minimizes the errors with respect to K_1 parameter estimates, setting β to 0.2 and G to 17. We note, however, this procedure produces the best possible outcome for SKMS but is only possible for simulation data where we know the ground truth. In the simulations, the ground truth is required to help us choose the correct parameters for SKMS. But in practice, the ground truth of G and β are unknown. Therefore, SKMS was not implemented for the pig study data.

2.3.3 A Bayesian Spatial Mixture Model(SMM)

Model

Uncertainty of the parameters in our model is the natural consequence of modelling framework. Bayesian method allows us to quantify the uncertainty probabilistically. But classical methods cannot provide such detailed description of the uncertainty. Therefore, Bayesian framework outperforms classical approach in measuring the uncertainty. Here, we describe our proposed modelling and estimation approach. We denote the reconstructed activity concentration data by $\mathbf{Y}_i = (\mathbf{Y}_i^1, \ldots, \mathbf{Y}_i^T) \in \mathbb{R}^T$, for voxel i. Each data point $\mathbf{Y}_i^t$ corresponds to the reconstructed activity concentration at time t. We assume that the data $\mathbf{Y}_i$ can be grouped into G spatially homogeneous groups, where within each group, all voxels share the same kinetic rate parameters (or TACs) and their variations are only due to noise. The number of groups, G, is treated as unknown and is chosen by the information theoretic model selection criterion, Bayesian information criterion (BIC).

We use a mixture of multivariate Gaussian distribution with G components to model the noisy data. Given the ROI used for the analysis may include some voxels outside the myocardium, as well as some noisy voxels with very little activity uptake inside the myocardium, we allow one component to cluster these types of voxels. We call this the noise component. The Potts model (Wu 1982) is used to account for spatial correlation between the TACs. This is achieved by introducing a set of auxiliary random variables $\mathbf{z} = (z_1, \ldots, z_n)$, where z_i takes one of the values $1, \ldots, G$, and represents the group/cluster membership for each voxel. Mathematically, each noisy TAC is given by the mixture of T-dimensional Gaussian,

$$f(\mathbf{Y}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \beta) = \sum_{g=1}^{G} f(z_i = g|\beta) MVN(\mathbf{Y}_i|z_i = g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}),$$

where $f(z_i = g|\beta)$ is the marginal density of the Potts model, $MVN(\mathbf{Y}_i|z_i = g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma})$

is the density of the multivariate Gaussian, and β is the parameter that reflects the spatial strength between voxels. A value of 0 indicates independence between voxels, while larger values of β will tend to cluster all voxels into one cluster. The inference of β can refer to Appendix. The mean vector of the gth multivariate Gaussian component is denoted by $\boldsymbol{\mu}_g$, and $\boldsymbol{\Sigma} = diag(\sigma^{2,1}, \ldots, \sigma^{2,T})$ is the covariance matrix, assumed to be the same for all mixture components. One may relax this assumption to allow more general covariance structure; however, in our simulations studies, the same covariance structure worked well. Here, $\sigma^{2,t}, t = 1, \ldots, T$ denotes the variance at time t, and the data are assumed to be temporally independent as data were based on the reconstructed image at each time frame. The mixture model representation allows the error distribution to be more flexible. We refer to our model as the SMM.

We set the mean vector for the noise component g^* as

$$\boldsymbol{\mu}_{g^*} = (\mu^1_{g^*}, \ldots, \mu^T_{g^*}),$$

where the $\mu^t_{g^*}, t = 1, \ldots, T$ are unknown parameters. This component is dominated by noise, taking small values compared with other voxels with larger TAC measurements. For the remaining components $g = 1, \ldots, G-1$, we model the mean vector as a function of the solution to the ordinary differential equation (ODE) describing the one-compartment model (Morris, Endres, Schmidt, Christian, Muzic Jr., and Fisher 2004), although extensions to more compartments are straightforward. Hence, for $t = 1, \ldots, T$, we set

$$\mu^t_g = \frac{1}{\Delta\tau_t} \int_{\tau_{t,s}}^{\tau_{t,e}} K^g_1 \left[\hat{C}_p(s) \otimes \exp(-k^g_2 s) \right] ds, \tag{2.3}$$

where $\hat{C}_p$ is a measured blood input function and $\Delta\tau_t = \tau_{t,e} - \tau_{t,s}$ is the duration of the tth time frame. The parameters K^g_1 and k^g_2 are the kinetic rate parameters specific for group g.

For the pig study data analyses, we modified Equation 2.3 to account for spill-over effects,

$$\mu_g^t = f_{LV}^g \hat{C}_{LV}(t) + f_{RV}^g \hat{C}_{RV}(t) + (1 - f_{LV}^g - f_{RV}^g) \frac{1}{\Delta\tau_t} \int_{\tau_{t,s}}^{\tau_{t,e}} K_1^g \left[\hat{C}_p(s) \otimes \exp(-k_2^g s) \right] ds, \tag{2.4}$$

where f_{LV}^g and f_{RV}^g denote the component specific spill-over fractions for the left and right ventricle respectively. $\hat{C}_{LV}$ and $\hat{C}_{RV}$ were obtained by manually averaging the TACs from the appropriate ROIs. $\hat{C}_p$ was taken as $\hat{C}_{LV}$ multiplied by the plasma fraction, where the plasma concentration ratio was estimated based on blood samples drawn from previous studies. Equation 2.3 is equivalent to calculate each element of the mean vector by using convolution. While Equation 2.4 is obtained by incorporating spill-over fractions. In summary, both of they are ways to calculate the concentrations given the kinetic parameters.

Prior Specifications

For Bayesian inference, we need to specify prior distributions for the unknown parameters $K_1^g, k_2^g, \mu_{g^*}^1, \ldots, \mu_{g^*}^T, \sigma^{2,1}, \ldots, \sigma^{2,T}, \beta, g = 1, \ldots, G-1$. We assume independent and uninformative priors for all the parameters, so that the priors are broadly noninformative.

For the kinetic rate parameters, we use the uniform distribution for all g, $K_{1g} \sim \mathcal{U}(a_{K_1}, b_{K_1})$ and $k_{2g} \sim \mathcal{U}(a_{k_2}, b_{k_2})$, where $\mathcal{U}$ denotes uniform distribution. We have used $(a_{K_1}, b_{K_1}) = (0.3, \infty)$ and $(a_{k_2}, b_{k_2}) = (0, \infty)$ in our simulation studies. In real applications, one can sometimes get very abnormal rate constants, and a lower value of a_{K_1}, such as 0.1 used for our pig study data, might be appropriate. Setting a_{K_1} much lower than the plausible ranges for K_1 will result in additional clusters of the noise voxels being estimated with the kinetic model, and will unnecessarily add to computational cost. For the mean vector of the noise component $\mu_{g^*}^t \sim \mathcal{U}(0, \infty)$, $t =$

$1, \ldots T$. Setting the prior for K_1 sufficiently away from zero allows us to distinguish between the noise component and the non-noise components. We set prior $\beta \sim \mathcal{U}(0, b_\beta)$, where we take b_β to be 1, so as to include most of the plausible values of β. Finally, for the variance parameters $\sigma^{2,t}, t = 1, \ldots, T$, we follow the standard approach and use the usual vague conjugate prior with inverse Gamma distribution $\sigma^{2,t} \sim IG(a, b)$, where $a = 0.001, b = 0.001$ for an uninformative prior on $\sigma^{2,t}$. For the pig study data, we define independent priors for the additional parameters $f_{LV}^g \sim U(0, 1)$ and $f_{RV}^g \sim U(0, 1), g = 1, \ldots, G.$, and set $a_{K_1} = 0.1$ and $b_{K_1} = 1$.

Markov chain Monte Carlo(MCMC)

Bayesian inference proceeds via the posterior distribution, obtained by the simple product of the likelihood and the priors in Section 2.3.3. The likelihood function is given by

$$f(\mathbf{y}, \mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \beta) = f(\mathbf{z}|\beta) \prod_{i=1}^{n} f(\mathbf{Y}_i|z_i, \boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}). \tag{2.5}$$

The posterior distribution is given by the Bayes theorem as the product of the likelihood and the priors

$$f(\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \beta|\mathbf{y}) \propto \prod_{i=1}^{n} f(\mathbf{Y}_i|z_i, \boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}) f(\mathbf{z}|\beta) f(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \beta). \tag{2.6}$$

where the term $f(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \beta)$ denotes the prior distribution.

MCMC algorithms were developed to sample from the posterior distribution, using a combination of random-walk Metropolis-Hastings and Gibbs updates. Details for the implementation of the algorithm for the model in Equation 2.3 are given in the Appendix, the model of Equation 2.4 is a straight forward extension. Note that occasionally identifiability issues arise in the MCMC estimation of mixtures. Parameters from different components can switch labelling as a result of the invariance of the posterior distribution with respect to labelling. This is not an issue when only the

MAP estimates are required. The simplest way to handle this is by imposing certain ordering constraints on parameters (Fernandez and Green 2002), or via postprocessing of the MCMC output (Zhu and Fan 2016). In this chapter, the large number of mixture components was adequately handled using an efficient postprocessing algorithm for MCMC output (Cron and West 2011).

Determination of the Number of Components G

One of the uncertainties of the above model is the selection of the value of G, which plays a crucial role in the resulting parameter estimation. For model-based inference, in which a likelihood is readily available, a number of model selection criteria are available, including Bayesian information criterion (BIC), integrated completed likelihood (ICL), deviance information criterion (DIC) and Akaike information criterion (AIC). The BIC is often considered to be more parsimonious and is the frequently adopted measure of goodness of fit of the model (Steele and Raftery 2009). Our approach uses BIC as the criterion to determine the optimal value of G.

The BIC (Schwarz et al. 1978) is given as

$$BIC = -2 \log f(\mathbf{y}|G, \hat{\mathbf{z}}_{MAP}, \hat{\theta}_{MAP}) + DF \times (\ln(n) - \ln(2\pi)), \tag{2.7}$$

where $f(\mathbf{y}|G, \hat{\mathbf{z}}_{MAP}, \hat{\theta}_{MAP})$ is the likelihood function corresponding to the model with G components, evaluated at the MAP estimator of $\mathbf{z}$ and θ, the vector of all remaining unknown parameters. DF is the number of parameters to be estimated, which includes all the unknown kinetic parameters for each cluster, the variance parameters, and any hyperpriors which are estimated. n is the number of observations or voxels. BIC penalizes models with too many parameters against the maximized log-likelihood (or fit to the data). Optimal choice of G corresponds to the model with the smallest BIC value.

Implementation

We tuned the Gaussian random-walk proposal distributions to obtain an optimal overall acceptance probability of around 20%-40%. For the simulation data sets, we used $K_1^{'g} \sim N(K_1^g, 0.006^2)$, $k_2^{'g} \sim N(k_2^g, 0.001^2)$, $\mu_{g^*}^{'t} \sim N(\mu_{g^*}^t, 0.00013^2)$ and $\beta' \sim N(\beta, 0.002^2)$. For the pig data, we used $K_1^{'g} \sim N(K_1^g, 0.005^2)$, $k_2^{'g} \sim N(k_2^g, 0.003^2)$, $\mu_{g^*}^{'t} \sim N(\mu_{g^*}^t, 0.001^2)$, $f_{LV}^{'g} \sim N(f_{LV}^g, 0.01^2)$, $f_{RV}^{'g} \sim N(f_{RV}^g, 0.01^2)$ and $\beta' \sim N(\beta, 0.004^2)$.

For a full Bayesian analysis of a single simulated data set, we ran MCMC for 10000 iterations with the first 4000 iterations discarded as burn-in, and we keep every tenth sample due to high autocorrelation in the MCMC sample. Note that for MAP estimates, taken as the set of parameter values which gave the highest posterior probability during the MCMC run, we used only 6000 iterations, as MCMC chains can be expected reach modal regions of the posteriors quite quickly. For the real data set, 8000 iterations of MCMC were obtained with the first 6000 discarded as burn-in.

We first determine the number of components G, by running MCMC for $G = 2, \ldots, 26$ and computing the model selection criteria based on BIC (see Section 2.3.3). Then for a fixed G, we run posterior inference for a given dataset. For the evaluation of the proposed algorithm, we used 25 replicate simulations. For each replication at the chosen value of G, we obtain MAP estimators for comparison with SCF and SKMS. The performance of MAP is known to be worse than the posterior mean estimate, but it is sufficient to provide a good guide on the quality of the inference. The results are presented in Section 2.5.

2.4 Simulations and application

We apply our approach to simulated one-compartment PET perfusion data and compare the performance of our approach with both the standard voxel-wise curve-fitting approach and the spatial temporal approach (Saad, Smith, Hamarneh, and

Möller 2007), using the true kinetic parameters as the gold standard. We also apply our method to an in vivo pig study data.

2.4.1 Performance Evaluation

For a given noise realization, n, we compute kinetic parameter bias of voxel i using

$$b_i^n = (\boldsymbol{k}_i^n - \boldsymbol{k}_i^{Tr})/(\boldsymbol{k}_i^{Tr}), \tag{2.8}$$

where b_i^n is the bias of estimated kinetic parameter using the true kinetic parameter $\boldsymbol{k}_i^{Tr}$ as gold standard. Based on 25 noise realizations, we compute the mean bias $\bar{b}_i$, the mean squared bias $\bar{b}_i^2$, and standard deviation s_i bias for voxel i using

$$\bar{b}_i = \frac{\sum_{n=1}^{N} b_i^n}{N}, \quad \bar{b}_i^2 = \frac{\sum_{n=1}^{N} (b_i^n)^2}{N}, \quad s_i = \sqrt{\frac{\sum_{n=1}^{N} (b_i^n - \bar{b}_i)^2}{N-1}}, \tag{2.9}$$

where N is the total number of noise realizations. We perform the calculations described above for SCF, SMM and SKMS methods and make a comparison between methods.

2.4.2 Simulation: Dynamic Cardiac Perfusion PET

All the simulation studies were performed using an NCAT torso phantom (W.Segars 2000) which consists of heart, lungs, liver, and soft-tissue compartments. The left ventricle (LV) myocardium was segmented into 17 standard segments.(Cerqueira, Weissman, Dilsizian, Jacobs, Kaul, Laskey, Pennell, Rumberger, Ryan, Verani, et al. 2002) The simulation was based on ^{18}F-flurpiridaz, which is a new myocardial perfusion tracer that exhibits rapid uptake and longer washout in cardiomyocytes. Based on the one-tissue compartmental model, the TAC of the tissue concentration, $C_t(t)$,

was simulated using

$$C_t(t) = K_1[C_p(t) \otimes \exp(-k_2 t)], \tag{2.10}$$

where $C_p(t)$ is the blood input function, K_1 and k_2 are kinetic rate constants for the segment, and $\otimes$ denotes convolution operation. The input function used in the simulation was based on a previously published ^{18}F-flurpiridaz study(Alpert, Fang, and El Fakhri 2012). During the study, the LV input function was extracted with generalized factor analysis on dynamic series(El Fakhri, Sitek, Guérin, Kijewski, Di Carli, and Moore 2005; El Fakhri, Sitek, Zimmerman, and Ouyang 2006). This LV input function was treated as the plasma input function.

The kinetic parameters, i.e., $\boldsymbol{k} = (K_1, k_2)$, assigned to 17 segments were based on the realistic values obtained from PET perfusion studies on normal patients. (Alpert, Fang, and El Fakhri 2012) In order to mimic a myocardial defect, the segment located in the anterior wall was assigned with values by lowering K_1 and k_2 by 50% and 20%, respectively, of their original values. We added the $18th$ segment to include other voxels not part of the left ventricle myocardium. Table 2.1 shows the kinetic parameters assigned to all the 18 segments in the myocardium. The blood input function $C_p(t)$ and TACs for one normal (basal inferoseptal) and one defect (apex) segments are shown in Figure 2.2.

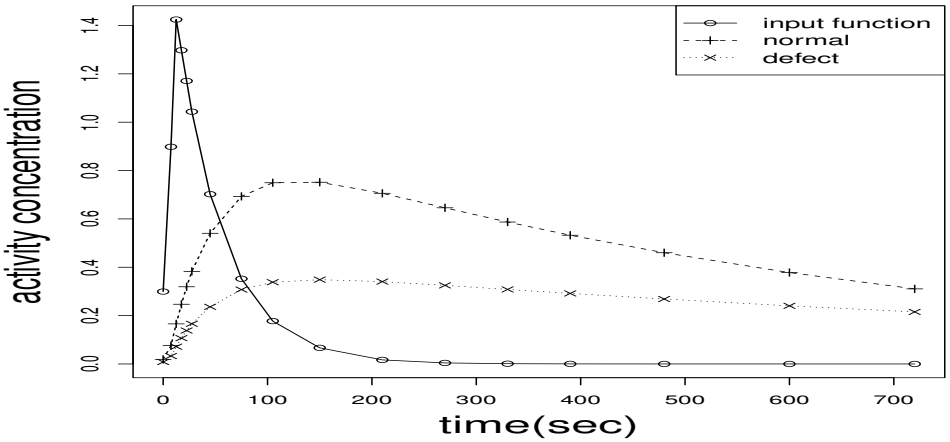


Figure 2.2: The input function and two TACs (one normal and one defect segment).

segment	K_1	k_2	segment	K_1	k_2
Basal anterior	0.3665	0.0627	Midinferior	0.7162	0.0799
Basal anteroseptal	0.6730	0.0740	Midinferolateral	0.8013	0.0997
Basal inferospetal	0.7656	0.0983	Midanterolateral	0.7720	0.0861
Basal inferior	0.7487	0.0635	Apical anterior	0.3653	0.0673
Basal inferolateral	0.9655	0.1032	Apical septal	0.8000	0.0861
Basal anterolateral	0.8021	0.0667	Apical inferior	0.7544	0.0717
Midanterior	0.3438	0.0541	Apical lateral	0.6816	0.1044
Midanteroseptal	0.7799	0.0877	Apex	0.3290	0.0554
Midinferoseptal	0.9016	0.0730	Others	0.7630	0.0820

Table 2.1: Segment names and their assigned K_1 values in mL/min/cc, k_2 values in 1/min (i.e.,the ground truth).

A system matrix corresponding to Philips Gemini PET-CT camera, which includes position dependent point spread function modelling, a forward-projection operator implemented using Siddon's method, line of response (LOR) normalization factors, and attenuation correction factors, was used to create noise-free sinograms from TACs. (Petibon, Ouyang, Zhu, Huang, Reese, Chun, Li, and El Fakhri 2013) The simulated sinogram data is equivalent to a 13-min dynamic PET scan with the framing scheme of 6 $\times$ 5s, 3 $\times$ 30s, 5 $\times$ 60s, and 3 $\times$ 120s frames. Twenty five dynamic PET noise realizations were generated. Both random and scatter events were not included in this study. The total number of events simulated in all the time frames is 50 M. The decay of the tracer was not simulated. Poisson noise was then added to each pixel in the sinogram based on the mean counts for the pixel. For each noise realization, the image reconstruction at each time frame was performed using standard ordered subset expectation maximization (Hudson and Larkin 1994)(OSEM) with 16 subsets and 8 iterations. No postreconstruction smoothing was applied. The physical dimension in the image reconstruction was 57.6cm $\times$ 57.6cm $\times$ 16.2cm, matrix dimension was 128 $\times$ 128 $\times$ 36, where the voxel size was 0.45cm $\times$ 0.45cm $\times$ 0.45cm.

2.4.3 Application: In-Vivo Pig Study

A pig with a body weight of 40 kg was scanned on a Siemens Biograph TruePoint PET/CT with the radiotracer ^{18}F-flurpiridaz. First, a planar x-ray topogram was performed to allow delineation of the field of view (FOV) and centering on the heart following CT and PET acquisitions. The cardiac CT was used for structure localization and later for attenuation correction during reconstruction of PET images. Emission PET data were acquired in 3D list mode and started concomitantly to the injection of ^{18}F-flurpiridaz, the injected activity was 11 mCI at the time of injection. List mode data were framed into dynamic series of 12 x 5, 8 x 15, 4 x 30, 5 x 60s. PET images were reconstructed using filtered back projection with minimal filtering (voxel size: 2.14x2.14x3 mm3, 55 slices). Attenuation correction was obtained from the CT images. Decay correction was applied and the first 10 min of the data are used for kinetic analysis. The input functions for the left and right ventricle were obtained by averaging the TACs from a manually defined region. A one-compartment model with spill-over correction was used. The described experiment was performed under a protocol approved by the Institutional Animal Care and Use Committee at the Massachusetts General Hospital.

2.5 Results and discussion

2.5.1 Model Selection

Figure 2.3 shows the BIC values (left panel) and the corresponding log likelihood (right panel) for competing models for a single noise realization. Horizontal lines in both subfigures denote the minimum and maximum values for BIC and the log likelihood respectively. The log likelihood values are expected to keep increasing with G, while the BIC penalizes the use of additional parameters in models with

larger G. Both the BIC and log likelihood changed dramatically from $G = 2$ to about $G = 10$, preferring models with larger G values, and this stabilized after around $G = 17$. Part of the changes seen here can be attributed to Monte Carlo errors. Thus, a parsimonious choice for G would be $G = 17$, representing the model with 16 TAC components and one noise component. Subsequent results for simulated data in this paper were generated by the model with $G = 17$. The same value of G was also found for the pig study data.

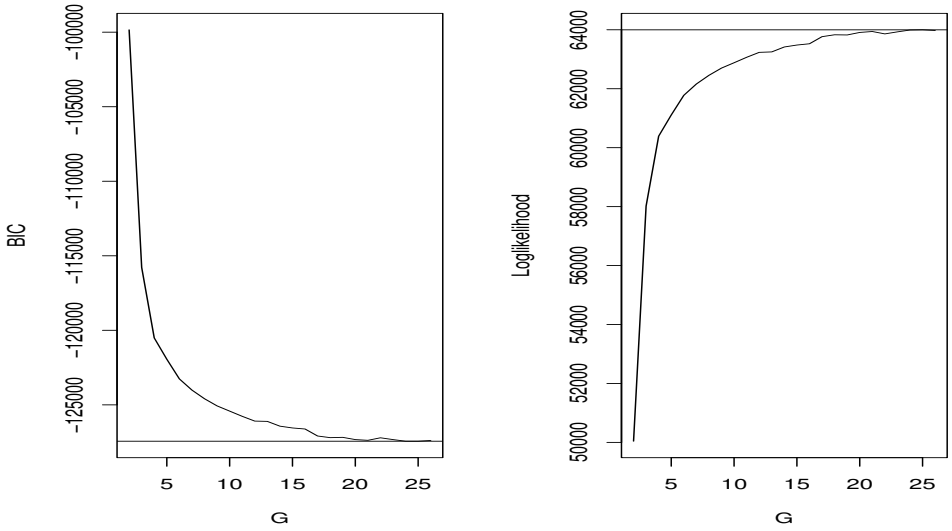


Figure 2.3: BIC values (left panel) and log likelihood (right panel) for $G = 2, 3, \ldots, 26$ in the spatial mixture model. The horizontal lines indicate the minimum and the maximum of BIC and log likelihood respectively.

2.5.2 Parameter Estimation and Comparison to Existing Methods

For the simulation data, Figures 2.4 and 2.5 show the corresponding marginal posterior distributions of K_1 and k_2 respectively, with vertical lines indicating the posterior mean, and the uncertainty of the estimates indicated by the spread of the distributions.

To assess the robustness of our estimation procedure and its performance against existing methods, we repeated our estimation procedure for 25 replicate data sets, obtained from the same simulation setup. We implemented the three competing

		K_1			k_2		
		min	med	max	min	med	max
$Abnormal$	SMM	0.006	0.04	0.12	0.06	0.25	1.12
	SCF	0.013	0.06	0.175	0.13	0.4	1.47
	SKMS	0.02	0.06	0.39	0.14	0.31	0.7
$normal$	SMM	0.006	0.03	0.29	0.01	0.09	0.37
	SCF	0.005	0.03	0.27	0.01	0.1	0.55
	SKMS	0.007	0.04	0.33	0.02	0.09	0.32
$Noise$	SMM	0.004	0.007	0.165	0	0	0
	SCF	0.0001	0.02	3.32	0	0.03	0.06
	SKMS	0.0005	0.05	0.22	0	2.96	13.05

Table 2.2: Summary statistics for each estimation.

methods SMM, SCF and SKMS. There was a single extremely large value of k_2 estimate from SKMS, which we omit from the results shown. As we know the ground truth in the simulation study, we evaluate the performance of three methods in different areas: abnormal tissues, normal tissues and noise region. Figure 2.6 shows the distribution of the mean squared biases of K_1 and k_2 in the abnormal, normal and noise regions, and the overall standard deviation of the biases. The computations were calculated according to Equation 2.9, with the exception that in the noise region, the bias was computed by setting the denominator of Equation 2.8, $\boldsymbol{k}_i^{Tr}$, to 1, since we cannot divide by zero.

Numerical results can be found in Table 2.2. Results for the abnormal ROI are shown in the first row of Figure 2.6. For K_1, the mean squared biases for SMM ranged from 0.006 to 0.12, with a median of 0.04. For SCF, the range was from 0.013 to 0.175, with a median of 0.06. For SKMS, the range was between 0.02 and 0.39, and the median was 0.06. For the k_2 estimation, the biases ranged from 0.06 to 1.12 for SMM, the median was 0.25. For SCF, the range was between 0.13 and 1.47, and the median was 0.4. For SKMS, the biases ranged from 0.14 to 0.7, and the median was 0.31.

The normal ROI is shown in the second row of Figure 2.6. For K_1, the mean

squared biases for SMM ranged from 0.006 to 0.29, with a median of 0.03. For SCF, the range was 0.005 to 0.27, with a median of 0.03. For SKMS, the range was between 0.007 and 0.33, and the median was 0.04. For the k_2 estimation, the biases ranged from 0.01 to 0.37 for SMM, with a median of 0.09. For SCF, the range was between 0.01 to 055, and a median of 0.1. For SKMS, the biases ranged from 0.02 to 0.32, and the median was 0.09.

The noise region is shown in the third row of Figure 2.6. For K_1, the mean squared biases for SMM ranged from 0.004 to 0.165, with a median of 0.007. For SCF, the range was 0.0001 to 3.32, with a median of 0.02. For SKMS, the range was between 0.0005 and 0.22, and a median of 0.05. For the k_2 estimation, the biases were approximately 0 for SMM. For SCF, the biases ranged between 0 and 0.06, and the median was 0.03. For SKMS, the biases ranged from 0 to 13.05, and the median was 2.96.

The last row of Figure 2.6 shows the standard deviations of the biases for K_1 and k_2. For K_1, the standard deviations of biases ranged from 0 to 0.34 for SMM, with a median of 0. For SCF, the range was between 0.008 to 1.70, with a median of 0.13. For SKMS, the range was between 0.073 to 0.63, and the median was 0.16. For the k_2 standard deviations of bias, the range was between 0 to 0.93 for SMM, with a median of 0. For SCF, the range was between 0.002 to 1.19, and a median of 0.01. For SKMS, the range was between 0.02 and 2.89, with a median of 1.20.

Figure 2.7 compares the bias and standard deviation of bias between SMM, SCF and SKMS for a single slice. The bias is calculated according to the first term in Equation 2.9, this is the average of the biases over 25 replications. Figure 2.7(a) shows the K_1 estimates. The biases ranged from -0.27 to 0.10, -0.28 to 0.09 and -0.33 to 0.08 respectively, for SMM, SCF and SKMS. Similarly, the standard deviations ranged from 0 to 0.26, 0 to 0.37 and 0 to 0.35 respectively. For the k_2 estimates in Figure 2.7(b), the biases ranged from -0.22 to 0.2, -1 to 1.06 and -0.46 to 0.28 respectively. The standard deviations ranged from 0 to 0.59, 0 to 0.86 and 0 to 2.89 respectively.

Figure 2.8 compares a single slice of the kinetic parametric images between SMM and SCF for the pig study. K_1 parameters were constrained to be between 0 and 1 in both SMM and SCF estimation, as unconstrained estimation lead to many physiologically implausible large values of K_1.

K_1 and k_2 are employed to evaluate the status of the tissue by doctors in disease diagnosis. For example, they are used to implement cancer detection. The more accurate their estimations are, the higher the accuracy of the diagnosis is.

2.5.3 Discussions

In this Section, estimation of K_1 and k_2 is discussed respectively. Specifically, different methods are compared in terms of the parameter estimations. The discussion is implemented in various perspectives, including point estimations in different regions and robustness in different regions. Then computational time was demonstrated. It concludes that our approach has several advantages over other methods.

This chapter proposes a novel method, SMM, that clusters voxel-wise TACs and estimates kinetic parameters simultaneously. Our modelling approach shares similarities to the recently proposed work (Lin, Haldar, Li, Conti, and Leahy 2014), where the mixture model was fitted to each voxel (while still borrowing information across nearby voxels) to overcome the issue of non-Gaussian error distributions. There is quite vital difference between their method and SMM. We allow several similar voxels to share the same parameter values, since separate mixture model fitted to each voxel introduces too many parameters, and thus lead to more estimation uncertainty. Our approach naturally allows us to constrain parameter estimates without the need to specify regularization parameters as in the usual Bayesian maximum a posterior (MAP) approaches. Finally, we allow the data to determine the most appropriate number of mixtures to fit to the data.

Our model-based approach offers several advantages, compared to other exist-

ing statistical approaches described above. Firstly, we require minimal user input in the algorithm, preferring to allow the data to dictate the optimal choices. One benefit of our modelling approach is in the determination of the optimal number of mixture components. Secondly, we also automatically compute the value of the smoothing parameter used in the MRF model. This unknown parameter is difficult to estimate, and in many applications of spatial modelling, the estimation of this parameter has not been carefully considered. The choice of both these parameters can have a big impact on results, since suboptimal choices will either result in higher bias or higher variance for the resulting parameter estimates. Finally, the Bayesian statistical framework allows us to quantify uncertainty probabilistically, since uncertainty in the model and parameters is a natural consequence of our modelling framework. An efficient MCMC algorithm allows us to provide parameter estimates, as well as uncertainty quantification simultaneously.

In Figure 2.3, the BIC values start to reach a minimum at around G =16 or 17. We chose to work with 17, but higher values of 18 or 19 will work equally well. These numbers are similar to the number of true segments simulated; however, we expect that this number can be different depending on the nature of the noise.

Figures 2.4 and 2.5 present the components' mean estimates for K_1 and k_2. Usually, the doctors use this information to determine whether abnormal region exist and where it is. Another attempt is to find one-to-one relationship between the clusters and true segments. In this case, it is difficult to make direct correspondences between the clusters we obtained with the true segments. The first four components correspond mostly to noise, the next four components correspond mostly to abnormal voxels and the rest belong to normal voxels. The discrepancy between the estimated values of K_1 and the truth is most obvious in the abnormal region, this is possibly a combination of partial volume effect, as well as misclassification of the normal voxels. Given that SMM outperforms the other two methods in the abnormal

region, we believe similar issueswith the data are affecting the other two methods also.

Despite the fact that it is difficult to make sense of individual clusters, aggregating the clusters can provide us with information about the larger ROIs. For instance, if we are interested in identifying the three regions of noise, abnormal and normal, we can aggregate the clusters according to $K_1 < 0.3$, $0.3 \leq K_1 < 0.6$ and $K_1 \geq 0.6$ respectively. A similar procedure can be used to classify the regions using the results from SCF and SKMS. In terms of misclassification rates, based on a single simulation data set, SMM classified 96.34% of noise voxels correctly, compared to 94.43% and 87.60% for SCF and SKMS. The misclassified voxels for SMM were all assigned to the abnormal voxels, this corresponds to the first four clusters in Figure 2.4. For the other two methods, they were spread between abnormal and normal voxels. For the abnormal region, SMM had a 100% correct classification, while this was only 62.68% for SCF and 52.82% for SKMS. All of the misclassifications in SCF and SKMS were allocated to noise. Finally, for the normal region, SMM, SCF and SKMS had 69.57%, 73.49% and 58.99% respectively for correct classifications, most of the misclassifications were found to be allocated to the abnormal region.

In terms of k_2 estimation, SMM is clearly better than the other two methods. This can be seen clearly in Figures 2.6 and 2.7. SKMS performed the worst, particularly in the abnormal and noise regions, their parameter estimation can be prone to very large biases. In the noise region, in particular, the median mean squared bias was around 2.96, while the other two methods were close to 0. Putting constraints on these parameters may prove useful.

For K_1 estimates, SKMS was marginally worse than the other two methods in terms of mean squared bias. In the abnormal region, SMM shows noticeably superior performance, where it can be seen in Figure 2.6, top row, the entire distribution of SMM is closer toward 0 than the other two methods. The difference in the normal

region is less obvious. The third row in Figure 2.6 shows the biases in the noise region; here, since SMM set K_1 in this region to 0, the graph can be interpreted by looking separately at the values of mean squared bias below $(0.3)^2 = 0.09$ and above. On average, voxels with bias greater than this value are essentially misclassified, i.e., they should be singled out as noise, but instead have significant values for the kinetic parameters. For SMM, there was an average misclassification rate of 3.66%, for SCF it is 5.71% and 11.65% for SKMS. SCF has the largest mean squared biases here, going up to 3.32, while the other two methods remain around 0.2.

In terms of the standard deviations of the bias, SMM performed the best, while SCF was the worst. The plot in the last row of Figure 2.6 shows that for K_1, the range for SCF goes up to 1.7, while for SMM and SKMS, this was only 0.34 and 0.63 respectively. In fact, 15% of the voxels estimated by SCF was greater than 0.34 (the largest value obtained by SMM), and 0.8% from SKMS.

The proposed method is clearly superior in terms of robustness, indicated by the substantially smaller standard deviation estimates, as can be seen in both Figure 2.6 and Figure 2.7. It also performed at least as good as, and sometimes better than the other two methods in terms of mean squared bias. In the single slice plot in Figure 2.7, where the mean of the raw biases was plotted, it is difficult to distinguish between the three methods. This is due to the fact that when raw biases are averaged, they will go toward zero as the effects of the large positive and negative biases cancel out. This will be true for all unbiased estimators regardless of how sensitive the estimations are to noise. In this sense, it is more useful to look at the mean squared or absolute biases.

In the real data application, it was not possible to compare the bias and standard deviations of the biases because the ground truth was not known. However, the parametric images shown in Figure 2.8 suggest that much smoother K_1 and k_2 images were produced by SMM compared to SCF. The white color in the K_1 images

indicates a value close to 1, which is the upper bound of the artificial constraint we used. It is clear from the figure that many values produced by the SCF method were simply truncated at this value. SMM estimation produced significantly less values close to the upper bound. The upper bound of 1 for K_1 is essentially arbitrary. For SMM estimation, if we remove this bound, we obtain two groups of voxels with physiologically implausibly high K_1 values whilst the rest of the voxel estimates remains unchanged, well below 1. However, in terms of the SCF estimation, raising the bounds to higher than 1 produced many more voxels between 1 and 2. However, since this was a resting pig, where the mean blood flow at rest is around 0.65 ml/min/cc, we do not expect flow to be above 1 at rest, so the SCF results with higher bounds would be difficult to interpret, since the higher values could also be due to spill-over from blood-pools, or voxels actually containing blood or noise.

In terms of computation, it took about 4 hours to complete all 6000 iterations for each noise realization of simulated data, using Matlab R2014b, running on a single node of the Linux computational cluster Katana at UNSW, Australia. This is equivalent to running on an average PC. The total number of voxels was 5746. We found that all the parameters converged quite quickly. For SCF and SKMS, the computational time was around 1 minute. For the pig study data involving 16821 voxels, and a longer time series involving 29 time points, the computational times were 23.5 hours and 6 hours for SMM and SCF respectively. We note that although SMM is computationally more expensive, it provides additional uncertainty estimation, which the other two faster methods do not. Parallel computation or other computational methods, such as variational Bayes (Attias 2000; McGrory, Titterington, Reeves, and Pettitt 2009), can be adopted to further speed up this process.

2.6 Summary

In this chapter, a novel spatiotemporal approach, SMM, is proposed to infer parametric PET images. By borrowing information from nearby voxels, SMM can be used to simultaneously estimate kinetic parameters and classify voxels with similar kinetic parameters into spatially homogeneous groups. We adopted the MRF to incorporate the spatial dependence of voxels. We developed an efficient MCMC algorithm for the computation, which estimates all unknown parameters, including the notoriously difficult spatial smoothness parameter β in the Potts model. The method provides parameter uncertainty estimation, as well as a principled way to determine the optimal number of voxel groups. We used simulated cardiac perfusion PET data to evaluate the performance of SMM and compared them with SCF and SKMS. SMM was substantially less sensitive to noise than the other methods, it also yielded an overall smaller bias than SCF and SKMS. In the pig study data, SMM was shown to produce smoother parametric images compared to the standard curve fitting.

Although simulation and experimental data were based on cardiac PET studies of a one-compartment model, the approach may benefit other dynamic PET procedures, as well as more complex compartmental models.

In the future, there are three directions to be considered to further develop our approach. First, we can relax the within cluster homogeneity assumption. This is easily achievable by relaxing the mean of the normal mixture to allow them to vary for each voxel observation. However, this substantially increases the number of parameters that needs to be estimated and presents a computational challenge. Second, we can consider the use of sinogram data rather than reconstructed data to estimate kinetic parameters, and this can reduce the additional noise introduced through the reconstruction step. However, this approach can be computationally challenging for full Bayesian analysis. Third, given its flexibility, SMM can be easily extended to

more advanced kinetic models, such as the two-compartmental tissue model without too much modification. Although we assess performance using simulations of cardiac perfusion PET imaging and demonstrate in vivo data for this application, our approach is not limited to this specific context and may also benefit other dynamic PET procedures as well as dynamic SPECT, dynamic contrast enhanced CT (DCE-CT), and dynamic contrast enhanced MR (DCE-MR).

In this chapter we mainly focus on how to estimate the kinetic parameters and other unknown parameters, while spatial strength in Potts model is estimated by thermodynamic integration method proposed in Green and Richardson (2002). The implementation of thermodynamic integration method can be found in Section 1.5. But inference of β still remains intractable in many scenarios, both for regular lattice and irregular lattice. I will propose a new method in next chapter to solve the inference of β when it is intractable, especially in large lattice.

Appendix

Markov chain Monte Carlo

We use MCMC for sampling from the joint posterior distribution of $\boldsymbol{z}$ and all other parameters, given by Equation 2.6. The prior distribution $f(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \beta)$ is taken as product of the individual prior components $f(K_1^1), \ldots, f(K_1^{G-1}), f(k_2^1), \ldots, f(k_2^{G-1})$, $f(\mu_{g^*}^1), \ldots, f(\mu_{g^*}^T), f(\sigma^{2,1}), \ldots, f(\sigma^{2,T}), f(\beta)$, as defined in Section 2.3.3.

The first term on the right side of Equation 2.6 is given by

$$
\begin{aligned}
&f(y_i|z_i = g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}) \\
&= (2\pi)^{-T/2}|\boldsymbol{\Sigma}|^{-1/2} \exp(-\frac{1}{2}(y_i - \boldsymbol{\mu}_g)'\boldsymbol{\Sigma}^{-1}(y_i - \boldsymbol{\mu}_g)),
\end{aligned}
$$

and the second term on the right side of the equation is given by

$$
f(\mathbf{z}|\beta) = \frac{1}{C(\beta)} \exp\{\beta \sum_{i \sim j} I(z_i = z_j)\}. \tag{2.11}
$$

This is the Potts model, where $\mathbf{z} = (z_1, \ldots, z_n)$, $I(\cdot)$ denotes indicator function taking value 1 if $z_j^{(l-1)} = g$ and 0 otherwise, and $i \sim j$ denotes the voxels j in the neighbourhood of voxel i. The partition function $C(\beta)$ is estimated offline using thermal dynamic integration (Green and Richardson 2002).

As the region of interest of PET image is not regular lattice, it is not proper to use the traditional neighborhood structure. In this scenario, we use an 8 nearest neighbour first order structure for the Potts model. Once neighborhood structure is defined, TDI can be implemented. Below I specifically demonstrate how to count the pairs in this case.

M is adopted to denote the mask of the PET image of interest. M is of the same dimensionality as PET image. Each entry of M could be 0 or 1, where 1 indicated the corresponding voxel is in region of interest, otherwise, voxel is out of the region of

interest. The generation of Potts model is different with regular lattice. Under irregular lattice, each voxel in ROI is conditionally dependent on its 8 nearest neighbors. Single-site update scheme is necessary. After one Potts model is generated, count the pairs in the Potts model. Only the pairs involving valid voxels are considered. Therefore, the difference between regular lattice and irregular lattice lies in generation of Potts model. According to the density function of regular lattice model in Equation 2.11, β in irregular lattice has a different meaning. But it still denotes spatial correlation between the voxels.

Our computational algorithm for the one-compartmental model in Equation 2.3 proceeds as follows:

Step 1 Set $l = 1$ and initialize parameters $K_1^{1,(0)}, k_2^{1,(0)}, \ldots, K_1^{G-1,(0)}, k_2^{G-1,(0)}, \boldsymbol{\mu}_{g^*}^{(0)}, \sigma^{2,1,(0)}, \ldots, \sigma^{2,T,(0)}, \mathbf{z}^{(0)}, \beta^{(0)}$.

Step 2 Update K_1^g, for $g = 1, \ldots, G-1$. Simulate a new value

$$K_1^{'g} \sim N(K_1^{g,(l-1)}, \delta_{K_1}^2)$$

and compute $\boldsymbol{\mu}_g'$ with $K_1^{'g}$, according to Equation (2.3). Set $K_1^{g,(l)}$ to $K_1^{'g}$ with probability α, where

$$\alpha = \min\{1, \alpha^\star\}$$

with

$$\alpha^\star = \frac{\prod_{i \in \{i: z_i^{(l-1)} = g\}} f(y_i | z_i^{(l-1)}, \boldsymbol{\mu}_g', \boldsymbol{\Sigma}^{(l-1)}) f(K_1^{'g})}{\prod_{i \in \{i: z_i^{(l-1)} = g\}} f(y_i | z_i^{(l-1)}, \boldsymbol{\mu}_g^{(l-1)}, \boldsymbol{\Sigma}^{(l-1)}) f(K_1^{g,(l-1)})}$$

.

Otherwise, set $K_1^{g,(l)}$ to $K_1^{g,(l-1)}$.

Step 3 Update k_2^g, For $g = 1, \ldots, G-1$. Analogously to Step 2.

Step 4 Update $\mu_{g^*}^t$, for $t = 1, \ldots, T$. Simulate a new value

$$\mu_{g^*}^{'t} \sim N(\mu_{g^*}^{t,(l-1)}, \delta^2_{\mu_{g^*}}).$$

Set $\mu_{g^*}^{t,(l)}$ to $\mu_{g^*}^{'t}$ with probability α, where

$$\alpha = \min\{1, \alpha^\star\}$$

with

$$\alpha = \frac{\prod_{i \in \{i: z_i^{(l-1)} = g^*\}} f(y_i | z_i^{(l-1)}, \boldsymbol{\mu}'_{g^*}, \boldsymbol{\Sigma}^{(l-1)}) f(\mu_{g^*}^{'t})}{\prod_{i \in \{i: z_i^{(l-1)} = g\}} f(y_i | z_i^{(l-1)}, \boldsymbol{\mu}_{g^*}^{(l-1)}, \boldsymbol{\Sigma}^{(l-1)}) f(\mu_{g^*}^{t,(l-1)}))}.$$

Otherwise, set $\mu_{g^*}^{t,(l)}$ to $\mu_{g^*}^{t,(l-1)}$.

Step 5 Update $\sigma^{2,t}$, for $t = 1, \ldots, T$. Simulate from the Inverse Gamma distribution

$$\sigma^{2,t,(l)} \sim IG\left(n/2 + a, \frac{1}{2}\sum_{i=1}^{n}(y_i^t - \boldsymbol{\mu}_g^{t,(l)})^2 + b\right).$$

Step 6 Update $\mathbf{z}$. Each $i = 1, \ldots, N$, compute

$$w_g = MVN(y_i; f(K_1^g, k_2^g), \boldsymbol{\Sigma}) \exp\{\beta^{(l-1)} \sum_{j, j \in \partial i} I(z_j^{(l-1)} = g)\}, \quad g = 1, \ldots G,$$

and normalize $w'_g = w_g / \sum_{g=1}^{G} w_g$, where $f(K_1^g, k_2^g)$ denotes Equation 2.3. ∂i denotes the set of neighbours of vertex i. Set $z_i^{(l)}$ according to the Multinomial distribution

$$z_i^{(l)} \sim MN(w'_1, \ldots, w'_G).$$

Step 7 Update β. Simulate a new value

$$\beta' \sim N(\beta^{(l-1)}, \delta^2_\beta)$$

and set $\beta^{(l)}$ to β' with probability α, where

$$\alpha = \min \left\{ 1, \frac{f(\mathbf{z}^{(l)}|\beta')f(\beta')}{f(\mathbf{z}^{(l)}|\beta^{(l-1)})f(\beta^{(l-1)})} \right\}.$$

Otherwise, set $\beta^{(l)}$ to $\beta^{(l-1)}$.

Step 8 set $l = l + 1$, if $l < L$, go to Step 2.

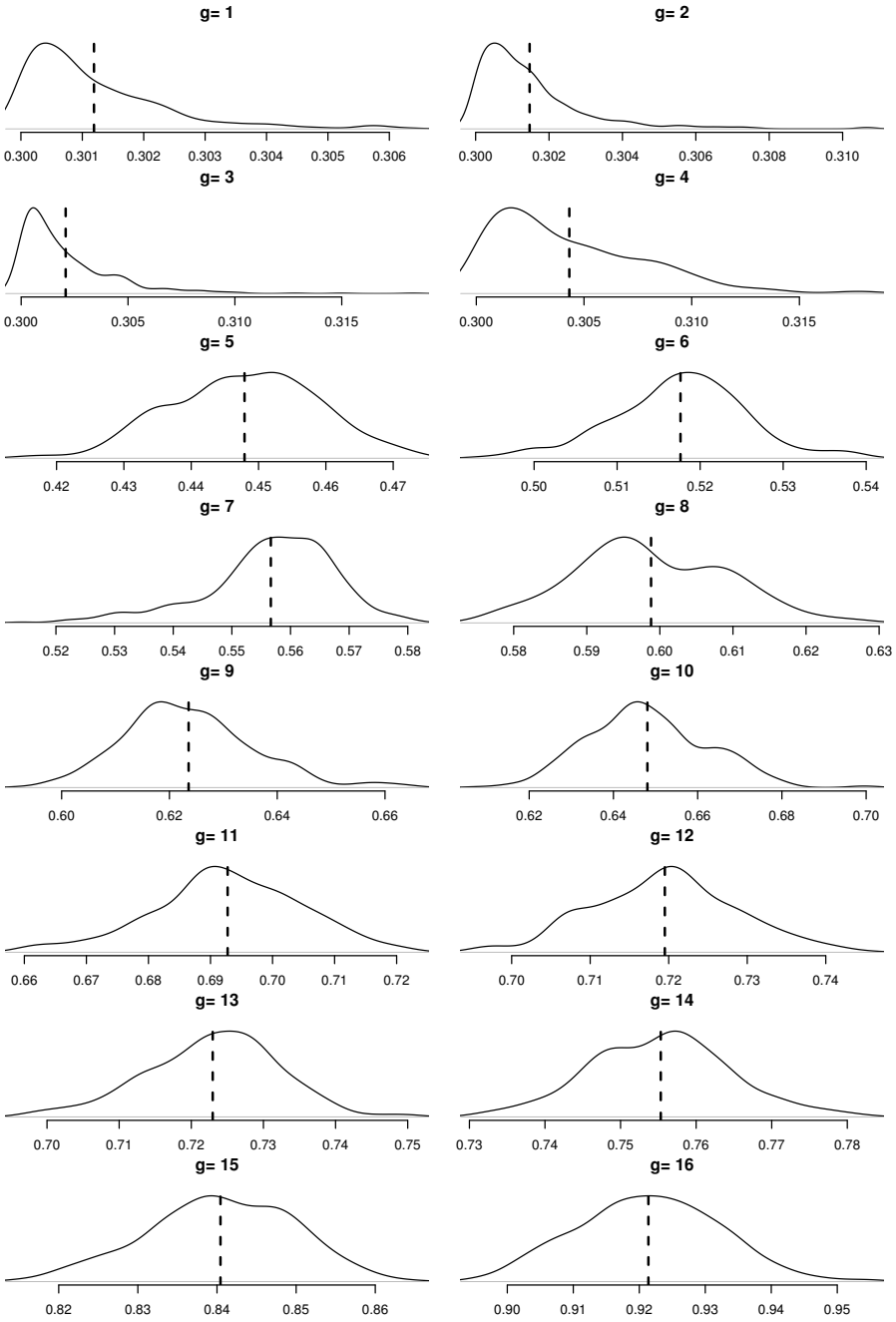


Figure 2.4: Marginal posterior density of K_1^g for $g = 1, \ldots, 16$ clusters. Vertical dashed line denotes corresponding posterior means. Based on a single noise realization of simulation data.

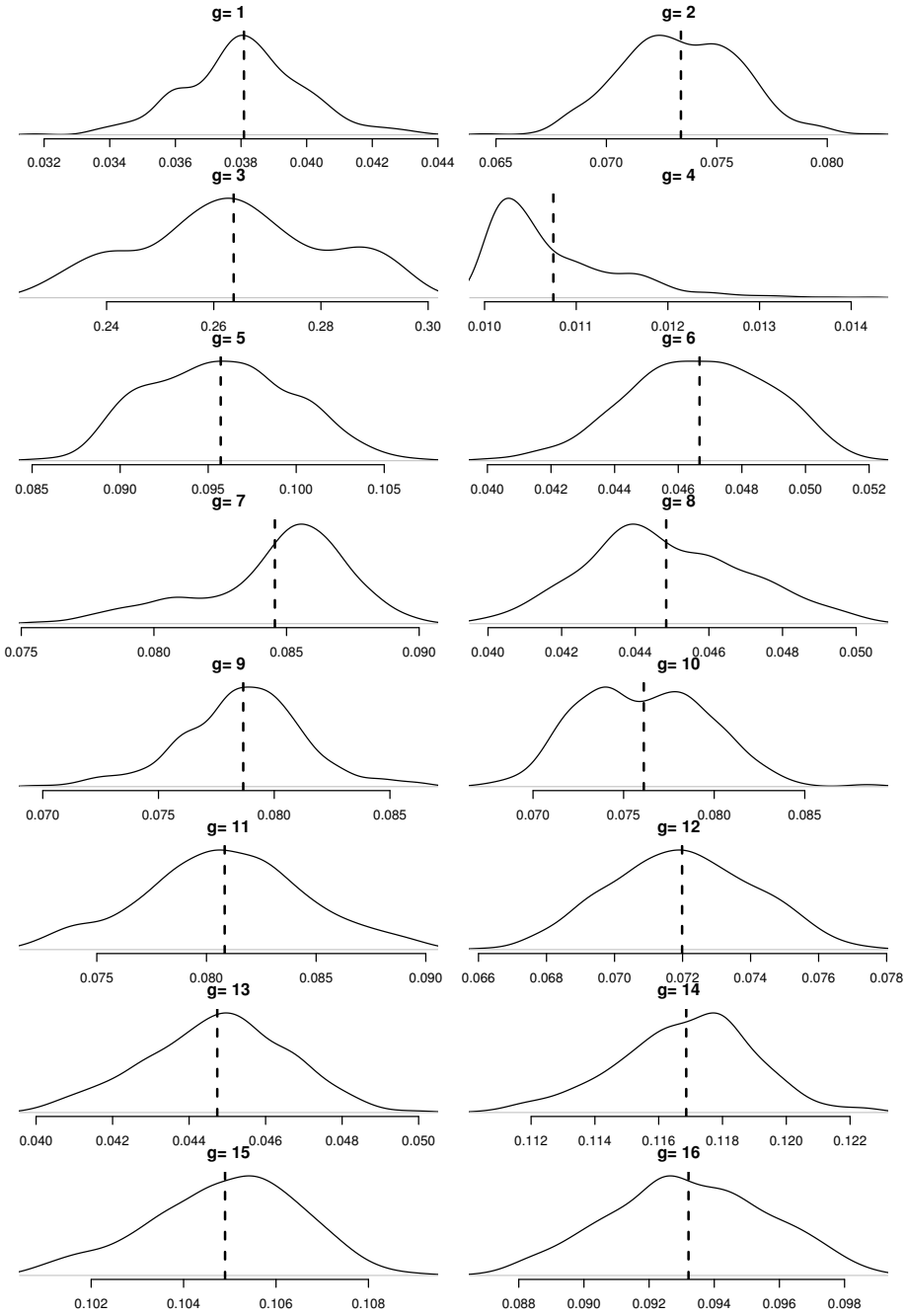


Figure 2.5: Posterior density of k_2^g for $g = 1, \ldots, 16$ clusters. Vertical dashed line denotes corresponding posterior means. Based on a single noise realization of simulation data.

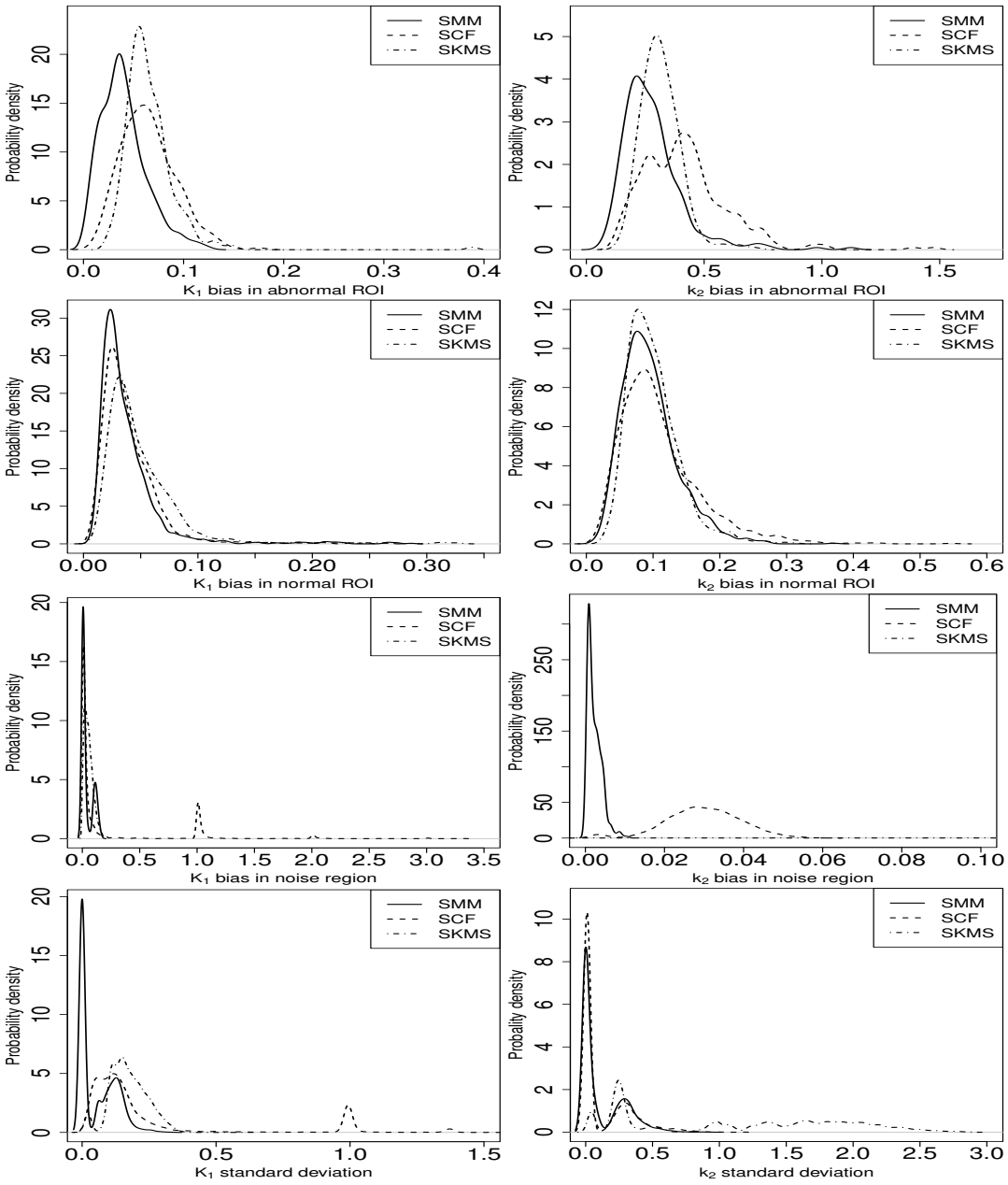
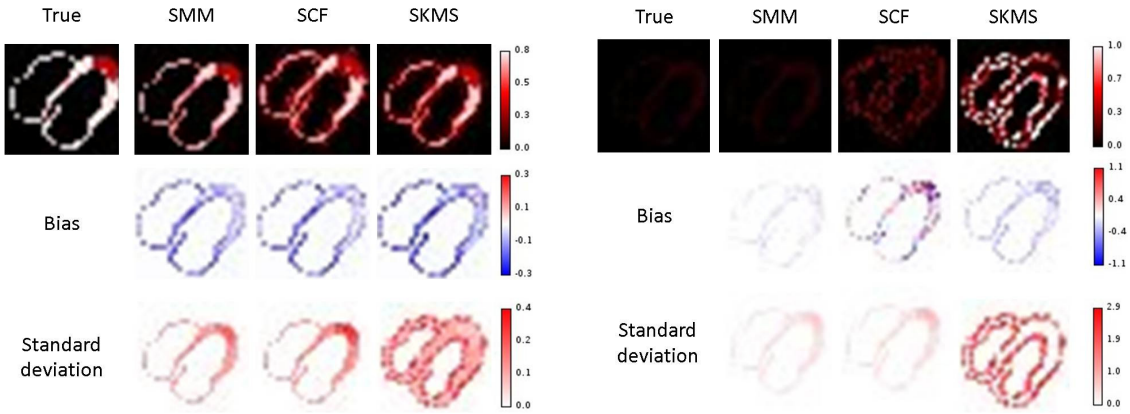


Figure 2.6: Distributions of the mean squared biases and standard deviation of biases for SMM (solid line); SCF (dashed line) and SKMS (dotted and dashed line). The first three rows show the mean squared biases for K_1 and k_2 in the abnormal, normal, and the noise ROIs respectively. The last row shows the standard deviation of the biases. Mean squared biases and the standard deviation of biases are calculated according to Equations 2.8 and 2.9, over 25 replicate simulation data sets.



(a) K_1 (b) k_2

Figure 2.7: Parameter estimates, bias and standard deviation of bias for a single slice of the image. Comparisons for K_1 (a) and k_2 (b) for 25 replications of simulation data.

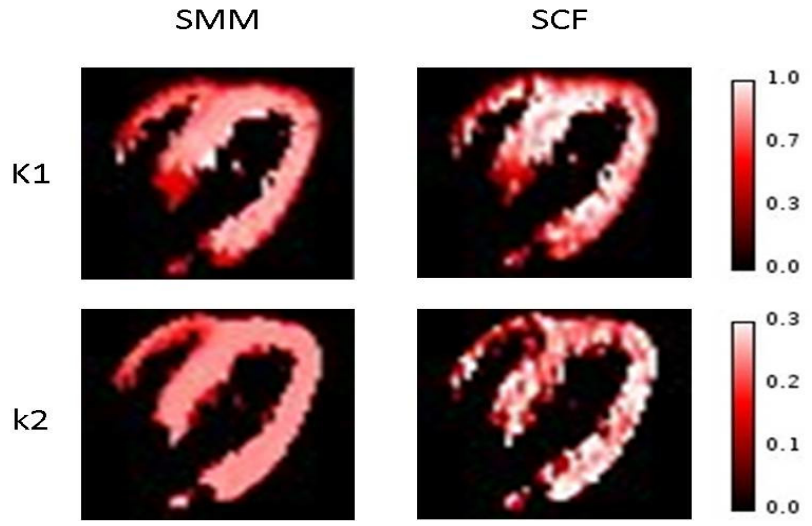


Figure 2.8: Parameter estimates for a single slice of the pig study data.

Chapter 3

A novel approach for markov random field with intractable normalizing constant on large lattice

In this chapter, normalizing constant issue is intensively discussed. An approach that can handle large size Potts model is proposed.

3.1 Introduction

Markov random field (MRF) models have an important role in modelling spatially correlated datasets. They have been used extensively in image and texture analyses (Nott and Rydén 1999, Hurn et al. 2003), image segmentation (Pal and Pal 1993, Van Leemput et al. 1999, Celeux et al. 2003, Li and Singh 2009), disease mapping (Knorr-Held and Rue 2002, Green and Richardson 2002), geostatistics (Cressie 1993) and more recently in social networks (Everitt 2012). In hidden Markov random field (HMRF) models, latent variables $\mathbf{z} = (z_1, \ldots, z_n)$ are introduced for each observed data $\boldsymbol{Y}_i, i = 1, \ldots, n$, where each pair $(\boldsymbol{Y}_i, z_i)$ has a corresponding spatial location. The MRF, and hence spatial interaction is modelled via $\mathbf{z}$ using an appropriate model,

such as, Potts or autologistic models.

In what follows, we describe our proposed methodology in terms of the q-state Potts model, although the method applies to other similar models, such as autologistic model and of course Ising model (a special case of Potts model when $q = 2$). In the Bayesian framework, the distribution $\pi(\mathbf{z}|\beta)$ can be seen as a prior distribution, and the hidden or missing observations $z_i, i = 1, \ldots, n$ are treated as unknown parameters to be estimated. For instance, a common form of the posterior distribution of a q-component spatial mixture model takes the form

$$\pi(\mathbf{z}, \beta, \theta|\mathbf{Y}) \propto \prod_{i=1}^{n} \pi(\boldsymbol{Y}_i|\theta, z_i)\pi(\mathbf{z}|\beta)\pi(\beta)\pi(\theta),$$

where $\pi(\boldsymbol{Y}_i|\theta, z_i)$ denotes the component distribution for $\boldsymbol{Y}_i$ conditional on the model parameters θ and z_i, $\pi(\theta)$ and $\pi(\beta)$ denote the prior and hyper prior for the unknown parameter vectors and $z_i = 1, \ldots, n$. Using the Potts model to define $\pi(\mathbf{z}|\beta)$, we have

$$\pi(\mathbf{z}|\beta) = \frac{1}{\mathcal{C}(\beta)} \exp\{\beta \sum_{i \sim j} I(z_i = z_j)\}, \tag{3.1}$$

where $i \sim j$ indicates that i and j are neighbours, and $\mathcal{C}(\beta) = \sum_{\mathbf{z}} \exp\{\beta \sum_{i \sim j} I(z_i = z_j)\}$ is the normalizing constant. $I(\cdot)$ is the indicator function, $I(z_i = z_j) = 1$ if $z_i = z_j$ is true, otherwise $I(z_i = z_j) = 0$.

Figure 3.1 (left panel) gives a pictorial illustration of a MRF with a first order neighbourhood structure, where each black site depends only on the four neighbouring gray sites on a 2D lattice. The 3D MRF is similarly defined with each site dependent on its neighbours on the left, right, front, back, above and below. The parameter β controls the degree of spatial dependence. See Wu (1982) for more illustrations on the Potts model.

For relatively small random fields (less than 10×10), the normalizing constant $\mathcal{C}(\beta)$ can be computed by summing exhaustively over all possible combinations of $\mathbf{z}$

for any given value of β. However, the calculation of $\mathcal{C}(\beta)$ becomes computationally intractable for large spatial fields. The posterior distribution $\pi(\theta, \mathbf{z}, \beta|y)$ is sometimes also referred to doubly-intractable distribution (Murray et al. 2006). This problem is well known in the statistical community, and has received considerable amount of attention in the literature, see Lyne et al. (2015) for a recent review.

Gelman and Meng (1998) used path sampling to directly approximate ratio of the normalizing constants, which can be used within posterior simulation algorithms such as MCMC, where only ratios are needed. Thermodynamic integration (TDI) is another approach which relies on Monte Carlo simulations. Green and Richardson (2002) for example adopted this approach by computing a look-up table offline. Other simulation-based methods can be found in Geyer and Thompson (1992), Gu and Zhu (2001), Liang (2007) and references therein. However, most methods utilising Monte Carlo become computationally expensive for very large lattices.

The pseudo likelihood (PL) method of Besag (1974) approximates $\pi(\mathbf{z}|\beta)$ as product of full conditional probabilities, where each term in the product is a full conditional of the neighbouring sites. The normalizing constant for each term in the product then becomes trivial to compute. Note however, that this is a type of composite likelihood (Lindsay 1988, Varin et al. 2011). The simplicity of the approach, coupled with its computational efficiency, makes the method still one of the most popular approaches in practice, particularly for large lattices. It has been noted in the literature that when the dependence is weak, the maximum pseudo-likelihood (MPLE) estimator behaves well and is almost efficient. In high dependence cases, the PL estimate is called into question, it has been shown to severely overestimate the dependence parameter, see Geyer and Thompson (1992). Hurn et al. (2003) comments that that PL should only be considered for dependences below the critical value, and its effects on modelling data with long range dependences are not clear. Cressie and Davidson (1998) proposed a similar method known as partially ordered

Markov models (POMMs), where the likelihood can be expressed as a product of conditional probabilities, without the need to compute the normalizing constant. POMM defines parent sites for each point on the lattice, and the point only depends on its parents. However, only a subset of MRFs are expressible as POMMs.

Reeves and Pettitt (2004) proposed a method for general factorizable models, which includes the autologistic and Potts model. This simple, yet effective approach is based on an algebraic simplification of the Markovian dependence structure, and is applicable to lattices with a small number of rows (up to 20). As a result of the factorisation, the normalizing constant can be computed over the much smaller subsets of $\mathbf{z}$, making such computations feasible. Friel et al. (2009) extended the work of Reeves and Pettitt (2004) to larger lattices by relaxing some of the dependence assumptions about $\pi(\mathbf{z}|\beta)$, so that the full model is a product of factors, each of which is defined on sublattices computed using the method of Reeves and Pettitt (2004). The sublattices are assumed to be independent, they term this reduced dependence approximation (RDA). The authors showed that RDA can be efficiently applied to the binary MRF, but concluded that the extension to the Potts model may not be computationally tractable. Another similar idea can be found in Bartolucci and Besag (2002), who also presented a recursive algorithm using the product of conditional probabilities, their method is only applicable to lattices of up to 12 rows and columns.

Finally, another class of methods completely avoid the computation of the normalizing constant by ingeniously employing an auxiliary variable, see Møller et al. (2006), Murray (2007), Murray et al. (2006). However, the method is computationally very expensive, as well as requiring perfect simulation (Propp and Wilson 1996). Liang (2010) proposed a double Metropolis-Hastings sampler, in which the auxiliary variable is drawn more efficiently. More recently, Liang et al. (2016) extended the exchange algorithm of Murray et al. (2006) to overcome the issue of obtaining per-

fect samples, using an importance sampling procedure coupled with a Markov chain running in parallel. Everitt (2012) proposed a sequential Monte Carlo method to deal with the same issue.

In many applications of MRFs, the size of the random field can be extremely large, the rows and columns of the lattices are often in the order of hundreds or even thousands. In this article, we propose a new approach which is able to handle arbitrarily large lattices. Our approach takes advantage of the conditional independence structure of the MRF defined on a regular lattice, and recursively divides the field into smaller sub-MRFs. Each sub-MRF is then approximated by another Potts model, with weaker spatial interaction as the size of the grid on the lattice increases.

3.2 A recursive decomposition method

Consider the first order neighbourhood structure defining the MRF. The left panel of Figure 3.1 depicts the location of the latent variable $\mathbf{z}$ defined on a regular lattice with a first order neighbourhood dependence structure. Here each black site depends only on its neighbouring grey sites. A natural consequence of this dependence structure is that, given the black sites, all the grey sites are independent, and vice versa. Thus conditioning on the grey sites, and decomposing the Potts model of Equation (3.1) we have

$$\pi(\mathbf{z}|\beta) \equiv \pi_{potts}(\mathbf{z}|\beta) = \pi(\mathbf{z}^{(1)}|\mathbf{z}^{(2)}, \beta)\pi(\mathbf{z}^{(2)}|\beta), \tag{3.2}$$

where $\mathbf{z}^{(1)}$ corresponds to the grey sites in Figure 3.1, left panel. The conditional independence property allows us to compute $\pi(\mathbf{z}^{(1)}|\mathbf{z}^{(2)}, \beta)$ directly as

$$\pi(\mathbf{z}^{(1)}|\mathbf{z}^{(2)}, \beta) = \prod_{i=1}^{n_1} \frac{\exp\{\beta \sum_{i\sim j} I(z_i^{(1)} = z_j^{(2)})\}}{\sum_{z_i^{(1)}=1,\ldots,q} \exp\{\beta \sum_{i\sim j} I(z_i^{(1)} = z_j^{(2)})\}} \tag{3.3}$$

producting over all n_1 observations in $\mathbf{z}^{(1)}$.

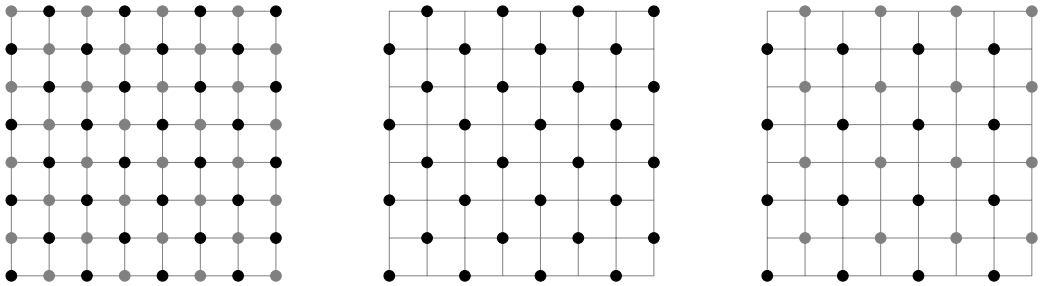


Figure 3.1: Left panel: a first order neighbourhood MRF, with black and grey points depicting $\mathbf{z}$. Each site only depends on the nearest four neighbours of the other color. Middle panel: the sub lattice $\mathbf{z}^{(2)}$. Right panel: $\mathbf{z}^{(2)}$ further divided into two parts based on the first order neighbourhood.

The field $\mathbf{z}^{(2)}$ is depicted by the middle panel in Figure 3.1. Here we approximate the dependence structure of this sub-MRF with another MRF model using the first order neighbourhood, as seen in the right panel of Figure 3.1. The dependence in $\mathbf{z}^{(2)}$ is weaker than the original MRF as the sites are further away from each other. Thus we approximate $\mathbf{z}^{(2)}$ again as a Potts model with first order neighbourhood $\pi_{potts}(\mathbf{z}^{(2)}|\alpha\beta)$. That is,

$$\pi(\mathbf{z}^{(2)}|\beta) \approx \pi_{potts}(\mathbf{z}^{(2)}|\alpha\beta) \tag{3.4}$$

with decay coefficient $0 \leq \alpha \leq 1$. Related references on long-range decay in spatial interactions can be found in Kosterlitz (1974), Wu (1982), Aizenman et al. (1988) and Luijten and Blöte (1995).

If the field in $\mathbf{z}^{(2)}$ is large, then we can apply the same principle to $\mathbf{z}^{(2)}$, as in Equation (3.2), to obtain $\mathbf{z}^{(3)}$ and $\mathbf{z}^{(4)}$, and so on. Until we end up with a Potts field for which computation for its normalizing constant becomes trivial. Hence, our approximation to the original Potts model by splitting the MRF into $2T$ fields is given by

$$\pi_{potts}(\mathbf{z}|\beta) \approx \left\{ \prod_{i \in \mathcal{I}} \pi(\mathbf{z}^{(i)}|\mathbf{z}^{(i+1)}, \alpha^{(i-1)/2}\beta) \right\} \pi_{potts}(\mathbf{z}^{(2T)}|\alpha^T\beta), \tag{3.5}$$

where $\mathcal{I} = \{1, 3, \cdots, 2T-1\}$. When $T = 0$, Equation (3.5) degenerates to the original Potts model. We term this approximation as recursive conditional decomposition approximation (RCoDA). In the approximation above only the last term needs the calculation of the normalizing constant, which is easy for small fields.

We have also considered more flexible forms for the decay structure by allowing a different decay coefficient per sublattice, so that

$$\pi_{potts}(\mathbf{z}|\beta) \approx \left\{ \prod_{i \in \mathcal{I}} \pi(\mathbf{z}^{(i)}|\mathbf{z}^{(i+1)}, \alpha_{(i-1)/2}\beta) \right\} \pi_{potts}(\mathbf{z}^{(2T)}|\alpha_T \beta), \tag{3.6}$$

where $\alpha_0 = 1$ and $0 < \alpha_{(i-1)/2} \leq 1$. The form in Equation (3.6) allows us to model arbitrary forms of decay for β. Our simulations using this form of decay (not shown in this article) produced similar results to those obtained via Equation (3.5), validating the power law decay within the Potts model setting. However, we note that the more flexible decay structure potentially provides a more flexible model than the standard Potts model.

Computational tractability dictates that we choose value of splits T, such that the Potts term on the right hand side of Equation (3.5) becomes small enough to be tractable. Simulation studies for varying T over a range of values of β showed that the results are largely insensitive to the choice of T. In practice, we can choose T so that the size of $\mathbf{z}^{(2T)}$ is no larger than 4×4. Note also that in relatively large fields with weaker spatial dependences, resulting in a large number of T, the factor α^{T-1} tends to zero. In these cases, the term $\pi_{potts}(\mathbf{z}^{(2T)}|\alpha^{T-1}\beta)$ in Equation (3.5) can be treated as an independent random field.

Equation (3.5) can be viewed as an approximation to the q-state Potts model. Alternatively, one can also view this model as being more flexible than the standard Potts model, particularly when one is interested in understanding different types of decay in the dependence when long range dependence is present. It is possible to

model the rate of decay differently to what is considered in this paper. Here the dependence at the T^{th} sublattice is modelled as $\alpha^T\beta$, where β is the global dependence parameter. This rate of decay was found by several authors in several difference applications, see Kosterlitz (1974), Wu (1982). We will investigate this assumption more closely in Section 3.4. Another important question when an approximation is used in place of the true likelihood, is whether this yields valid inference. Monahan and Boos (1992) introduces the notion of validity of posterior inference based on the correct coverage probability. We will also validate the use of RCoDA under this notion in Section 3.4.

3.3 Extensions to the second order structure

The most common neighbourhood structures in MRFs are the first and second order (Besag 1974). One of the most common types of the second order structure for 2D MRFs is shown in Figure 3.2, where each site has eight neighbours. Different definitions of neighbourhood structure affect the implementation of our algorithm. Suppose we have a pixel v. In 2D MRF, assuming v locates at (i, j), its second order neighbourhood includes $\{(i-1, j), (i-1, j+1), (i, j+1), (i+1, j+1), (i+1, j), (i+1, j-1), (i, j-1), (i-1, j-1)\}$. Figure 3.5 presents its second order neighbourhood in multiple scenarios, including 3D scenarios. Figure 3.3 and Figure 3.4 show the 18 and 26 neighbourhood structures in 3D.

Our proposed methodology requires that we split the entire lattices into non-overlapping sublattices. Here we use the"coding method" approach to obtain the sublattices (see Besag (1974), Winkler (2003) and Wilkinson (2005)). The "coding method" can be utilized to partition the lattices into several non-overlapping sublattices. Given a lattice, there could be more than one way to partition the lattice. For example, each pixel can be split as a sublattice. There is one value essential to

our method, which is called chromatic number. Chromatic number is the minimum number of sublattices one lattice can be partitioned into. More details on these can be found in Feng (2008) and Feng et al. (2012). The chromatic number for a first order structures is 2 in both 2D and 3D lattices, and 4, 4 and 8 in the second order neighbourhoods structures with 8 neighbours in 2D, 18 neighbours in 3D and 26 neighbours in 3D respectively.

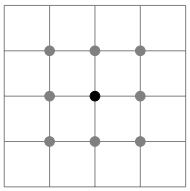


Figure 3.2: The second order structure in 2D MRF. Gray sites are neighbourhoods of the black site.

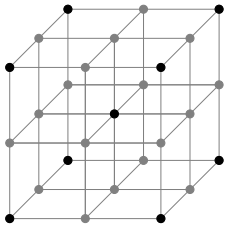


Figure 3.3: The first type of second order structure in 3D lattice: 18 neighbourhoods structure. All the gray sites are neighbourhoods of the black sites.

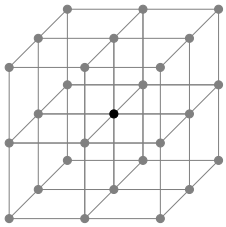


Figure 3.4: The second type of the second order structure in 3D lattice: 26 neighbourhoods structure. All the gray sites are neighbouhoods of the black site.

Figure 3.5: The second order structure in multiple scenarios.

Focusing on the case of the second order neighbourhood in 2D, we proceed by first identifying the 4 sublattices using the coding method. Figure 3.10(a) shows the corresponding lattice being split into 4 sublattices, corresponding to $(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{z}^{(3)}, \mathbf{z}^{(4)})$. Following the same decomposition as in Equation (3.2), we obtain

$$\pi(\mathbf{z}|\beta) = \pi(\mathbf{z}^{(1)}|\mathbf{z}^{(2)}, \mathbf{z}^{(3)}, \mathbf{z}^{(4)}, \beta)\pi(\mathbf{z}^{(2)}|\mathbf{z}^{(3)}, \mathbf{z}^{(4)}, \beta)\pi(\mathbf{z}^{(3)}, \mathbf{z}^{(4)}|\beta). \tag{3.7}$$

The first term on the right hand side of Equation (3.7) can be estimated as product of full conditionals similarly to Equation (3.3), see Figure 3.10(a) for the neighbourhood of $\mathbf{z}^{(1)}$.

The second term $\pi(\mathbf{z}^{(2)}|\mathbf{z}^{(3)}, \mathbf{z}^{(4)}, \beta)$ cannot be computed exactly, see Figure 3.10(b) for a pictorial depiction of the field for $(\mathbf{z}^{(2)}, \mathbf{z}^{(3)}, \mathbf{z}^{(4)})$. This term is the marginal

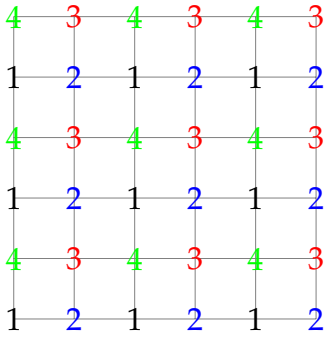


Figure 3.6: $(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{z}^{(3)}, \mathbf{z}^{(4)})$

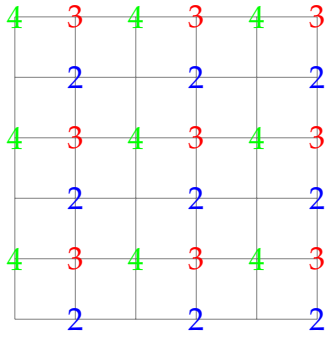


Figure 3.7: $(\mathbf{z}^{(2)}, \mathbf{z}^{(3)}, \mathbf{z}^{(4)})$

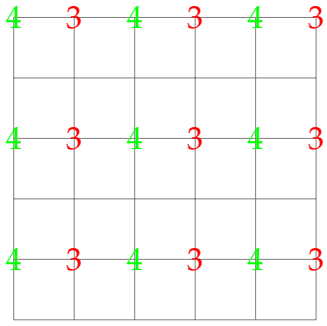


Figure 3.8: $(\mathbf{z}^{(3)}, \mathbf{z}^{(4)})$

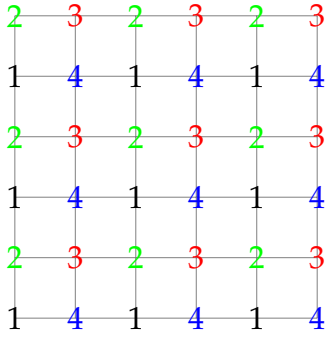


Figure 3.9: Alternative labelling

Figure 3.10: (a) Using the coding method approach, a 6×6 lattice is split into 4 sublattices. Each sublattice is labelled by corresponding number. (b) Sublattices with $\mathbf{z}^{(1)}$ removed. (c) Sublattice of $\mathbf{z}^{(3)}, \mathbf{z}^{(4)}$. (d) Alternative labelling, swapping 2 with 4 in (a).

likelihood of the second order neighbourhood Potts model with $\mathbf{z}^{(1)}$ integrated out, and would be as difficult to compute as the original problem. We consider two types of approximations for this term. In our first approximation, we assume conditional independence between $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$, thus allowing $\pi(\mathbf{z}^{(2)}|\mathbf{z}^{(3)}, \mathbf{z}^{(4)}, \beta)$ to be computed similarly to the first term, producting over all conditionally independent terms. We term this approach as RCoDA marginal (RCoDA-M). In our second approximation, using a similar approach to pseudo-likelihood approaches, we re-write the first two terms on the right hand side of Equation (3.7) as

$$\pi(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}|\mathbf{z}^{(3)}, \mathbf{z}^{(4)}, \beta) = \pi(\mathbf{z}^{(1)}|\mathbf{z}^{(2)}, \mathbf{z}^{(3)}, \mathbf{z}^{(4)}, \beta)\pi(\mathbf{z}^{(2)}|\mathbf{z}^{(1)}, \mathbf{z}^{(3)}, \mathbf{z}^{(4)}, \beta), \tag{3.8}$$

where both terms on the right hand side can be computed easily due to the conditional independence properties of these two subfields. We term this approach as RCoDA conditional (RCoDA-C).

Finally, the remaining field involving only $(\mathbf{z}^{(3)}, \mathbf{z}^{(4)})$ (as shown in Figure 3.10(c)), can again be approximated by a second order neighbourhood Potts model of the form $\pi_{potts}(\mathbf{z}^{(3)}, \mathbf{z}^{(4)}|\alpha\beta)$. This is done similarly to the first order case, and again modelling the spatial correlation with a decay term α. Note that the distances between sites only increase either between rows, or columns depending on the iteration of the recursion. For example, in Figure 3.10(c), the distance between the rows has doubled. But the distance between the columns is still unchanged. If we keep decompose the resulting lattice, the dependence can only decay in the rows' direction. According to the definition of β, it denotes the overall dependence in a lattice. Therefore, if the dependence only decay in one direction, we can't use the overall decay structure for β. To overcome this issue, we use alternate labelling between each iteration of the recursion. For example, the labels between 2 and 4 are swapped in 3.10(d) after each recursion, increasing the distance between columns after this iteration. In summary, for every two iterations the distances change uniformly over the entire field. Since the distance increases in both directions, in other words, uniformly, the decay structure can be used in the second order neighbourhood structure.

Although more complicated neighbourhood structures work under the same principle, their conditional independence structures may not be as easy to take advantage of, especially those with higher chromatic numbers.

3.4 Simulation study

In this section we perform extensive simulation studies to validate the proposed approach. Where possible, we compare our results with other existing methods.

Simulations are performed for both first and second order neighbourhoods defined on a regular 2D lattice.

3.4.1 First order neighbourhood

We first evaluate the performance of our estimation of β, for the first order neighbourhood dependences. We consider 2D lattices of sizes 32×32, 128×128 and 256×256. It is well known that the Potts model exhibits the so called phase transition, where for $\beta > \beta_{crit}$, the model will transit from disordered to ordered pattern or phase. This means that the sites will eventually all be in the same state as β increases. For a general q-state model, the precise value of the critical value is difficult to determine. For the Ising model ($q = 2$) defined over 2D lattice, Potts (1952) suggests setting $\beta_{crit} = \log(1 + \sqrt{q})$, with $\beta_{crit} \approx 0.88$ for $q = 2$ and $\beta_{crit} \approx 1.01$ for $q = 3$. Barkema and de Boer (1991) suggests setting the critical values to 0.44 for $q = 2$ and 0.503 for $q = 3$. This is not compatible with the conclusion in Potts (1952) because they use different definitions of Potts model. In Barkema and de Boer (1991) Potts model is defined as $\pi(\mathbf{z}|\beta) = \exp\{\beta \sum_{i\sim j} z_i z_j\}/\mathcal{C}(\beta)$, where $z_i \in \{-1, 1\}$. This defination is different with Equation 3.1. But in essence, they have same conclusion on critical value. Here we will restrict our analyses to β below the critical values recommended by Potts (1952), and consider the set of values $0.1, 0.2, \ldots, 0.8$ for β .

For each value of β, we simulated 200 replicate datasets from the q-state Potts model using MCMC. Data from the Potts model was generated using Gibbs sampling using purpose written codes in Matlab. The final iterate after 5000 MCMC steps was then used as the observed data from the Potts model. Throughout our implementations of RCoDA, the priors $\beta \sim U(0, 0.9)$ and $\alpha \sim U(0, 1)$ were used, and MCMC was used to obtain posterior estimates for both α and β. Approximately 6000 iterations with the first 2000 iterations as burn in were sufficient to obtain convergence for all models implemented. For lattices of sizes 32×32, 128×128 and 256×256,

we decomposed the field until the smallest one is 4×4, corresponding to $T = 6, 10, 12$ respectively for the three different sized lattices.

For comparison, we also implemented PL (Besag 1974), TDI (Green and Richardson 2002) and RDA (Friel et al. 2009) methods. With the exception of RDA, all methods were implemented in Matlab, RDA was implemented using the modified codes kindly provided by the authors. Although RDA can be applied to lattices of any size, RDA was only implemented for the small field with $q = 2$. Because the method was developed for $q = 2$, and the available codes were not applicable to larger sized lattices. When m is too large, the codes can't be successfully implemented. For all the implementation of RDA, we choose $m_1 = 10$.

TDI was implemented following the procedures in Green and Richardson 2002. We choose the grid of β as follows: the interval of β is chosen as $[0, 3.1]$ and the interval was equally segmented with step size equal to 0.01. The Monte Carlo samples of Potts model were generated by using Gibbs sampler in Feng (2008). We run the Gibbs sampler for 5000 iterations which ensures the target distribution has converged. Then we continued to run the Gibbs sampler for another 5000 iterations, generating 5000 samples of Potts model.

Table 3.1 shows the root mean squared error of the β estimation for $q = 2$ and $q = 3$, for lattice sizes of 32×32, 128×128 and 256×256. The results are very similar for the different values of q and β. While all the methods obtained small root mean squared error estimates, the performances in larger lattices between the different methods were almost indistinguishable.

We have also investigated the effects of using different values of T, i.e., the number of times to split the random field, and again the results were broadly insensitive to this specification. Numerical results are omitted from presentation here.

To further investigate the appropriateness of using the decay rate of $\alpha^T\beta, 0 < \alpha < 1$ over the T splits of the random field, we separately estimated the value of

β			0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.800
32^2	RCoDA	q=2	0.039	0.047	0.048	0.053	0.053	0.057	0.057	0.051
		q=3	0.039	0.047	0.051	0.051	0.053	0.049	0.049	0.046
	PL	q=2	0.043	0.046	0.044	0.049	0.048	0.046	0.046	0.053
		q=3	0.044	0.046	0.047	0.049	0.051	0.044	0.042	0.047
	TDI	q=2	0.040	0.042	0.042	0.043	0.038	0.037	0.036	0.032
		q=3	0.040	0.044	0.045	0.045	0.045	0.039	0.034	0.034
	RDA	q=2	0.040	0.043	0.042	0.043	0.038	0.037	0.036	0.032
		q=3	-	-	-	-	-	-	-	-
128^2	RCoDA	q=2	0.011	0.012	0.011	0.014	0.012	0.014	0.016	0.017
		q=3	0.011	0.012	0.011	0.011	0.011	0.012	0.012	0.013
	PL	q=2	0.011	0.011	0.011	0.012	0.011	0.011	0.012	0.012
		q=3	0.011	0.012	0.011	0.011	0.011	0.010	0.011	0.011
	TDI	q=2	0.011	0.011	0.010	0.010	0.009	0.009	0.008	0.007
		q=3	0.011	0.011	0.010	0.010	0.010	0.009	0.008	0.008
256^2	RCoDA	q=2	0.006	0.005	0.006	0.006	0.006	0.008	0.009	0.013
		q=3	0.006	0.006	0.005	0.007	0.006	0.006	0.007	0.007
	PL	q=2	0.006	0.005	0.006	0.006	0.005	0.006	0.006	0.006
		q=3	0.006	0.005	0.006	0.006	0.005	0.006	0.006	0.006
	TDI	q=2	0.005	0.005	0.006	0.005	0.005	0.004	0.004	0.004
		q=3	0.006	0.006	0.005	0.005	0.005	0.005	0.004	0.004

Table 3.1: Root mean squared error of β for a first order neighbourhood dependence. Based on 200 simulated data sets for each 32×32, 128×128 and 256×256 lattices. $q = 2$ and $q = 3$.

β_T for each T^{th} sublattice using the full Potts model, PL was used to obtain the estimate for β_T. Figure 3.11 shows the averaged estimate of β_T and $\alpha^T\beta$ over 200 data sets simulated at $\beta = 0.5, 0.6, 0.7, 0.8$. The model used here was a $q = 2$ state Potts model over a 256×256 lattice for different values of β. For very small lattice sizes or very weak dependences, the PL estimate is not reliable, possibly due to excessive boundary influence. Therefore, we show the decay for T up to 8, corresponding to the smallest estimated lattice size of 16x16. Figure 3.11 shows the curve $\alpha^T\beta$ and β_T for $T = 0, \ldots, 8$, with the true $\beta = 0.5, 0.6, 0.7, 0.8$. The graphs show a good match between the estimated β_T and $\alpha^T\beta$, suggesting such a decay structure is appropriate.

Figure 3.12 shows the 95% empirical coverage probabilities, estimated over varying values of β and for $q = 2$ and $q = 3$ on a 32×32 lattice. For a given value of β, we simulated 200 datasets based on β. For each dataset, a 95% posterior credibility

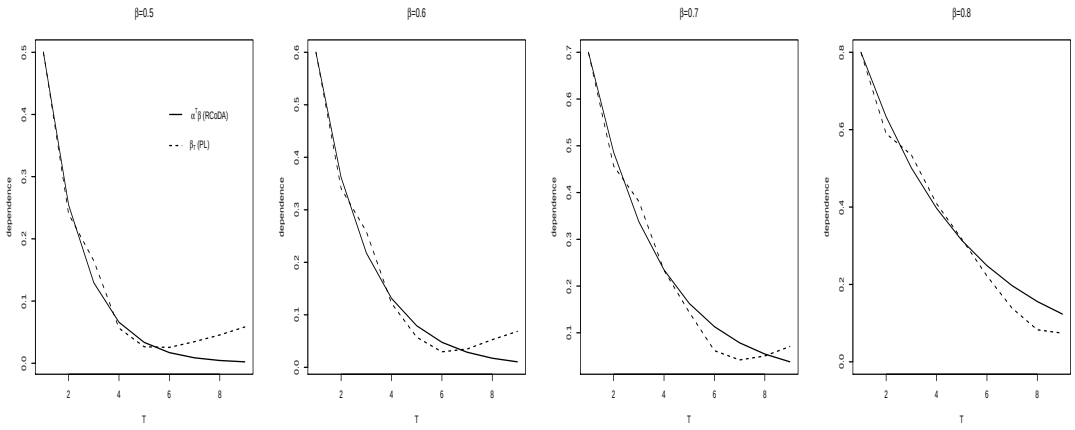


Figure 3.11: Plot of $\alpha^T\beta$ using RCoDA (solid line) against β_T (dashed line), estimated by PL for the T^{th} sublattice, for a $q = 2$ model over 256×256 lattice, and at $\beta = 0.5, 0.6, 0.7, 0.8$ from left to right.

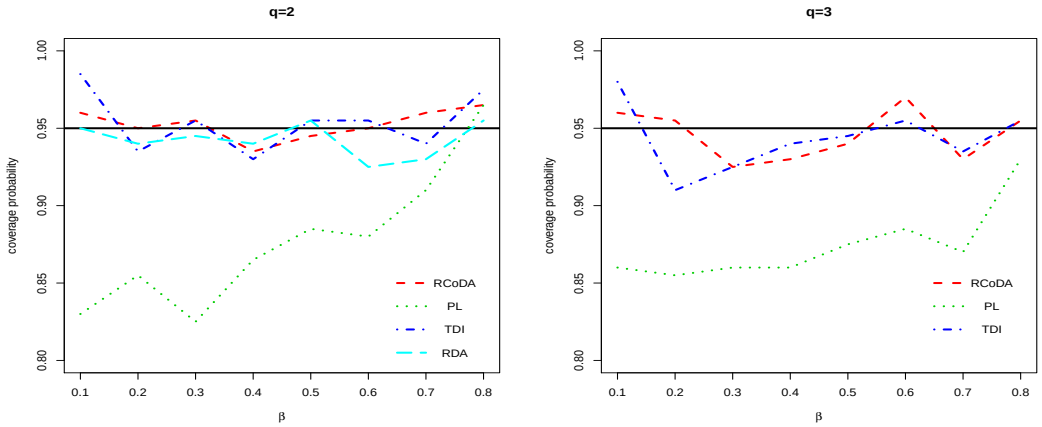


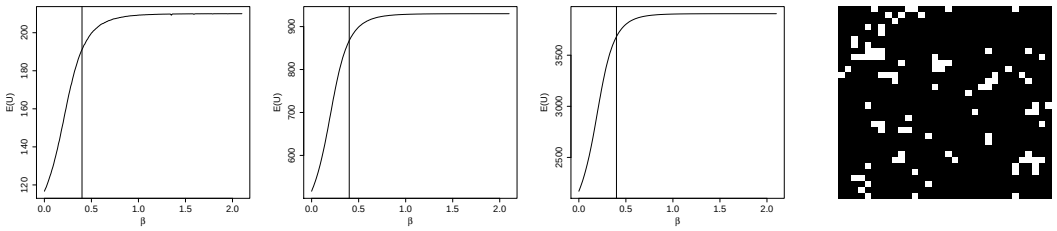
Figure 3.12: 95% empirical coverage probabilities for the 32 × 32 lattice with a first order neighbourhood, $q = 2$ (left) and $q = 3$ (right).

interval of β is recorded and the proportion of intervals containing the initial value of β was recorded. It can be seen that the coverage probabilities of RCoDA, TDI and RDA are all close to the nominal level, suggesting that these methods yield valid inferences, see Monahan and Boos (1992). For TDI, this is expected, since the likelihood is exact. However, the coverage of PL is noticeably smaller than the nominal level, particularly at the weaker dependences. The phenomenon also corresponds to a generally narrower posterior variance estimate from our simulation results (not shown here). This is unsurprising since the pseudo-likelihood is a special case of

composite likelihoods, and direct computation using MCMC can result in posterior variances that are too small, see Varin et al. (2011) and Pauli et al. (2011) for discussions.

3.4.2 Second order neighbourhood

For the second order neighbourhood study, we again considered the $q = 2$ and $q = 3$ state Potts model over 32×32, 128×128 and 256×256 lattices. The RDA method was omitted here. In order to determine the critical value for β, we monitored the changes in the value of $E(U(\mathbf{z})|\beta)$, where $U(\mathbf{z}) = \sum_{i\sim j} I(z_i = z_j)$. $U(\mathbf{z})$ is the total number of pairs in $\mathbf{z}$. Figures 3.13(a)-(c) presents the changes in $E(U(\mathbf{z})|\beta)$ as β changes, for a number of different sizes of lattices. The estimated value of $E(U(\mathbf{z})|\beta)$ was obtained by Monte Carlo method similar to that used for TDI. It can be seen that the estimates stabilise around 0.4. Figure 3.13(d) presents one realization of the Ising model at $\beta = 0.4$, where the figure begins to be dominated by one colour, which is a sign of phase transition. Therefore, we restrict our study to $\beta < 0.4$. See also Green and Richardson (2002), Gelman and Meng (1998) and Moores et al. (2015) who discusses the uses of $E(U(\mathbf{z})|\beta)$ in inference.



(a) 8×8 MRF (b) 16×16 MRF (c) 32×32 MRF (d) Ising model

Figure 3.13: Estimates of $E(U(\mathbf{z})|\beta)$for Ising model over different lattice size: (a) 8×8 (b) 16×16 and (c) 32×32. Vertical line correspond to $\beta = 0.4$. (d) shows simulation of one realization of the Ising model at $\beta = 0.4$.

Table 3.2 shows the root mean squared errors of the β estimation for $q = 2$ and 3 over the varying lattice sizes, using RCoDA-C, RCoDA-M, PL and TDI. The results

were computed over 200 simulated data sets at $\beta = 0.1, 0.2$ and 0.3. The results suggest no significant difference in performance over the values of q. For the larger lattices, RCoDA-C, PL and TDI all performed similarly in terms of root mean squared errors. RCoDA-M, which assumes marginal independence, was worse overall compared to RCoDA-C, which uses a partial pseudo-likelihood. For the 32×32 lattice, RCoDA methods performed worst, this suggests that it is not suitable to use decomposition in second order neighbourhoods when lattice sizes are too small, since the method of splitting requires that we should have at least several iterations. So when the lattice size is too small, the relative bias will be larger.

Figure 3.14 shows the empirical coverage probabilities computed under similar conditions to those for first order neighbourhood simulations. Again, we see that the PL methods do not achieve good coverage, where as both RCoDA and TDI achieve good coverage, with RCoDA-C performing fairly consistently better.

		32^2				128^2				256^2			
β	q	RCoDA-M	RCoDA-C	PL	TDI	RCoDA-M	RCoDA-C	PL	TDI	RCoDA-M	RCoDA-C	PL	TDI
0.1	2	0.029	0.029	0.026	0.025	0.008	0.008	0.003	0.006	0.005	0.004	0.003	0.003
	3	0.031	0.033	0.029	0.027	0.004	0.004	0.003	0.007	0.004	0.004	0.003	0.003
0.2	2	0.036	0.031	0.025	0.024	0.015	0.007	0.003	0.006	0.013	0.004	0.003	0.003
	3	0.031	0.029	0.026	0.025	0.009	0.004	0.003	0.006	0.009	0.004	0.003	0.003
0.3	2	0.038	0.027	0.021	0.018	0.032	0.007	0.002	0.004	0.031	0.004	0.002	0.002
	3	0.031	0.025	0.020	0.019	0.023	0.004	0.003	0.004	0.023	0.004	0.003	0.002

Table 3.2: Root mean squared error of β for a second order neighbourhood dependence. Based on 200 simulated data sets for each 32×32, 128×128 and 256×256 lattices. $q = 2$ and $q = 3$.

3.5 Real data application

Texture analysis is an important branch of computer vision and pattern recognition. Texture analysis characterises regions on an image by their texture content. Any measure (such as gray scale on a photographic image) that provides a value at each pixel, can be used to segment the image into regions of similar textures.

Natural scenes such as sand, stones, grass, leaves, bricks and other objects cre-

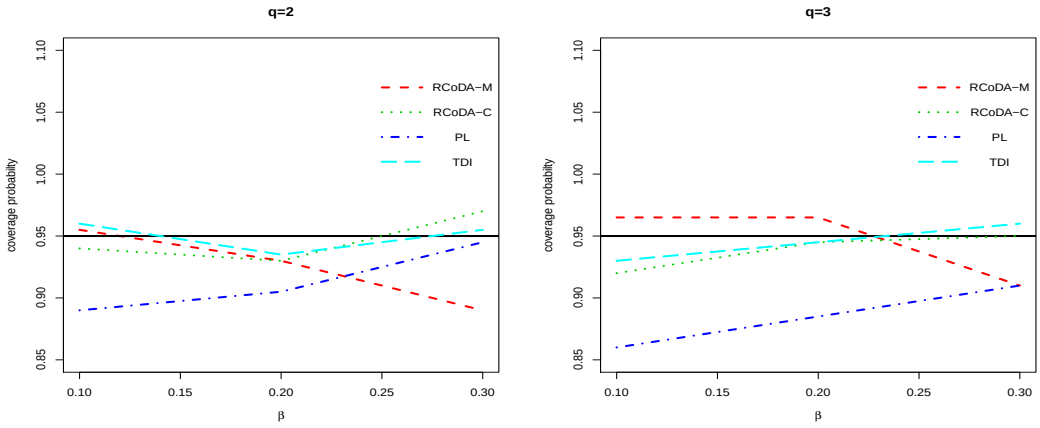


Figure 3.14: 95% empirical coverage probabilities for the 32×32 lattice with a second order neighbourhood, $q = 2$ (left) and $q = 3$ (right).

ate a textured appearance in images. Texture gives us information about the spatial arrangement of intensities in an image. In robotic applications, interest may be in separating weed from grass in automatic weed control systems (Watchareeruetai et al. (2006)). In texture synthesis, the primary purpose is to reproduce and enlarge the texture in a given image so that the natural and synthetic texture will be visually indiscernible. Haindl et al. (2012) proposed the use of Markov random field models for this purpose. The images are available online, at `http://sipi.usc.edu/database/database.php?volume=textures`. The image was originally studied in Brodatz (1966). Without loss generality, we take the first 256 rows and 256 columns as our data of interest, see Figure 3.15.

We use a two-component Gaussian mixture model to model the grass data. The posterior distribution is given in Equation 1.2.2, with $\pi(y_i|\theta, z_i)$ given by the component Normal distribution according to z_i, with parameters μ_j and $\sigma_j, j = 1, 2$ indicating the component mean and variance. The distribution of $\mathbf{z}$ is the Ising model as given in Equation 3.1. We set prior distributions for $\mu_j \sim N(0.5, 100^2)$ and $\sigma_j^2 \sim IG(0.001, 0.001), j = 1, 2$. The prior of β is set to to Uniform distribution between 0 to 4. A Metropolis-Hastings algorithm was used in the MCMC. 6000 MCMC iterations was implemented, while the first 2000 iterations were thrown away as

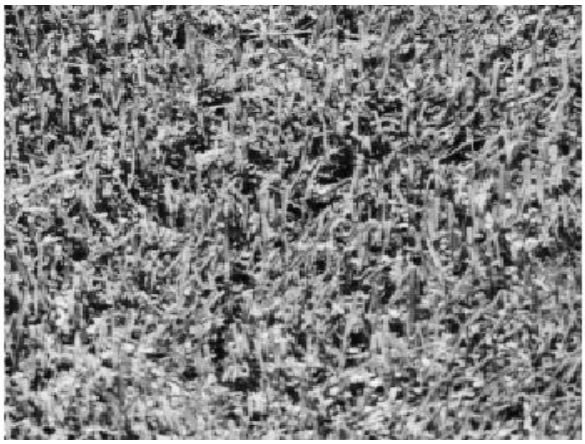


Figure 3.15: Grass image.

burn-in. We fitted the Ising model with first order and second order neighbourhood structure respectively. The results are presented in Table 3.3.

	μ_1	μ_2	σ_1	σ_2	β
PL-(F)	0.251 (0.0022)	0.609 (0.0015)	0.013 (0.00026)	0.019 (0.00027)	1.364 (0.017)
RCoDA-(F)	0.265 (0.0028)	0.620 (0.0017)	0.014 (0.00037)	0.018 (0.00029)	1.280 (0.022)
TDI-(F)	0.302 (0.0015)	0.650 (0.0010)	0.017 (0.00022)	0.013 (0.00014)	0.841 (0.0033)
PL-(S)	0.236 (0.0022)	0.599 (0.0015)	0.011 (0.00027)	0.021 (0.00029)	0.600 (0.0066)
RCoDA-C-(S)	0.252 (0.0025)	0.611 (0.0016)	0.013 (0.00029)	0.019 (0.00030)	0.567 (0.0080)
TDI-(S)	0.303 (0.0015)	0.649 (0.0010)	0.017 (0.00021)	0.013 (0.00016)	0.373 (0.0013)

Table 3.3: Posterior mean and standard deviation (in brackets) of grass data using PL, RCoDA and TDI respectively. (F) denotes first order neighbourhood structure. (S) denotes second order neighbourhood structure.

Table 3.3 presents the posterior mean and standard deviation of the two-component Gaussian spatial mixture model using TDI and RCoDA (only RCoDA-C was implemented for the second order neighbourhood) and PL. For both neighbourhood structures, the estimates for β were considerably different between the three methods, although the component mixture parameters were fairly similar. In both cases,

	PL-(F)	RCoDA-(F)	TDI-(F)	PL-(S)	RCoDA-(S)	TDI-(S)
95%	99.35	99.54	98.97	99.41	99.35	99.03
90%	96.77	97.24	96.11	97.16	97.08	96.18
80%	87.04	88.11	88.88	87.67	88.10	89.03

Table 3.4: Percentages of observed pixels which fall within the 95%, 90% and 80% of the posterior predictive distributions.

PL gave the largest estimate for β, followed by RCoDA and TDI always produced smaller estimates for β. Since for simulated data, where we know that the data comes from the Potts model, the results produced by the three methods were very similar, this suggests that the grass image may not closely follow a Potts model. However, since we do not know the truth, the effect of the three different methods becomes difficult to evaluate.

In order to assess the estimation from the three different approaches, we consider the use of posterior predictive distributions. For each posterior sample, we can simulate an image dataset, consequently, for each pixel, we can compare the observed value of that pixel with the posterior predictive distribution for that pixel. Table 3.4 shows the percentage of observed pixels which fall within a 95%, 90% and 80% of the posterior predictive distributions. We can see that here the three methods are quite similar, RCoDA having the higher proportions in most cases, indicative of a slightly better performance.

This example illustrates that for real datasets, the effect of possible model misspecification has different implications depending on the computational methods used. While a posterior predictive check appears to suggest all the methods are performing similarly, the posterior parameter estimates are quite different. This illustrates the importance of model checking and validation in this type of applications.

3.6 Discussions

In this chapter I have proposed a new method of estimating the q-state Potts model without having to compute the usual intractable normalizing constant. My method recursively partitions a regular lattice into a conditionally independent sublattice and approximates the other by another Potts model with a weaker dependence. By doing so, the method effectively avoids the computation of the troublesome normalizing constant. I presented the method in terms of first and second order neighbourhood structure on a 2D lattice. More complex lattices and dependence structures may be possible but would be much more difficult to work with. The method was demonstrated for $q = 2$ and $q = 3$ in this chapter, but can be applied to any q.

The proposed method is computationally efficient, the computational complexity is of the same order of magnitude as that of PL. Table 3.5 shows computation time for different algorithms. All methods are implemented for first order neighbouhood structure problem. Both RCoDA, PL and TDI are implemented in Matlab. The detailed information of CPU of our machine is: Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz. While RDA is implemented in C language.

size	32×32	64×64	128×128	256×256
RCoDA	0.014	0.018	0.034	0.075
PL	0.001	0.003	0.010	0.036
TDI	0.006	0.006	0.006	0.006
RDA	0.015	0.029	-	-

Table 3.5: Computation time in seconds per iteration of MCMC. RDA is not implemented for large lattice.

Regarding computational time, RCoDA is hybrid of PL and TDI. RCoDA generates a residual small lattice after several decompositions. The residual lattice can be coped with in two different ways, as shown in Section 3.2. Residual lattice is either treated as independent Potts model, or calculated by TDI. In the latter case, RCoDA's computational time includes calculation time of TDI.

Once the precomputation table is finalised, computational time for TDI is fixed. Each iteration, TDI needs to calculate corresponding normalizing constant given the fixed spline. It weakens TDI's computational advantage that precomputation table will take long time. though several choices can be used to calculate the final lattice $\mathbf{z^{2T}}$, TDI is adopted in this paper. Therefore, complexity of RCoDA likelihood involves PL calculation and TDI calculation. Moreover, the extra parameter α will double computation time for each iteration. This is indicated in Table 3.5 that RCoDA takes approximately double time of the summation of PL and TDI. Although PL is the fastest algorithm among the four, it has been shown that variance estimation in PL is not reliable. TDI has to implement off-line precalculation about normalizing constant before the inference. And the off-line calculation takes significant time. It is shown in Table 3.5 that RDA can' t handle Potts model on large lattice. Therefore, RCoDA is the only algorithm which is proper to use in large lattice. As size of Potts model increases, the time consumed by RCoDA is approaching to two times of PL. Because for very large lattice, the residual small lattice can be considered as independent.

We have shown through our simulation studies that RCoDA obtains the correct empirical coverage probabilities, whereas PL does not always do so. We have shown that for the first order neighbourhood, the estimation in terms of root mean squared error is competitive with several existing methods for different values of q and lattice sizes. For the second order neighbourhood structure, RCoDA produces better results when the size of the lattice is large.

Summarily, the advantage of RCoDA over other methods such as RDA and TDI, is its scalability in q and lattice size. The significant difference between RCoDA and these two methods is whether normalizing constant is calculated directly. Both RDA and TDI aims to approximate normalizing constant $\mathcal{C}(\beta)$, while RCoDA view $\mathbf{z}$ as joint distribution of $z_i, i = 1, 2, \cdots, n$. RCoDA avoids normalizing constant $\mathcal{C}(\beta)$ by

approximation joint distribution directly.

There are some drawbacks of RCoDA as well. First, RCoDA takes longest time in moderate size of Potts model, where RDA is still available. Even though TDI may takes longer time than RCoDA when precalculation is included, RCoDA has no advantage in terms of computational time. Second, RCoDA is troublesome with irregular lattice. For irregular lattice, shape of Potts model on each sublattice changes after each decomposition. Since β denotes overall spatial correlation in Potts model, different shapes lead to different meanings of β. In other words, for irregular lattice, meaning of β is not consistent over all decomposition. The level of consistency of β during decomposition determines how appropriate to use RCoDA. I denote the consistency as 100% For regualr lattices, since all the sublattices are exactly of identical shape with the original lattice. The higher consistency is, the more appropriate to use RCoDA. According the above rule, the lattices whose irregularity only happens at the boundary are more appropriate to apply RCoDA than the ones whose irregularity happens at both the boundary and internal area. It is complicated to measure consistency regarding shapes during decomposition. Given the complexity, it is not worthy to measure the consistency in terms of computational efficiency. Even though consistency is measured, it can't be included in current RCoDA method. But this does not mean that RCoDA is not proper to all the irregular lattices. For the lattices where only on the boundary is irregular, the consistency is very high. RCoDA is still one of competitive methods to deal with normalizing constant issue in practice.

Only the standard form of Potts model is discussed in this chapter. Generalization of the proposed RCoDA methods to variants of Potts models, such as the Potts model with an external field is of interest in my future work. I will develop another method in next chapter which will alleviate the drawbacks of RCoDA mentioned before.

Chapter 4

Monte Carlo method for partial conditional distribution in Markov random field

In this Chapter, another method was proposed for solving the same problem discussed in Chapter 3. The density function of the Potts model is decomposed into many conditional distributions. They are substituted by corresponding sufficient summary statistics. The distributions of the summary statistics are approximated by Monte Carlo simulations. Therefore, the density function of the Potts model is calculated without calculation of the normalizing constant.

4.1 Monte Carlo method

4.1.1 Conditional decomposition

The regular lattice $\mathbf{z}$ can be vectorized by row or by column. Here, we choose to vectorize it by column. For $n \times n$ MRF, $\mathbf{z}$ is transformed into a vector $z_v = (z_1, z_2, \cdots, z_{n^2-1}, z_{n^2})'$. We rewrite the likelihood function of the Potts model accord-

ing to the following conditional decomposition

$$\pi(\mathbf{z}|\beta) = \pi(z_1|\beta)\prod_{i=2}^{n^2}\pi(z_i|z_{1:i-1},\beta) \approx \prod_{i=2}^{n^2}\pi(z_i|(z_{1:i-1}\cap\partial i),\beta), \tag{4.1}$$

where $z_{1:i-1}$ denotes the set $\{z_1, z_2, \cdots, z_{i-1}\}$, ∂i denotes the neighbourhood of i. Let S_i denote the intersection, i.e. $S_i \equiv (z_{1:i-1}\cap\partial i)$. As n increases, the contribution of z_1 is asymptotically irrelevant. Therefore, we can ignore $\pi(z_1|\beta)$ in the middle step of Equation 4.1. Such setting can be also found in White et al. (2015). Although $\pi(z_1|\beta)$ has very little impact on the final result of computation, the computation of$\pi(z_1|\beta)$ itself is not easy. Therefore, ignoring $\pi(z_1|\beta)$ will alleviate the burden of the whole computation of Equation 4.1. Besag (1974) suggested the unilateral scheme approximation to approximate $\pi(z_i|z_{1:i-1},\beta)$. According to unilateral approximation, any term $\pi(z_i|z_{1:i-1},\beta)$ in Equation 4.1 can be approximated as $\pi(z_i|(z_{1:i-1}\cap\partial i),\beta),\ i = 2,3,\cdots,n^2$. All the pixels in the Potts model are homogeneous (Feng et al. (2012)). Therefore, the marginal distribution of z_i does not vary with the location of z_i. Although unilateral approximation is simple to implement, its drawback is obvious as it drops some dependence during the approximation.

It is notable that $\pi(z_i|z_{1:i-1},\beta) = \pi(z_i|(z_{1:i-1}\cap\partial i),\beta)$ does not hold. We can only use $\pi(z_i|(z_{1:i-1}\cap\partial i),\beta)$ to approximate $\pi(z_i|z_{1:i-1},\beta)$. According to the Markov property of Potts model, the conditional distribution of z_i only depends on its full neighbourhood given all the other z_js. In other words, when the conditional items include all the neighbours of z_i, the other z_js can be ignored. Apparently, $z_{1:i-1}$ does not include the full neighbourhood of z_i. Therefore, the equation $\pi(z_i|z_{1:i-1},\beta) = \pi(z_i|(z_{1:i-1}\cap\partial i),\beta)$ does not hold.

Here is the proof why $\pi(z_i|z_{1:i-1},\beta) = \pi(z_i|(z_{1:i-1}\cap\partial i),\beta)$ does not hold. It is straight-

forward to see that,

$$\begin{aligned}\pi(z_i|z_{1:i-1},\beta) &= \int \pi(z_i|z_{1:i-1},z_{i+1},z_{i+n},\beta)\pi(z_{i+1},z_{i+n}|z_{1:i-1},\beta)dz_{i+1}dz_{i+n}\\ &= \int \pi(z_i|z_{\partial i},\beta)\pi(z_{i+1},z_{i+n}|z_{1:i-1},\beta)dz_{i+1}dz_{i+n},\end{aligned}$$

and

$$\begin{aligned}\pi(z_i|(z_{1:i-1}\cap\partial i),\beta) &= \int \pi(z_i|(z_{1:i-1}\cap\partial i),z_{i+1},z_{i+n},\beta)\pi(z_{i+1},z_{i+n}|(z_{1:i-1}\cap\partial i),\beta)dz_{i+1}dz_{i+n}\\ &= \int \pi(z_i|z_{\partial i},\beta)\pi(z_{i+1},z_{i+n}|(z_{1:i-1}\cap\partial i),\beta)dz_{i+1}dz_{i+n}.\end{aligned}$$

Where z_{i+1}, z_{i+n} denote the other neighbours of z_i which are not included in the set $z_{1:i-1}$. Obviously, $\pi(z_{i+1},z_{i+n}|(z_{1:i-1}\cap\partial i),\beta) \neq \pi(z_{i+1},z_{i+n}|z_{1:i-1},\beta)$. Therefore, $\pi(z_i|z_{1:i-1},\beta) \neq \pi(z_i|(z_{1:i-1}\cap\partial i),\beta)$.

If $(z_{1:i-1}\cap\partial i)=\partial i$, terms at the right side of Equation 4.1 becomes $\pi(z_i|\partial i)$ which can be calculated without much trouble according to the following full conditional distribution,

$$\pi(z_i=k|\partial i)=\frac{\exp\{\beta\sum_{j\in\partial i}I(z_j=k)\}}{\sum_{l=1}^{q}\exp\{\beta\sum_{j\in\partial i}I(z_j=l)\}},\quad k=1,2,\cdots,q. \tag{4.2}$$

However, it happens more often that $(z_{1:i-1}\cap\partial i)$ is a proper subset of ∂i. In such scenarios, $\pi(z_i|(z_{1:i-1}\cap\partial i),\beta)$ is named as partial conditional distribution (PCD). Although this PCD is distributed as a multinomial distribution, the analytic form with respect to β is unknown. Therefore, the analytic form of Equation 4.1 remains unknown. For each term in the right side of Equation 4.1, Monte Carlo method is adopted to investigate the PCDs, which is introduced in Section 4.1.2.

4.1.2 Monte Carlo approximation of PCD

As an example, the first order neighbourhood structure in the Ising model is utilized to illustrate our Monte Carlo approximation in this Section. Besag (1974) suggested the first order neighbourhood structure which indicated that pixels which are on the left, right, back and front are deemed as the neighbours of the central pixel. The structure can be visualized in the left panel of Figure 4.1 where $\{z_A, z_B, z_D, z_E\}$ are neighbours of z_C.

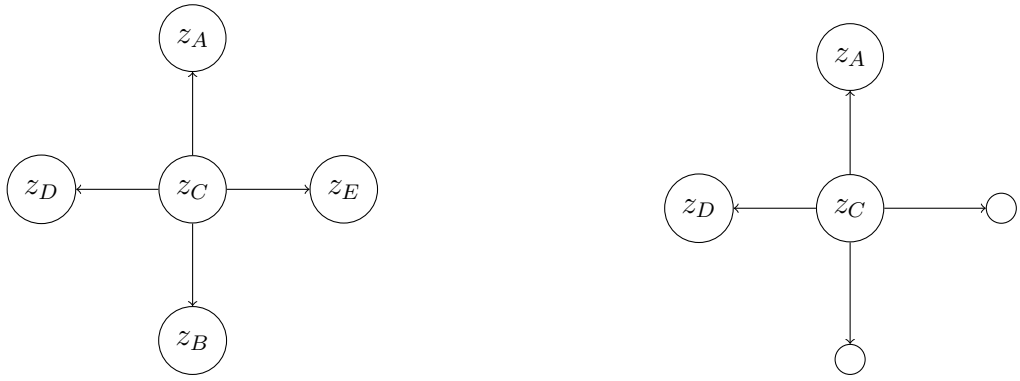


Figure 4.1: Left panel: the first order neighbourhood dependence structure. Right panel: partial conditional dependence of z_C given z_A and z_D.

A typical example of set S_i in the right side of Equation 4.1 satisfies the following, $S_i \subset \partial i$ and $S_i \neq \emptyset$. Due to most of the S_i's having two components, we suppress the subscript in S_i to S to make the notation simpler. Monte Carlo simulation can be implemented regarding PCD directly, as it is known that $\pi(z_C|z_A, z_D, \beta)$ is distributed as Bernoulli distribution. Let $\hat{\pi}(z_i|(z_{1:i-1} \cap \partial i), \beta)$ denote the Monte Carlo approximation of $\pi(z_i|(z_{1:i-1} \cap \partial i), \beta)$. Thus, Equation 4.1 can be approximated as the following,

$$\pi(\mathbf{z}|\beta) \approx \prod_{i=2}^{n^2} \hat{\pi}(z_i|(z_{1:i-1} \cap \partial i), \beta). \tag{4.3}$$

However, Monte Carlo simulation regarding summary statistics is adopted in this approximation. The reasons why Equation 4.3 is not utilized is discussed in Section 4.3.1. Also the detailed description of the Monte Carlo simulation is outlined in the

following pages.

A new random variable P^S is defined for pixel i, which denotes the number of pairs of z_i given S. The random variable P^S is a sufficient summary statistic of this PCD as P^S provides all necessary information required for inference of β. The possible realization of P^S is in $\{0, 1, 2\}$, as there are two elements in S. Even though the analytic form of P^S is unknown, it is certain that the P^S is distributed as a multinomial distribution. It is straightforward to see that P^S is sufficient statistic of $\pi(z_i|S, \beta)$. In the rest of this Section, Monte Carlo simulation regarding the distribution of P^S is discussed.

As there are two components in S, $\pi(z_C|z_A, z_D, \beta)$ is used to denote the typical PCD of interest which corresponds to the typical term in the right side of Equation 4.1. The right panel of Figure 4.1 shows geographical relationship between z_C and its two conditional items z_A and z_D. We term the structure in the right panel of Figure 4.1 as unit structure. Conditional type (CT) is defined prior to description of the Monte Carlo approximation method. CT denotes possible combination of conditional items z_A and z_D. The combination of z_A and z_D determines different multinomial distribution. In other words, the number of CT's is equal to the number of multinomial distributions to approximate. Combinatorially it is easy to see that there are 2^2 combinations of z_A and z_D.

We take advantage of interchangeability of the Potts model as well as the unit structure. Interchangeability means that combination is irrelevant with positions of conditional items in unit structure. In other words, z_A and z_D can switch their positions and their combination remains unchanged. In addition, the values of conditional items have no real meaning and relabelling them makes no difference to conditional type. Given these properties of the unit structure, the number of CT's is reduced from 2^2 to 2. The first CT is $z_A = z_D$, while the second CT is $z_A \neq z_D$.

The goal of Monte Carlo approximation is to approximate the distribution of

P^S under different CT. Since there are two elements in S, the sample space of P^S is $\{0, 1, 2\}$. More specifically, we need to approximate $\pi(P^S = 0), \pi(P^S = 1)$ and $\pi(P^S = 2)$ for each CT using Monte Carlo method. Let CT_1 denote the the first CT $z_A = z_D$ and CT_2 denote the second one $z_A \neq z_D$.

In context of the $n \times n$ Potts model, our Monte Carlo process is described as follows:

1. Given β, generate one realization of the Potts model $\mathbf{z}$ using the Gibbs sampler as in Feng (2008).

2. For each pixel z_i, initially identify related unit structure centered at z_i. Subsequently, determine the CT of each unit structure according to the relationship between conditional items. Then count the number of pairs in the unit structure and record it accordingly. In one sweep of this step, n^2 unit structures are identified and n^2 numbers are recorded.

3. Repeat the previous two steps until the Monte Carlo sample size is large enough. Subsequently, enough counts of pairs are obtained for each CT.

4. For each CT, normalize the counts to obtain corresponding frequencies in multinomial distribution.

Following the above procedure, we can obtain approximated distributions of P^S given β. The same procedure can be implemented over a grid of β. These approximations are referred to as look-up table of the method. If the grid of β is dense enough, good approximations of the multinomial distributions with respect to β would be achieved.

Once the multinomial distributions of the P^S are approximated, the likelihood function of Equation 4.1 can be given as

$$\pi(\mathbf{z}|\beta) \propto \prod_{i=2}^{n^2} \hat{\pi}(P^{S_i}|\beta), \tag{4.4}$$

where $\hat{\pi}(P^{S_i}|\beta)$ denotes the approximated multinomial distribution of P^{S_i}. We term the method using Equation 4.4 as Monte Carlo approximation of partial conditional distribution (MCAPCD).

The proposed method has a number of advantages. First of all, approximation of multinomial distributions can be achieved by drawing samples from a Potts model of arbitrary size due to the independence between the multinomial distributions and the sizes of the Potts models. MCAPCD can use samples from small Potts models to infer about large Potts models. This alleviates burden in making inference about large Potts models. The statement on the independence between the multinomial distributions and the sizes of the Potts models will not be proved here. Instead, I will show part of our simulation results to justify the independence. The distribution of Ising model under CT_1 is taken as an example. The detailed procedures of the simulation are same as the steps stated in the Monte Carlo procedures above. The estimated multinomial distributions for three different sized Potts models are shown in Table 4.1. We can see that the difference between different sized models is trivial. In practice, it is much more difficult to generate a large Potts model than a small one in terms of computation. More importantly, unlike other Monte Carlo methods, such as TDI, it is unnecessary to sample for all kinds of Potts models of different sizes, since the approximation of multinomial distribution does not change with the sizes of Potts models. Once the approximated multinomial distributions are obtained, they can be reused in inferring about different sized Potts models. Last, but not least, MCAPCD saves a great deal of time on Monte Carlo sampling as one realization of $\mathbf{z}$ produces n^2 samples in MCAPCD. In contrast, most other Monte Carlo methods count one realization as one sample. In this sense, MCAPCD is approximately n^2 times faster than other methods, such as TDI (Green and Richardson 2002).

β	32×32			64×64			128×128		
0.050	0.474	0.000	0.526	0.475	0.000	0.525	0.475	0.000	0.525
0.100	0.451	0.000	0.549	0.450	0.000	0.550	0.450	0.000	0.550
0.150	0.425	0.000	0.575	0.425	0.000	0.575	0.425	0.000	0.575
0.200	0.398	0.000	0.602	0.400	0.000	0.600	0.400	0.000	0.600
0.250	0.376	0.000	0.624	0.375	0.000	0.625	0.375	0.000	0.625
0.300	0.352	0.000	0.648	0.351	0.000	0.649	0.351	0.000	0.649
0.350	0.325	0.000	0.675	0.326	0.000	0.674	0.326	0.000	0.674
0.400	0.302	0.000	0.698	0.302	0.000	0.698	0.303	0.000	0.697
0.450	0.281	0.000	0.719	0.279	0.000	0.721	0.279	0.000	0.721
0.500	0.254	0.000	0.746	0.255	0.000	0.745	0.255	0.000	0.745
0.550	0.236	0.000	0.764	0.231	0.000	0.769	0.231	0.000	0.769
0.600	0.206	0.000	0.794	0.208	0.000	0.792	0.208	0.000	0.792
0.650	0.184	0.000	0.816	0.185	0.000	0.815	0.185	0.000	0.815
0.700	0.162	0.000	0.838	0.162	0.000	0.838	0.162	0.000	0.838
0.750	0.139	0.000	0.861	0.139	0.000	0.861	0.139	0.000	0.861
0.800	0.116	0.000	0.884	0.115	0.000	0.885	0.115	0.000	0.885
0.850	0.085	0.000	0.915	0.089	0.000	0.911	0.089	0.000	0.911
0.900	0.056	0.000	0.944	0.061	0.000	0.939	0.062	0.000	0.938
0.950	0.037	0.000	0.963	0.044	0.000	0.956	0.046	0.000	0.954
1.000	0.030	0.000	0.970	0.033	0.000	0.967	0.035	0.000	0.965

Table 4.1: The multinomial distribution of Ising model under CT_1 for different sized model, 32×32 and 64×64.

4.1.3 Generalization to higher order Potts model

MCAPCD can be generalized to higher order Potts models. For example, the second order neighbourhood structure proposed in Besag (1974) is considered in this Section. For higher order Potts models, more complicated unit structures apply. The second order structure is presented in Figure 4.2 where the pixels except z_C are neighbours of z_C. Figure 4.3 demonstrates a typical unit structure for the second order Potts model.

The difference between the second order and the first order is that there are four rather than two conditional items in each unit structure. It is obvious that more conditional items result in more CT's, leading us to approximate more multinomial distributions. However, it is guaranteed that the number of distributions to approx-

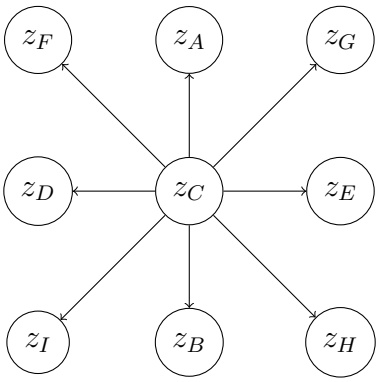


Figure 4.2: Second oder neighbourhood structure. The pixels other than z_C are neighbours of z_C.

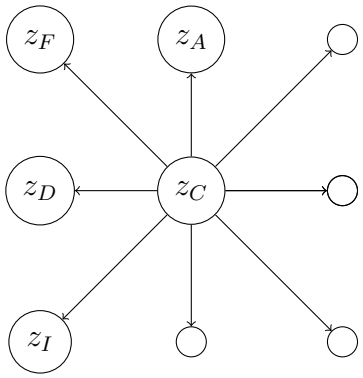


Figure 4.3: Partial conditional distribution.

Figure 4.4: Second order Potts model.

imate is less than $q^{(\#S)}$, which is the number of distributions to approximate if summary statistics are not utilized. Although there are more CT's to approximate for the second order neighbourhood structure, the main procedure remains the same as the first order neighbourhood structure. As a consequence, the details of MCAPCD for the second order neighbourhood structure will be omitted.

It is noticeable that in both the first order and the second order neighbourhood structures, most typical terms are illustrated in details. Despite that, there are special pixels which are different in each scenario. For example in the first order neighbourhood structure, only one neighbour is given in the unit structure for the pixels which are located on the left or upper boundaries of the lattice. In this case, the conditional type should be determined separately. Thus, the multinomial distribution should be approximated separately. Nonetheless, the main process is same as the above.

4.2 Simulation study

In this Section, the simulation results from different methods are demonstrated. Various scenarios are included: different q and different lattice size. The look-up tables

required in TDI and MCAPCD are calculated beforehand.

4.2.1 First order neighbourhood lattice

It is well known that the Potts models exhibit the so-called "phase transition", where for $\beta > \beta_{crit}$, the Potts models will transit from a disordered to an ordered pattern or phase. This means that the sites will eventually all be in the same state as β increases. For a general q-state model, the precise value of the critical value is difficult to determine. For $q = 2, 3, 4$, Potts (1952) developed the exact solution of critical points $\beta_{crit} = \log(1 + \sqrt{q})$ which is about 0.88 for the Ising models ($q = 2$). Here we will restrict our analyses to β below the critical values, and consider the set of values $0.1, 0.2, \ldots, 0.8$ for β .

We have also compared performance of methods for the Potts models with $q = 3$, omitting the method RDA, which cannot be easily extended to the case for higher q.

As shown in Table 4.2, three different sizes are demonstrated, including 32×32, 128×128 and 256×256. Root mean squared error (RMSE) is selected to indicate the goodness of estimation. Overall, MCAPCD outperforms other methods with respect to RMSE. More detailed, the RMSE behaves quite differently in terms of different factors. In terms of spatial correlation strength β, MCAPCD and TDI perform better as β increases. Whereas other methods have the opposite behavior. In terms of the sizes of Potts models, all methods improve their performances as the sizes of Potts models get larger, since larger Potts models lead to larger sample sizes. The performance seems to have no correlation with q. In other words, these methods can be applied to Potts model with any q.

4.2.2 Second order neighbourhood lattice

The phase transition also exists in the second order neighbourhood Potts models. The critical values for the second order Potts models cannot be obtained in a closed

β			0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.800
32^2	RCoDA	q=2	0.039	0.047	0.048	0.053	0.053	0.057	0.057	0.051
		q=3	0.039	0.047	0.051	0.051	0.053	0.049	0.049	0.046
	PL	q=2	0.043	0.046	0.044	0.049	0.048	0.046	0.046	0.053
		q=3	0.044	0.046	0.047	0.049	0.051	0.044	0.042	0.047
	TDI	q=2	0.040	0.042	0.042	0.043	0.038	0.037	0.036	0.032
		q=3	0.040	0.044	0.045	0.045	0.045	0.039	0.034	0.034
	MCAPCD	q=2	0.037	0.046	0.044	0.043	0.039	0.037	0.031	0.028
		q=3	0.039	0.045	0.046	0.043	0.039	0.038	0.032	0.034
128^2	RCoDA	q=2	0.011	0.012	0.011	0.014	0.012	0.014	0.016	0.017
		q=3	0.011	0.012	0.011	0.011	0.011	0.012	0.012	0.013
	PL	q=2	0.011	0.011	0.011	0.012	0.011	0.011	0.012	0.012
		q=3	0.011	0.012	0.011	0.011	0.011	0.010	0.011	0.011
	TDI	q=2	0.011	0.011	0.010	0.010	0.009	0.009	0.008	0.007
		q=3	0.011	0.011	0.010	0.010	0.010	0.009	0.008	0.008
	MCAPCD	q=2	0.011	0.012	0.011	0.011	0.009	0.010	0.009	0.006
		q=3	0.012	0.011	0.012	0.011	0.009	0.009	0.010	0.008
256^2	RCoDA	q=2	0.006	0.005	0.006	0.006	0.006	0.008	0.009	0.013
		q=3	0.006	0.006	0.005	0.007	0.006	0.006	0.007	0.007
	PL	q=2	0.006	0.005	0.006	0.006	0.005	0.006	0.006	0.006
		q=3	0.006	0.005	0.006	0.006	0.005	0.006	0.006	0.006
	TDI	q=2	0.005	0.005	0.006	0.005	0.005	0.004	0.004	0.004
		q=3	0.006	0.006	0.005	0.005	0.005	0.005	0.004	0.004
	MCAPCD	q=2	0.007	0.006	0.005	0.005	0.005	0.005	0.005	0.004
		q=3	0.006	0.006	0.006	0.006	0.005	0.005	0.006	0.004

Table 4.2: Root mean squared error of β for a first order neighbourhood dependence. Based on 200 simulated data sets for each 32×32, 128×128 and 256×256 lattices. $q = 2$ and $q = 3$.

form. Chapter 3 concluded that the critical values for the second order Potts models should be smaller than 0.4. Therefore, all the simulations were implemented for the Potts models with $\beta < 0.4$. For more details about critical values of the second order Potts models, see Chapter 3.

Table 4.3 shows RMSE for all the methods with $q = 2$ and $q = 3$. Overall, MCAPCD outperforms other methods with respect to RMSE. As spatial correlation becomes stronger, MCAPCD performs better. While other methods obtain larger RMSE. In terms of size of Potts model, all the methods can improve their performance as size increases. But q does not have much impact on the performance of these methods. In summary, MCAPCD gets much smaller RMSE under various sce-

β		0.1		0.2		0.3	
		q=2	q=3	q=2	q=3	q=2	q=3
32^2	RCoDA	0.039	0.039	0.047	0.047	0.048	0.051
	PL	0.043	0.044	0.046	0.046	0.044	0.047
	TDI	0.040	0.040	0.042	0.044	0.042	0.045
	MCAPCD	0.031	0.029	0.031	0.026	0.020	0.021
128^2	RCoDA	0.011	0.011	0.012	0.012	0.011	0.011
	PL	0.011	0.011	0.011	0.012	0.011	0.011
	TDI	0.011	0.011	0.011	0.011	0.010	0.010
	MCAPCD	0.009	0.008	0.007	0.006	0.008	0.005
256^2	RCoDA	0.006	0.006	0.005	0.006	0.006	0.005
	PL	0.006	0.006	0.005	0.005	0.006	0.006
	TDI	0.005	0.006	0.005	0.006	0.006	0.005
	MCAPCD	0.004	0.004	0.005	0.003	0.008	0.002

Table 4.3: Root mean squared error of β for a second order neighbourhood dependence. Based on 200 simulated data sets for each 32×32, 128×128 and 256×256 lattices. $q = 2$ and $q = 3$ are included.

narios.

4.2.3 Coverage probability

In order to investigate the validation of estimated variance, we calculate the coverage probabilities for each method. The coverage probability is calculated as follows. Given a fixed β, generate 200 replications of the Potts model. Draw MCMC samples from posterior distribution of $\pi(\beta|Z)$, and calculate corresponding upper and lower quantiles. Count the number where the posterior interval covers the true β.

Figure 4.5 demonstrates the 95% coverage probabilities of the Ising models. All methods except PL are deviated from 95%. This indicates all methods except PL can obtain correct variance.

4.2.4 Computation time

The four algorithms RCoDA, PL, MCAPCD and TDI were implemented in Matlab. The detailed information of the CPU of our machine is as follow: Intel(R) Core(TM)

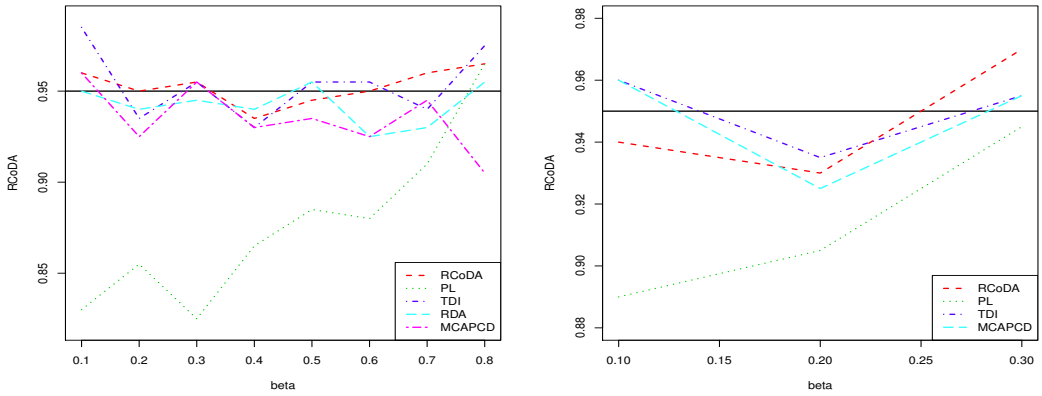


Figure 4.5: 95% empirical coverage probabilities for the 32 $\times$ 32 lattice under both first order (left) and second order neighbourhood (right).

i7-3770 CPU @ 3.40GHz. RDA was implemented in C language since the code was kindly provided by the authors (slightly modified for out purposes). Table 4.4 shows computation time for different algorithms. All methods were implemented for the first order neighbourhood structure Ising model.

size	32×32	64×64	128×128	256×256
RCoDA	0.014	0.018	0.034	0.075
PL	0.001	0.003	0.010	0.036
TDI	0.006	0.006	0.006	0.006
RDA	0.015	0.029	-	-
MCAPCD	0.00006	0.00006	0.00007	0.00007

Table 4.4: Computation time in seconds per iteration of MCMC. RDA is not implemented for large lattice.

There are two methods which require look-up tables to be pre-calculated: TDI and MCAPCD. For the moment, computational time for look-up tables are ignored. Suppose that all the prerequisites are well prepared, computational time for TDI and MCAPCD are fixed. Each iteration, TDI needs to calculate the corresponding normalizing constant given the fixed spline. While MCAPCD needs to calculate the likelihood using Monte Carlo approximations for each term. Approximately, these calculation times are constant with respect to the sizes of the Potts models. Among

all the above methods, RDA is the most inflexible method with respect to the sizes of the Potts models.

As TDI is adopted to calculate the final lattice $\mathbf{z^{2T}}$ in RCoDA. Therefore, the complexity of RCoDA involves the complexities of both PL calculation and TDI calculation of $\mathbf{z^{2T}}$, that is shown in Equation 3.5. Moreover, the extra parameter α will double the computational time for each iteration. This is indicated in Table 4.4 where RCoDA takes approximately the double time of PL and TDI combined.

Given that generation of one Potts model takes the same times for both TDI and MCAPCD, look-up table of MCAPCD takes much less time than TDI. As we mentioned before, one $n \times n$ Potts model sample is considered n^2 samples in MCAPCD, whereas it is only one sample in TDI. In this sense, MCAPCD is n^2 times more efficient than TDI. Moreover, once the look-up tables were generated, they can be applied to infer about any size of the Potts model in MCAPCD. Whereas TDI has to generate the look-up tables for different sizes of the Potts models. Hence, MCAPCD outperforms TDI overall and thus all other methods in terms of computational time.

In the above Section, we compared computational times between different algorithms. It was unfair to ignore pre-calculation time in TDI and MCAPCD. As we just mentioned, TDI needs to calculate the look-up tables when the size of the Potts model changes. Compared to MCAPCD, it is less efficient. Therefore, we focus on the pre-calculation of MCAPCD. There are two problems to resolve prior to the discussion of pre-calculation time of MCAPCD. Firstly, how dense of a grid of β is dense enough? Secondly, what sample size is large enough to approximate each PCD? In this Chapter, the step size of β was set to be 0.001 when the look-up tables were calculated. To address the first, we think that is dense enough to model the true relationship between β and the PCD's. For the second, a small size (32×32) of Potts model was repeated for 500 times to generate related samples which is 512000 samples in total. This means 512000 samples are generated to approximate PCD's

for each β. It takes 18.7 seconds to generate 512000 samples. There will be 1000 data points in the interval of $[0, 1]$, since the step size is 0.001. Therefore, the pre-calculation will take 5 hours. At the cost of 5 hours, MCAPCD offers super efficient calculation in the further inference according to the comparison in Table 4.4.

4.3 Discussion

4.3.1 Summary statistic

The number of pairs in the Potts model has been adopted as a sufficient summary statistic of the Potts model, for example Green and Richardson (2002), Ibáñez and Simó (2003), Grelaud et al. (2009). The P^S here measures the number of pairs between conditional items and the pixel of interest. Mathematically, $P^S = \sum_{i\sim j, j\in S} I(z_i = z_j)$, where S denotes the set of conditional items. The P^S is a sufficient summary statistic of PCD, since no further information is required to make inference about the parameters of interest given P^S. Therefore, P^S can be adopted to make inference about β in this Chapter.

In Section 4.1.2, Monte Carlo simulation was implemented regarding the summary statistic $\pi(P^S|\beta)$ rather than $\pi(z_i|S, \beta)$. Computational efficiency is the main reason why the summary statistic was utilized.

Monte Carlo approximation of $\pi(z_i|S, \beta)$ can be described as follows: first of all, given β, generate the Potts model $\mathbf{z}$. Secondly, identify all the unit structures which have identical conditional items with the PCD of interest and record the realization of z_i accordingly. Thirdly, normalize all the realizations of the PCD and then the multinomial distribution is approximated. In the above procedures, q probabilities need to be approximated for each PCD, where q is the number of possible realizations for each pixel.

Although both the PCD and P^S can be used to approximate corresponding prob-

abilities, P^S is more preferable in terms of computational efficiency. The computational complexity of approximation of P^S is dependent on the number of elements in the set S, which is denoted by $\#S$. For the first order neighbourhood structure Potts model, $\#S$ is no more than 2, regardless the value of q. Thus, the complexity of P^S is constant with respect to q. In contrast, the computational complexity of approximation of PCD is dependent on q, since each PCD is a q-term multinomial distribution. In other words, the computational complexity of approximation of PCD is linearly related to q. Given that $q \geqslant 2$, computation involving P^S is always more efficient than PCD under the first order neighbourhood structure.

More importantly, every possible combination of the CT's needs to be approximated in Monte Carlo approximation of PCD. In contrast, Monte Carlo approximation of P^S can take advantage of interchangeability of the Potts models, thereby reducing the number of distributions to approximate. This advantage becomes more significant under the higher order neighbourhood structures , such as those discussed in 4.1.3.

4.3.2 MCAPCD for irregular lattice

There are various types of boundary conditions that can be used to define the lattices of the Potts models. The common conditions include free boundary conditions, torus (periodic) boundary conditions, plus/minus boundary conditions, and mixed boundary conditions. Boundary conditions have been discussed in Fisher and Barber (1972), Cardy (1986), Hongler and Kytölä (2013). Among these boundary conditions, free boundary conditions and torus boundary conditions are the most widely used ones.

Literally, all the boundary conditions differ from each other regarding how to define the neighbours of the boundary pixels. In a regular lattice, each internal pixel has four neighbours. If we define that pixels on the boundary have reduced number

of neighbours, then free boundary conditions is adopted. As shown in the left panel of Figure 4.7, all the pixels on the edges have only two or three neighbours. If the lattice is considered as a torus, then torus boundary conditions are defined. Under torus boundary conditions, the last row of a lattice is deemed to be adjacent to the first row. An example of torus is demonstrated in Figure 4.6.

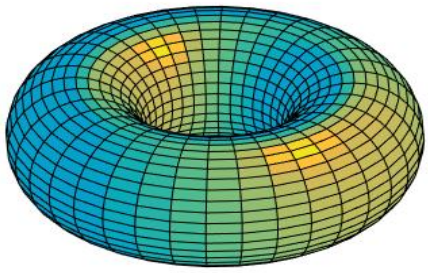


Figure 4.6: Torus.

Through out this Chapter, free boundary conditions were utilized. As we adopt free boundary conditions, an irregular lattice is considered as a generalization of a regular lattice. More specifically, in a regular lattice, the boundary area consists of four edges where the pixels have a reduced number of neighbours. Examples of a regular and irregular lattice are demonstrated in Figure 4.7. An irregular lattice can always be included in a regular lattice, as shown in the right panel of Figure 4.7. In irregular lattices, the boundary area is constituted of more irregular edges which may lead to more pixels having a reduced number of neighbours. For instance, pixel A in the right panel of Figure 4.7 has only two neighbours, whereas it would have four neighbours in the regular lattice.

In summary, the irregularity changes the boundary area, leading to extra irregular pixels which generate special conditional distributions in MCAPCD. The special

pixels have been discussed in the end of Section 4.1.3. The special pixels can be handled naturally in the algorithm of MCAPCD. Therefore, irregularity of lattices do not cause much trouble in the implementation of MCAPCD.

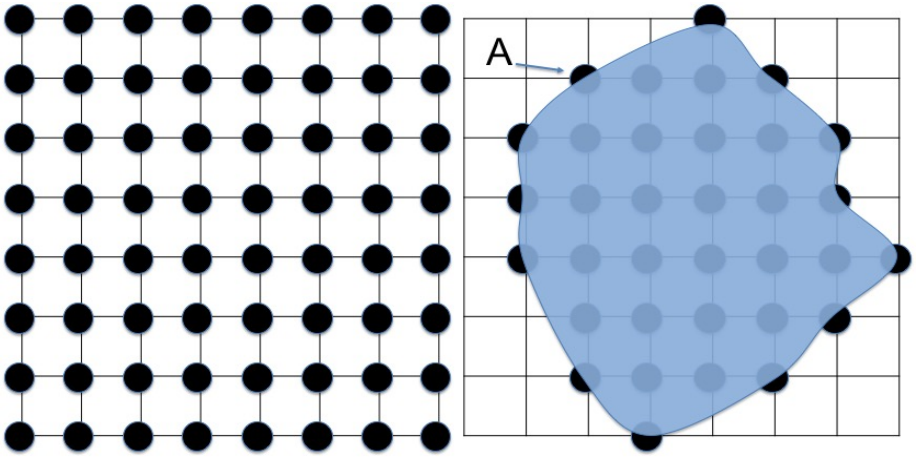


Figure 4.7: 8×8 lattice. Left panel: regular lattice. Right panel: irregular lattice.

4.3.3 Relationship with other methods

Our Monte Carlo methods are closely related to Approximate Bayesian Computation (ABC) methods and the synthetic likelihood method of Wood (2010).

Approximate Bayesian computation

ABC methods are likelihood-free techniques which require no calculation of specific likelihood functions. Thus, ABC is widely used for models whose likelihood functions are intractable or very expensive to calculate. It was firstly proposed in the context of biology and ecology (Beaumont, Zhang, and Balding 2002; Beaumont 2010). Then it has been applied in broader areas. For a review of ABC, see Marin et al. (2012).

In essence, ABC is a technique of information reduction. A summary statistic $\mathbf{S}$ which is employed for inference is defined for each dataset. The very basic algorithm of ABC is described below. Firstly, generate a value of θ of interest from its

prior distribution. Secondly, generate an artificial dataset x given the value of β. Thirdly, compare the distance between $\mathbf{S}(x)$ and $\mathbf{S}(y)$, where y denotes the observation. Given a preset tolerance ε, we accept θ, if $\| \mathbf{S}(x) - \mathbf{S}(y) \| < \varepsilon$ holds, where $\|\cdot\|$ denotes a suitable norm. Fourthly, repeat the above three steps for enough times. Usually, N is predefined, where N denotes the total number of accepted samples. The sampling procedure will continue to run until N samples are accepted. Finally, the posterior distribution of θ is approximated as $\pi(\theta|y) \approx \pi(\theta| \| \mathbf{S}(x) - \mathbf{S}(y) \| < \varepsilon)$. If $\varepsilon = 0$, ABC is referred as exact Bayesian computation (EBC).

ABC and MCAPCD have something in common. Both of these methods adopt summary statistics to reduce dimension. Dimension reduction through the summary statistic is essential for ABC, and the MCAPCD adopts the summary statistic to facilitate the whole computation. However, the summary statistics have different roles in the two methods. Compared to ABC which is a likelihood-free algorithm, MCAPCD has to approximate likelihood function. In MCAPCD, the summary statistic is adopted to substitute the terms in the original likelihood function. Therefore, the likelihood is calculated given the approximation of distributions of the summary statistic. In this sense, our proposed method is not a likelihood-free technique like ABC. This highlights the most significant difference between these two methods.

Synthetic likelihood

Wood (2010) proposed a synthetic likelihood method to infer about nonlinear dynamic systems. Synthetic likelihood is one of information induction techniques. This method aimed to solve dynamic systems where minor changes in noise cause drastic changes in the system trajectory.

Rather than computing the tolerance in ABC, the synthetic likelihood approximates the likelihood using Monte Carlo samples by assuming a distributed form for the samples.

Synthetic likelihood requires a summary statistic to represent the dataset. Given the parameters of interest, a large number of dataset are simulated. Then the summary statistics are calculated accordingly and are assumed to be distributed as multivariate Gaussian distribution. Given the above samples of the summary statistics, the mean and covariance matrix in the multivariate distribution are estimated. Synthetic likelihood then uses the density function of multivariate Gaussian distribution to infer about the parameters in the original likelihood. In this way, calculation of the original intractable likelihood is avoided.

The common feature between the synthetic likelihood and our proposed method is that both Monte Carlo simulation and summary statistic are involved in each methods. Parametric density functions of summary statistics are employed to determine the sampling of parameters of interest. The synthetic likelihood method assumes the summary statistics are distributed as multivariate Gaussian distribution. The assumption can be justified when the sample size approaches to infinity. In MCAPCD, the distribution of summary statistic is known as multinomial distribution.

4.4 Summary

MCAPCD was proposed to solve the normalizing constant problem in the Potts models by avoiding the calculation of the intractable normalizing constant. The method takes advantage of conditional decomposition of the original likelihood function. The likelihood function is transformed to be the product of many conditional density functions that are substituted by the corresponding summary statistic. The distribution of summary statistic is approximated through Monte Carlo simulation. By doing this, the calculation of intractable normalizing constants is avoided and the inference of β can be implemented.

MCAPCD was proposed on account of two considerations. Firstly, it is fast. It takes less time than PL without counting in the calculation of the look-up tables. Even though the pre-calculation is taken into account, it has advantages over the other methods in terms of computational efficiency as discussed in Section 4.2.4. Secondly, MCAPCD has advantage when dealing with irregular lattices. The irregularity of lattices only changes boundary area where the number of neighbours is reduced compared with the interior area. The new boundary area may add more burden on computation, but it will not affect the implementation of algorithm itself. Given these two advantages of MCAPCD, it can be applied in broader applications.

Chapter 5

Relabelling algorithms for mixture models with applications for large datasets

Bayesian inference of mixture models always encounter the problem of label switching. Even the normalizing constant problem was overcome by the methods in previous Chapters, label switching problem could be another issue. In this chapter, label switching problem is discussed. Many algorithms for this issue are reviewed. We propose a new method to particularly solve label switching in spatial mixture model.

5.1 Introduction

Mixture models have been used extensively, in areas such as nonparametric density estimation (Norets 2010) and model based clustering (Banfield and Raftery 1993, McLachlan and Basford 1988). Other applications include micro-array analysis (McLachlan et al. 2002), disease mapping (Green and Richardson 2002), finance analysis (Brigo and Mercurio 2002; Alexander 2004; Xu and Knight 2013), texture models

(Permuter et al. 2003; Sujaritha and Annadurai 2011), ecology (Ullah et al. 2015), image analysis (Brazey and Portier 2014), density estimation (Zhu 2016). These models provide a flexible way of modelling heterogeneous data. We are concerned with finite mixture distributions of K components with density given by

$$p(x_i|\phi) = \sum_{k=1}^{K} w_k f(x_i \mid \theta_k) \tag{5.1}$$

for some data $x_i \in \mathcal{R}^d, d \geq 1, i = 1, \ldots, n$, where $f(x_i|\theta_k)$ is the kth component density of the mixture, with parameters θ_k. For instance, $f(x_i|\theta_k)$ can be an univariate or a multivariate Normal distribution, where parameter vector θ_k represents the mean and variance/covariance of the Normal distribution. Finally, w_k is the weight of the kth component density, such that $\sum_{k=1}^{K} w_k = 1$. We will denote the entire q-dimensional set of parameters as $\phi = ((w_1, \theta_1), \ldots, (w_K, \theta_K))$. Comprehensive reviews of finite mixture models can be found in Titterington et al. (1985), McLachlan and Peel (2004), Marin et al. (2005), Frühwirth-Schnatter (2006) .

Bayesian analyses of finite mixture models typically involve the use of Markov chain Monte Carlo (MCMC) sampling from the posterior distribution, where label switching becomes an issue. This occurs as a result of the invariance of Equation (5.1) with respect to the reordering of the components such that

$$\sum_{k=1}^{K} w_k f(x_i \mid \theta_k) = \sum_{k=1}^{K} w_{\nu_k} f(x_i \mid \theta_{\nu_k}), \tag{5.2}$$

where $\{\nu_1, \ldots, \nu_K\}$ is an arbitrary permutation of $\{1, \ldots, K\}$. The total number of permutations is $K!$. If the priors of the parameters are the same or exchangeable, the posterior distribution will be invariant under the permutation. One can visualise the occurrence of label switching within an MCMC sampler. For instance, the parameters of the first component may move to the modal region of the second component as the Markov chain explores the state space, and vice versa. While the posterior

density remains invariant to the labelling, the correct ordering of the labels should have swapped the two sets of parameters.

Many methods have been developed to resolve the issue of identifiability in Bayesian inference. Jasra et al. (2005) provided a detailed and insightful review of developments on this topic up to around 2005. The simplest method is to impose an artificial identifiability constraint. For instance, Richardson and Green (1997) suggested ordering the location parameters of a univariate Normal mixture model, such that $\mu_1 < \ldots < \mu_K$, where μ_k corresponds to the mean parameter of the kth component. Imposing such identifiability constraints can also be seen as a modification of the prior distribution. The method is simple in terms of computational complexity and can also be implemented within the MCMC sampler. However, it was demonstrated in Jasra et al. (2005) and Celeux et al. (2000) that the method can fail to fully resolve the issue of identifiability in some cases. Additionally, as often occurs in more complex situations as the dimension of the parameter space increases, there may no longer be a natural ordering on the parameters. See Frühwirth-Schnatter (2011) for an example in the case of multivariate Normal mixtures.

Another class of relabelling algorithms, perhaps the best known algorithms in the literature to date, is based on decision theoretic arguments. Samples from MCMC output are post-processed according to some loss function criterion, see Stephens (1997a), Stephens (2000), Celeux (1998), Celeux et al. (2000), Hurn et al. (2003) and references therein. These methods work well, and are considered to be theoretically better justified by Jasra et al. (2005). However, they are computationally intensive. Thus for large datasets or high dimensions, they become impractical to use.

Finally, a different approach, based on probabilistic relabelling, can be found in the works of Sperrin et al. (2010) and Jasra (2005), which involves the calculation of the likelihood of the permutations $\{\nu_1, \ldots, \nu_K\}$. Sperrin et al. (2010) gave an EM-type algorithm for its estimation. Puolamaki and Kaski (2009) developed a relabelling

approach which requires the introduction of a discrete latent variable in the original probabilistic model. More recently, Yao and Lindsay (2009) proposed an algorithm based on the mode of the posterior and an ascent algorithm for each iteration of the MCMC sample. Yao and Li (2014) proposed a method which minimizes the class probabilities to a fixed reference label. Yao (2012) proposed to assign the probabilities for each possible labels by fitting a mixture model to the permutation symmetric posterior. Although many of these algorithms were demonstrated to work well, they do not scale up well for large data or high dimensions.

Many modern applications of mixture models involve increasingly large datasets, such as those in genetic studies, and in medical image analyses. In many of these problems, the number of mixture components K is typically small, $K < 10$. The number of observations N can be huge, in the order of millions, and the parameter space q can also be very large, in the order of hundreds (see for example Zhu et al. 2016). In this situation, we found that well established algorithms that have good theoretical properties such as Stephens (2000) and Celeux et al. (2000), as well as many of the more recent developments mentioned above, quickly becomes computationally infeasible. A number of lesser well known algorithms have appeared in various literature over more recent years appear to perform efficiently for this situation, but their properties have not been well explored and extensively compared.

Motivated by the lack of guidance in choosing an appropriate relabelling algorithm in practice, where a balance between computational efficiency and theory must be struck. Our article has a two fold purpose: first, we extensively review and compare existing algorithms which are better suited to the large N and large q problem. Secondly, we introduce a new relabelling algorithm which is interpretable under the squared loss function, and compare its performance to existing methods. We note here that the problem of large K is particularly difficult for all relabelling algorithms. We note that large K can occur in relatively small data sets also, and of course the

larger K values will lead to larger q values. Methods that scale well under all three criterions K, N and q are particularly difficult to find, and we will discuss this further later. This article primarily focuses on the large N and q problems. Section 5.2 gives a brief review of all the existing algorithms studied in this article. Section 5.3 introduces a new algorithm , and we extensively compare these algorithms in Section 5.4, and conclude with some discussions and recommendations in Section 5.5.

5.2 Review of existing relabelling algorithms

In this section, we focus our review on relabelling methods which can handle high (q) dimensional problems, and those which will scale up well for large (N) dataset. In addition, readers are referred to the excellent review of Jasra et al. (2005) for a more general review for developments prior to 2005. We will focus more closely on scalable algorithms to large data.

We broadly separate the class of relabelling algorithm into two categories. One works on the full set of q-dimensional parameters ϕ, and we refer to these as full parameter space relabelling. A second category works on the allocation parameters only. We shall refer to these as the allocation space relabelling algorithms.

5.2.1 Full parameter space relabelling algorithms

Celeux et al (1998, 2000)

Celeux (1998) and Celeux et al. (2000) provided a simple algorithm for relabelling. A reference modal region is selected using the initial MCMC output, and subsequent points are then permuted with respect to the reference points, according to a k-means type algorithm.

Let $\phi^j = ((w_1^j, \theta_1^j), \ldots, (w_K^j, \theta_K^j))$ denote the vector of parameter estimates at the j^{th} iteration of the MCMC output. Initialise with the first m sample outputs, where m

is sufficiently large to ensure that the initial estimates are a reasonable approximation to the posterior means, but not so large that label switching has already occurred. Celeux et al. (2000) suggested that $m = 100$ is typically sufficient. Define component specific location and scale measures

$$\bar{\phi}_i = \frac{1}{m}\sum_{j=1}^{m}\phi_i^j$$

and

$$s_i = \frac{1}{m}\sum_{j=1}^{m}(\phi_i^j - \bar{\phi}_i)^2$$

for $i = 1, \ldots, q$. Then treating this as the initial ordering, $K! - 1$ other location and scale labels are produced from this set. We denote the entire initial set of permutations of location and scale values by $\{\bar{\phi}_{\nu_k}^{[0]}, s_{\nu_k}^{[0]}\}$, where ν_k denotes the set of all possible permutations.

Subsequent iterations of the relabelling algorithm then proceeds by allocating the permutation ν_{k^*} to the $m+r^{th}$ MCMC output vector ϕ^{m+r} which minimises the scaled Euclidean distance of all components $i = 1, \ldots, q$, namely we find the permutation ν_{k^*}

$$\nu_{k^*} = \underset{\nu_k}{\text{argmin}} \sum_{i=1}^{q} \frac{\phi_i^{m+r} - \bar{\phi}_{\nu_k,i}^{[r-1]}}{s_{\nu_k,i}^{[r-1]}},$$

where $\bar{\phi}_{\nu_k,i}^{[r-1]}$ and $s_{\nu_k,i}^{[r-1]}$ are respectively the i^{th} coordinate of the current estimate of the location and scale vector with respect to the permutation ν_k. Finally, the location and scale vectors are updated with the new r^{th} sample.

This algorithm works by minimising the scaled Euclidean distance to the cluster centers, assuming the initial centers provided a good estimate. In practice, the use of component variance for scaling, leads to those components with very small variances dominating the others, hence leading to inaccurate relabelling in these types of problems, as demonstrated in our simulation studies in later sections.

Früwirth-Schnatter (2011)

Frühwirth-Schnatter (2006) and Frühwirth-Schnatter (2011) proposed to apply the standard k-means algorithm with K clusters to all the MCMC sample output, with the posterior mode estimator $\phi_1^*, \ldots, \phi_K^*$ serving as starting value for the cluster means. They suggested that each element of the parameter vector should be standardised.

If the simulation clusters are well separated, then the classification sequence given by the classification index is a permutation. That is, the k-means algorithm allocates each component parameter vectors to exactly K clusters. However, this is not always the case, and the algorithm can often allocate multiple components to the same cluster. Frühwirth-Schnatter (2011) suggested that a simple check by ordering of the sequence of classification index. If this does not equal $\{1, \ldots, K\}$ then the sample is simply excluded.

The algorithm is very simple and efficient. It is easy to understand as it uses the well known k-means clustering algorithm. However, it can become inefficient when cluster components are very close to each other, leading to allocation of multiple components into the same cluster. Since such samples are then excluded for analyses, this can result in high proportion of waste of MCMC samples, which can themselves be expensive to calculate in high dimensional problems.

Marin et al (2005)

Marin et al. (2005) provided an algorithm for the reordering of MCMC output of size M. They first found the posterior mode ϕ^*, then for each sample, computed

$$\nu_{k^*} = \underset{\nu_k}{\text{argmin}} < \phi_{\nu_k}, \phi^* >_q,$$

where $<>_q$ is the canonical scalar product of $\mathcal{R}^q$.

Thus each MCMC output was reordered with respect to the approximate posterior MAP estimator. Several authors, e.g. Jasra et al. (2005) and Papastamoulis and Iliopoulos (2010) comment on the simplicity of the method, but note that it may fail when there's genuine multimodality in the parameters due to the dependence on the MAP estimator which can ignore the existence of minor modes.

5.2.2 Allocation space relabelling algorithms

In the allocation algorithms (see Richardson and Green 1997), a latent variable (allocation variable) $\mathbf{z}$ is introduced for each observation, which indicates the component membership. This approach is often used when clustering observations into different subsets is the aim. Relabelling based on allocation variable alone has the advantage that its computational cost is invariant to increases in the dimensionality of the parameter space.

The allocation variable is obtained by augmenting Equation (5.1) with the auxiliary variable $\mathbf{z} = (z_1, \ldots, z_n)$, such that

$$p(z_i = k) = w_k, \text{ for } k = 1, \ldots, K,$$

and

$$p(x_i|\phi, z_i) = f(x_i \mid \theta_k, z_i),$$

so that

$$p(x_i|\phi) = \sum_{k=1}^{K} w_k f(x_i \mid \theta_k, z_i). \tag{5.3}$$

Note that when the allocation variable is not used, the algorithms in this section can be used by computing a plug-in estimate of the allocation for each MCMC iteration j,

$$\hat{z}_i^j = \underset{k}{\text{argmax}}\ w_k f(x_i|\phi^j, z_i = k)/p(x_i|\phi^j). \tag{5.4}$$

Similar approaches can be found in for example, Stephens (2000).

Cron and West (2011)

Cron and West (2011) provided a relabelling algorithm based entirely on the latent variables. Define $\hat{z}$ to be the vector with n elements $\hat{z}_i$, which either arises naturally via the allocation sampler as in Equation (5.3), or it can be determined according to Equation (5.3). So $\hat{z}$ assigns each data observation to its modal component under the current set of classification probabilities. Define $\hat{z}^R$ as the classification vector with elements $\hat{z}_i^R$ at some reference point, ideally taken as the posterior mode. They suggested a Bayesian EM algorithm for the identification of posterior mode.

For each MCMC iteration, the algorithm proceeds by calculating the misclassification of $\hat{z}$ relative to $\hat{z}^R$, and permuting the component labels of z to maximise the match with $\hat{z}^R$ by calculating a misclassification cost matrix C, defined as

$$C_{hj} = \{\hat{z}_i^R = h \wedge \hat{z}_i \neq j\}, \quad i \in 1 \dots n, \quad j, h = 1, \dots, k.$$

Permutation of the misclassification matrix can be performed efficiently with the so-called Hungarian Algorithm (Munkres 1957), and the column permutation that minimises the $tr(C)$ is then recorded for each iteration of the MCMC sample.

Papastamoulis and Iliopoulos (2010)

Papastamoulis and Iliopoulos (2010) and Papastamoulis (2014) introduced a similar algorithm . Their algorithm can be seen as a modification of the pivotal reordering algorithm of Marin et al. (2005). The method was justified via an equivalence class representation, by redefining the symmetric posterior distribution to a nonsymmetric one via the introduction of an equivalence class.

More specifically, to determine the equivalence class, a vector z^* will be selected

to act as a pivot, such as the posterior mode. Then for each MCMC sample output z, the permutation that makes z as similar as possible to z^* will be selected. Hence the algorithm works very similar to Marin et al. (2005) with the difference being that the similarity measure here is based on the allocation variable defined as

$$S(z_1, z_2) := \sum_{i=1}^{n} I(z_{1i} = z_{2i})$$

for two allocation vectors z_1, z_2, where $I(A)$ is the indicator function of A.

5.3 A variance based relabelling algorithm

In this section, we propose a new algorithm motivated by the expected posterior mean squared loss function,

$$L(\phi, \hat{\phi}) = \mathbb{E}_{p(\phi|x)}[(\phi - \hat{\phi})^2] = \text{var}(\phi) + (\mathbb{E}(\phi) - \hat{\phi})^2, \tag{5.5}$$

where $p(\phi|x)$ is the posterior distribution. Thus minimising the above loss function amounts to minimizing

$$(\nu_k^*, \hat{\phi}^*) = \underset{\nu_k, \hat{\phi}}{\text{argmin}} \left[\text{var}(\phi_{\nu_k}) + (\mathbb{E}(\phi_{\nu_k}) - \hat{\phi})^2 \right]. \tag{5.6}$$

Since for a given permutation ν_k, setting $\hat{\phi}^*$ to the posterior mean minimises the second term in the above loss function. Hence to minimise Equation (5.5), we should find the permutation that minimises the posterior variance of the parameters.

In practice, exhaustive minimisation of Equation (5.6) is computationally prohibitive for large numbers of sample output. So similarly to Celeux (1998), Marin et al. (2005), Cron and West (2011), we first find reference points in the modal locations, and iteratively minimize the variance of the posterior samples with respect

to the permutations in the modal region. The following proposition shows that provided that the cluster means do not change very quickly, minimisation of Equation (5.6) can be performed iteratively.

Proposition 1 *Let* $V_m^* = \sum_{i=1}^q \widehat{var}(\phi_{\nu^*,i}^{[m]})$ *denote the minimum total variance of the parameters* $\phi_{\nu^*,i}^{[m]}$ *with corresponding optimal permutations* ν^**, based on* m *iterates of the MCMC output. Let* $V_{m+1}^* = \sum_{i=1}^q \widehat{var}(\{\phi_{\nu^{**},i}^{[m]}, \phi_{\nu^{m+1},i}^{(m+1)}\})$ *denote the minimum total variance based on the sample with one additional MCMC output, with the optimal permutations given by* ν^{**} *and* ν^{m+1}*. Denote the parameter means by* $\bar{\phi}_{\nu^*,i}^{[m]}$ *and* $\bar{\phi}_{\nu^{**},\nu^{m+1},i}^{[m+1]}, i = 1, \ldots, q$*. Suppose that* $\bar{\phi}_{\nu^*,i}^{[m]} \approx \bar{\phi}_{\nu^{**},i}^{[m]}$*, then the optimal permutations* $\nu^{**} = \nu^*$*, and* V_{m+1}^* *can be minimised by permutation of the vector* $\phi^{(m+1)}$ *only.*

Proof: See Appendix.

Thus as long as the successive parameter means do not change much under optimal reordering, we can minimize the variance criterion iteratively, only reordering each new sample, while keeping the ordering of the previous samples unchanged. This condition is reasonable in standard MCMC sampling where parameters do not change values very drastically from iteration to iteration. In problems of genuine multimodality, the standard MCMC sampler may fail to reach minor modes (see Jasra et al. 2005). In these cases, more advance sampling techniques allowing for larger MCMC moves, such as simulated tempering or adaptive MCMC, may be necessary. Thus in these cases, our algorithm may not be appropriate to apply to these types of algorithms.

5.3.1 Minimum Variance algorithm

Here we give an algorithm based on minimising the variance of the parameters. The algorithm is based on the full parameter space, similar to those in Section 5.2.1.

Step 1: Select m posterior samples from the modal region, such that no switching has occurred.

Step 2: Exclude the samples used in Step 1. For $r = 1, \dots M$, each successive iteration of the MCMC output is relabelled according to

$$\nu_k^{(m+r),*} = \underset{\nu_k^{(m+r)}}{\operatorname{argmin}} \sum_{i=1}^{q} \widehat{\operatorname{var}}(\{\phi_{\nu_k^*,i}^{[m+r-1]}, \phi_{\nu_k^{m+r},i}^{r}\}),$$

where $\widehat{\operatorname{var}}(\phi_{\nu_k,i}^{[m+r-1]})$ is the sample variance for the ith parameter, under the permutation ν_k^*, corresponding to the set of previous $m+r-1$ samples. Relabel the $(m+r)$th sample according to $\nu_k^{(m+r),*}$.

In Step 1, we choose a small set of modal posterior samples, where no switching has occurred, but a good estimate of the posterior means can be obtained. This is similar to the approach suggested in Celeux (1998). A number between 50 to 100 is typically sufficient. Step 2 involves only permuting the labelling of the rth sample to minimise the overall posterior variance including the new sample ϕ^r. A computationally efficient update of the variance for each of the ith component is given by iteratively computating:

$$\bar{\phi}_i^{[m+r]} = \frac{1}{m+r}[(m+r-1)\bar{\phi}_i^{[m+r-1]} + \phi_i^r]$$

and

$$\widehat{\operatorname{var}}(\phi_i^{[m+r]}) = \frac{m+r-2}{m+r-1}\widehat{\operatorname{var}}(\phi_i^{[m+r-1]}) + \frac{1}{m+r}(\phi_i^r - \bar{\phi}_i^{[m+r-1]})^2,$$

where $\bar{\phi}_i^{[m]}$ denotes the sample mean of the ith parameter based on m samples.

5.3.2 Simultaneous monitoring of MCMC convergence

We note an interesting connection of the variance based relabelling algorithm with the well known Gelman and Rubin convergence assessment. Given J parallel MCMC sequences, each with length M, Gelman and Rubin (1992) suggested to monitor the so called potential scale reduction factor R at MCMC iteration m, estimated as

$$\hat{R} = \sqrt{\frac{\widehat{\text{var}}(\phi_i)}{W}},$$

where

$$\widehat{\text{var}}(\phi_i) = \frac{m-1}{m}W + \frac{1}{m}B$$

and W is the within chain variance of the ith marginal parameter based on m samples,

$$W = \frac{1}{J}\sum_{j=1}^{J} \widehat{\text{var}}(\phi_i^{[m]}).$$

Note that W is readily given by Step 2 in the algorithm above.

B is the between chain variance

$$B = \frac{M}{J-1}\sum_{j=1}^{J}(\bar{\phi}_{i,j}^{[m]} - \bar{\phi}^{[m]})^2,$$

where $\bar{\phi}_{i,j}^{[m]}$ is the sample mean of the j chain, for the ith parameter based on m samples, and $\bar{\phi}^{[m]} = \frac{1}{J}\sum_{j=1}^{J}\bar{\phi}_{i,j}^{[m]}$. Again $\bar{\phi}_{i,j}^{[m]}$ is given in Step 2 of the algorithm for a given jth chain. Thus the potential scale reduction factor is readily calculated, a value approaching 1 is indicative of MCMC convergence.

Thus to monitor the convergence of multiple MCMC sequence for each marginal parameter i, the above algorithm only has to be modified slightly. In Step 1, instead of selecting samples m from a single chain, we will select J equal sized samples $m_j, \sum_{j=1}^{J} m_j = m$ amongst the modal regions of the J parallel chains. Then in Step

2, for each chain $j = 1, \dots J$, and their respective initial samples m_j, carry out Step 2 and calculate $\hat{R}$.

5.4 Examples

In this section, we will compare all the algorithms presented above in several examples involving both real and simulated data. All algorithms were coded by the authors in Matlab, with the exception of Cron and West (2011), where we used the codes supplied by the authors on their website *https://stat.duke.edu/gpustatsci/software.html*. The algorithm of Papastamoulis et al (2010) is available as an R package, see Papastamoulis and Papastamoulis (2013). All computations were carried out on Ubuntu (x86_64) with kernel version of 3.2.0-53-generic.

5.4.1 Univariate mixtures

Simulated data

We first consider two univariate mixture models, a three-component and a five-component model,

$$0.10N(-20, 1) + 0.65N(20, 3) + 0.25N(21, 0.5) \tag{5.7}$$

$$0.20N(19, 5) + 0.20N(19, 1) + 0.25N(23, 1) + 0.20N(29, 0.5) + 0.15N(33, 3). \tag{5.8}$$

In the three-component model, the final two components are very close together. We expect that it will be easy to identify the first component, but not the last two. Similarly within the five-component model, the first two components will be extremely difficult to separate. This example was also studied in detail by Papastamoulis and Iliopoulos (2010).

We use 100 data points simulated from each of the two models, and follow the

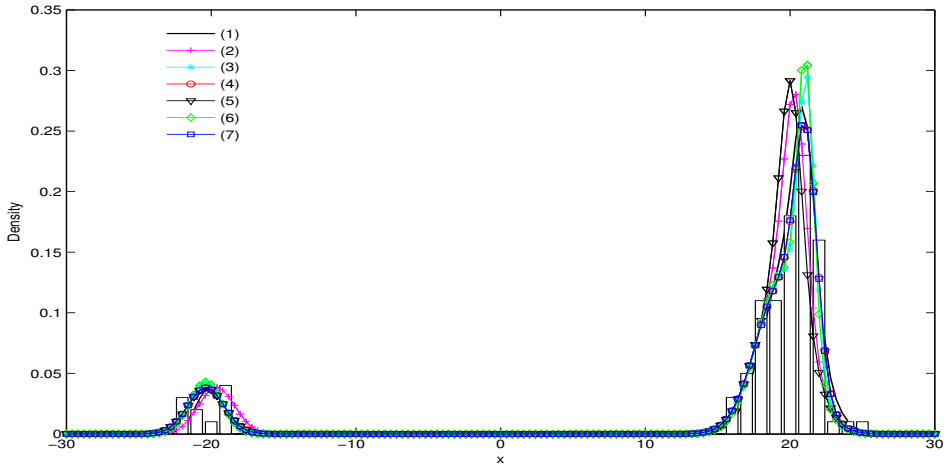
MCMC sampler of Richardson and Green (1997). For both models, we run 80,000 iterations of MCMC sampling and discard the first 20,000 as burn in. Figure 5.1 shows the density estimates using each of the six different methods we discussed, superimposed with their true density. Clearly, all methods agree in regions where identifiability is easily separable, and differences between the different methods are more pronounced where components are very close together. This is the case for the last two components in Model (5.7) and the first two components in Model (5.8).

Overall, with the exception of the method of Celeux et al (1998, 2000), the other methods give similar density estimates. It can be seen that in both examples, the k-means method of Früwirth-Schnatter (2011) is most similar to the equivalence class method of Papastamoulis et al (2010), although one is based on the full parameter space and the other is based on only the allocation variables. It can be seen that the left hand tail of Model (5.8) is under-estimated by the method of Papastamoulis et al (2010) relative to the other methods. We will return to this issue later.

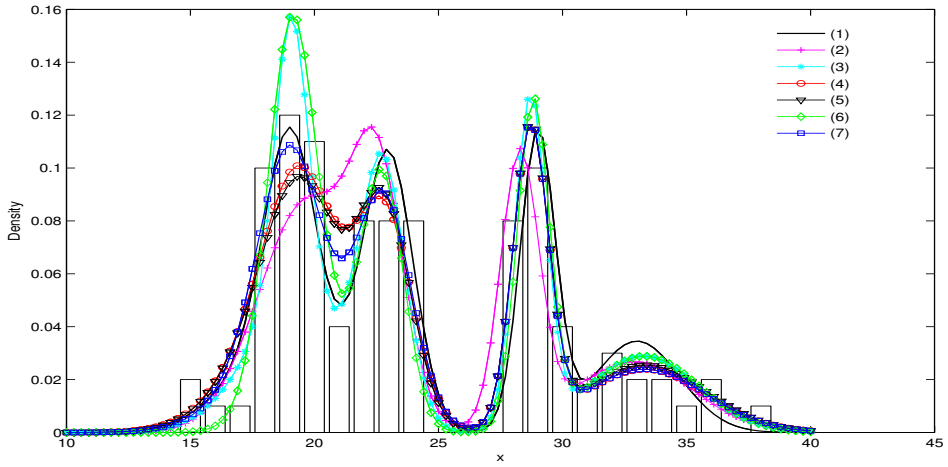
We find that the method of Celeux et al (1998, 2000) does not perform well in both cases. This is due to the use of a scaled distance, the process can be dominated by those components with very small variances. From the misclassification table given in Table 5.1, we can see that the second component of Model (5.8) has been completely misclassified by the method of Celeux et al (1998, 2000), the second component was dominated by the first component.

Finally, we present a more thorough comparison of the six different method in Table 5.2, where we give an estimate of the KL distance between the true density and the estimated densities, the overall misclassification rates (as computed in Table 5.1), the total variance of the parameter estimates and the CPU time.

Overall, Celeux et al (1998, 2000) has the largest KL distance, overall misclassification rate and total variance, although its computational time is competitive with the other algorithms. We note that the method of Früwirth-Schnatter (2011) requires



(a) Mixture of Equation (5.7)



(b) Mixture of Equation (5.8)

Figure 5.1: Histogram of 100 simulated observations from model (5.7) and (5.8). The superimposed lines correspond to (1) true density, (2) Celeux et al (3) Früwirth-Schnatter (4) Marin et al (5) Cron and West (6) Papastamoulis et al (7) Minimum Variance.

far more MCMC sample output than the other methods, since samples which has been clustered into less than K components has been discarded by the algorithm. Hence to obtain 60,000 samples, we run approximately 27,000 additional MCMC iterations for Equation (5.7) and an additional 90,000 iterations for (5.8). Thus even though it is a fast algorithm itself, the computational overheads in the additional MCMC sampling makes this algorithm by far the most computationally costly. In addition, although the method achieves good misclassification rate, it would appear we cannot trust the resulting parameter estimates, see Table 5.3. We believe this may be attributed to the non-random exclusion of samples from the MCMC output.

The remaining methods of Marin et al (2005), Cron and West (2011), Papastamoulis et al (2010) and the proposed Minimum Variance algorithm, all performe relatively well. Minimum Variance gives the smallest KL distance, with similar results using Marin et al (2005). The best method in terms of misclassification rate is Papastamoulis et al (2010), with Cron and West (2011) marginally worse. In terms of posterior variance, Minimum Variance algorithm produced the smallest values, closely followed by Papastamoulis et al (2010). In terms of CPU time, all methods are efficient, the best ones being Papastamoulis et al (2010) and Marin et al (2005), and the Minimum Variance algorithm is the slowest here.

Finally, the parameter estimates given in Table 5.3 show that while the mean parameters are fairly well estimated by most methods, the variance estimates are quite different. It is clear that Marin et al (2005), Cron and West (2011) and Minimum Variance all overestimate the 2nd and the last variance components of Equation (5.8). Papastamoulis et al (2010) underestimates the variance of component one while overestimating the variance of the last component. Overall, the variance estimates are generally smaller from Papastamoulis et al (2010) than the other three methods, and is generally underestimated relative to the true values. While the variance estimates are generally overestimated from Marin et al (2005), Cron and West (2011) and Min-

imum Variance relative to the true values.

	Eq. (5.7)			Eq. (5.8)				
True	10	0	0	20	0	0	0	0
	0	65	0	0	20	0	0	0
	0	0	25	0	0	25	0	0
				0	0	0	20	0
				0	0	0	0	15
Celeux et al	10	0	0	18	0	2	0	0
	0	43	22	20	0	0	0	0
	0	11	14	2	1	22	0	0
				0	0	0	20	0
				0	0	0	1	14
Früwirth-Schnatter	10	0	0	3	15	2	0	0
	0	55	10	0	20	0	0	0
	0	14	11	0	2	23	0	0
				0	0	0	20	0
				0	0	0	2	13
Marin et al	10	0	0	6	12	2	0	0
	0	41	24	3	17	0	0	0
	0	6	19	0	3	22	0	0
				0	0	0	20	0
				0	0	0	2	13
Cron and West	10	0	0	6	12	2	0	0
	0	44	21	3	17	0	0	0
	0	16	9	0	3	22	0	0
				0	0	0	20	0
				0	0	0	2	13
Papastamoulis et al	10	0	0	6	12	2	0	0
	0	42	23	0	20	0	0	0
	0	6	19	2	1	22	0	0
				0	0	0	20	0
				0	0	0	2	13
Minimum Variance	10	0	0	6	12	2	0	0
	0	40	25	0	20	0	0	0
	0	6	19	2	1	22	0	0
				0	0	0	20	0
				0	0	0	2	13

Table 5.1: Misclassification matrix for the six methods. Each i, jth entry of the misclassification matrix denotes the number of observations which is classified as component j, while actually it belongs to component i. The row corresponding to True gives the true cluster membership of the observed data.

	KL Distance		Misclassification		Total Variance		Time (sec)	
	(5.7)	(5.8)	(5.7)	(5.8)	(5.7)	(5.8)	(5.7)	(5.8)
Celeux et al	0.21	0.38	33%	26%	62.61	82.89	19.55	42.69
Früwirth-Schnatter	0.08	0.11	24%	21%	1.68	8.89	14.38	28.35
Marin et al	0.07	0.14	30%	22%	2.30	55.79	17.66	40.19
Cron and West	0.31	0.14	37%	22%	2.38	56.11	26.63	38.84
Papastamoulis et al	0.11	0.35	29%	19%	2.35	52.96	16.11	25.05
Minimum Variance	0.07	0.11	31%	19%	2.30	52.02	24.83	50.98

Table 5.2: Comparison of KL distance relative to the true distribution, misclassification rate, total variance for the parameter estimates and computation time, for the six different methods outlined, using simulated data from Equations (5.7) and (5.8) .

Real data: Galaxy dataset

Here we compare the various methods on the well known galaxy data, which has been studied extensively in the relabelling literature, see for example Stephens (1997b), Celeux et al. (2000), Jasra et al. (2005). This dataset consists of the velocities of several galaxies diverging from our own galaxy. The original dataset consists of 83 observations, but one of them is recorded as infinite. We leave this one out and use the remaining 82 observations. We follow the setup of Richardson and Green (1997) in setting up the model and MCMC sampling, and fix the number of mixture components at 6, which was shown to have the highest posterior model probability. We run 80,000 MCMC iterations and discard the first 20,000 iterations, keeping the final 60,000 samples. For the method of Früwirth-Schnatter (2011), we run an additional 320,000 iterations.

Figure 5.2 shows histogram and density estimate of galaxy data. Here the differences between the methods are more pronounced than in the previous example. Again, it is clear that both Celeux et al (1998, 2000) and Früwirth-Schnatter (2011) are not performing well. Both the figure and Table 5.4 show that the method of Marin et al (2005), Cron and West (2011) and Minimum Variance are the most similar to each other, and give smaller total variance estimates. Papastamoulis et al (2010) is the most efficient in terms of computing time, suggesting that the method scales up

	$\hat{w}_k$		$\hat{\mu}_k$		$\hat{\sigma}_k^2$	
	(5.7)	(5.8)	(5.7)	(5.8)	(5.7)	(5.8)
True	0.10	0.20	-20.00	19.00	1.00	5.00
	0.65	0.20	20.00	19.00	3.00	1.00
	0.25	0.25	21.00	23.00	0.50	1.00
		0.20		29.00		0.50
		0.15		33.00		3.00
Celeux et al	0.12	0.24	-19.63	19.40	1.55	3.45
	0.55	0.19	19.41	20.20	3.13	4.05
	0.33	0.21	20.37	22.58	0.57	1.06
		0.20		28.28		0.60
		0.16		32.85		5.69
Früwirth-Schnatter	0.12	0.05	-20.37	15.80	1.24	0.87
	0.58	0.34	19.52	19.15	0.97	0.46
	0.30	0.25	21.16	22.83	0.32	0.88
		0.21		28.75		0.55
		0.15		33.41		3.66
Marin et al	0.12	0.23	-20.37	19.01	1.57	4.90
	0.55	0.21	19.42	19.43	3.14	1.98
	0.33	0.21	21.11	22.85	0.55	1.23
		0.20		28.72		0.50
		0.15		33.30		6.25
Cron and West	0.12	0.22	-20.37	18.99	1.57	4.82
	0.56	0.21	19.45	19.47	3.14	2.14
	0.32	0.21	21.07	22.84	0.55	1.14
		0.20		28.72		0.50
		0.16		33.30		6.25
Papastamoulis et al	0.12	0.18	-20.37	19.51	1.25	1.93
	0.56	0.25	19.44	19.05	2.87	0.84
	0.32	0.21	21.08	22.73	0.30	0.76
		0.20		28.72		0.42
		0.16		33.30		4.89
Minimum Variance	0.12	0.22	-20.37	19.50	1.57	5.43
	0.55	0.22	19.42	18.94	3.14	1.49
	0.33	0.21	21.10	22.85	0.55	1.18
		0.20		28.72		0.50
		0.15		33.30		6.25

Table 5.3: Parameter estimates using the six different methods. The left part of each column corresponds to Equation (5.7), the right part of each column corresponds to Equation (5.8).

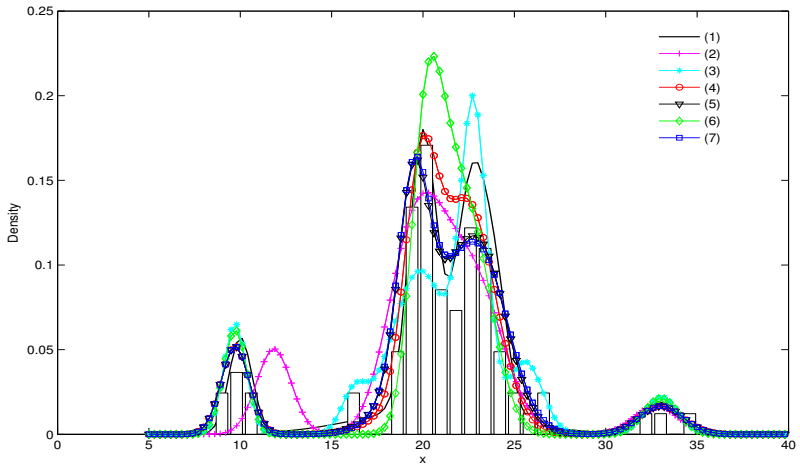


Figure 5.2: Histogram of the galaxy data. The superimposed lines correspond to (1) posterior MAP density estimate (2) Celeux et al (3) Früwirth-Schnatter (4) Marin et al (5) Cron and West (6) Papastamoulis et al (7) Minimum Variance.

	KL Distance	Total Variance	Time (sec)
Celeux et al	2.94	91.87	2.94
Früwirth-Schnatter	1.89	8.89	71.70
Marin et al	1.37	37.57	40.45
Cron and West	1.40	40.07	49.79
Papastamoulis et al	2.61	52.06	29.35
Minimum Variance	1.30	38.52	87.70

Table 5.4: Comparison of KL distance relative to the MAP density estimate, total variance for the parameter estimates and computation time, using the six different methods, for the galaxy data.

well with the number of components. We see from the table of parameter estimates in Table 5.5 that the estimates of variances are generally smaller for Papastamoulis et al (2010) than the other methods.

MAP	$\hat{w}_k$	0.09	0.31	0.15	0.33	0.07	0.05
	$\hat{\mu}_k$	10.01	20.00	20.60	22.76	24.14	32.86
	$\hat{\sigma}_k^2$	0.40	0.57	16.38	0.84	0.42	1.08
Celeux et al	$\hat{w}_k$	0.12	0.15	0.27	0.18	0.23	0.05
	$\hat{\mu}_k$	11.84	18.73	20.19	21.74	22.95	32.72
	$\hat{\sigma}_k^2$	0.90	1.48	1.75	4.19	2.25	1.41
Früwirth-Schnatter	$\hat{w}_k$	0.10	0.04	0.35	0.35	0.10	0.05
	$\hat{\mu}_k$	9.71	16.38	19.79	22.58	25.48	33.03
	$\hat{\sigma}_k^2$	0.37	0.50	0.47	1.14	0.76	0.81
Marin et al	$\hat{w}_k$	0.10	0.22	0.20	0.22	0.20	0.05
	$\hat{\mu}_k$	9.72	19.46	20.40	22.12	23.46	33.02
	$\hat{\sigma}_k^2$	0.59	0.65	4.20	3.30	1.81	1.42
Cron and West	$\hat{w}_k$	0.10	0.27	0.16	0.16	0.27	0.05
	$\hat{\mu}_k$	9.72	19.86	20.71	22.15	22.72	33.02
	$\hat{\sigma}_k^2$	0.59	0.76	4.94	2.26	2.00	1.42
Papastamoulis et al	$\hat{w}_k$	0.10	0.25	0.20	0.13	0.27	0.05
	$\hat{\mu}_k$	9.72	20.00	20.79	21.74	22.90	33.02
	$\hat{\sigma}_k^2$	0.42	0.63	0.76	0.92	1.33	0.89
Minimum Variance	$\hat{w}_k$	0.10	0.22	0.20	0.22	0.21	0.05
	$\hat{\mu}_k$	9.72	19.46	20.38	22.08	23.52	33.02
	$\hat{\sigma}_k^2$	0.59	0.68	3.63	3.67	1.98	1.42

Table 5.5: Parameter estimates for galaxy dataset using different relabelling algorithms and the MAP estimate.

5.4.2 Multivariate spatial mixture model for image processing

We consider a multivariate spatial mixture model in the context of image analysis, where the both the dimension of the mixture, as well as the dataset itself can be large.

We use a simulated 3-D image of $50 \times 50 \times 16$ voxels, this is equivalent to having 40,000 observations. We assume that each voxel comes from a 3 dimensional mixture model of two components, with mean parameters $\mu_1 = [4, 5, 6]$ and $\mu_2 = [6, 7, 8]$

respectively. The corresponding covariance matrices are:

$$0.5 \times \begin{pmatrix} 1.00 & 0.80 & 0.64 \\ 0.80 & 1.00 & 0.80 \\ 0.64 & 0.80 & 1.00 \end{pmatrix} \text{ and } 0.5 \times \begin{pmatrix} 1.00 & 0.50 & 0.25 \\ 0.50 & 1.00 & 0.50 \\ 0.25 & 0.50 & 1.00 \end{pmatrix}.$$

In real applications, such as in dynamic positron emission tomography (PET), or functional MRI studies, the number of observations and the dimensions of the mixture is much larger, but the number of mixture components are typically between 3 to 5. This example demonstrates the need for fast and reliable relabelling algorithms.

To simulate a spatially dependent image, we first simulate the voxels using a Potts (or Ising in the case of two component mixtures) model, with spatial correlation parameter set to 0.3 (Feng et al. 2012), and then assign voxel values according to the component Normal distributions. See Figure 5.3 for a plot of the true allocations.

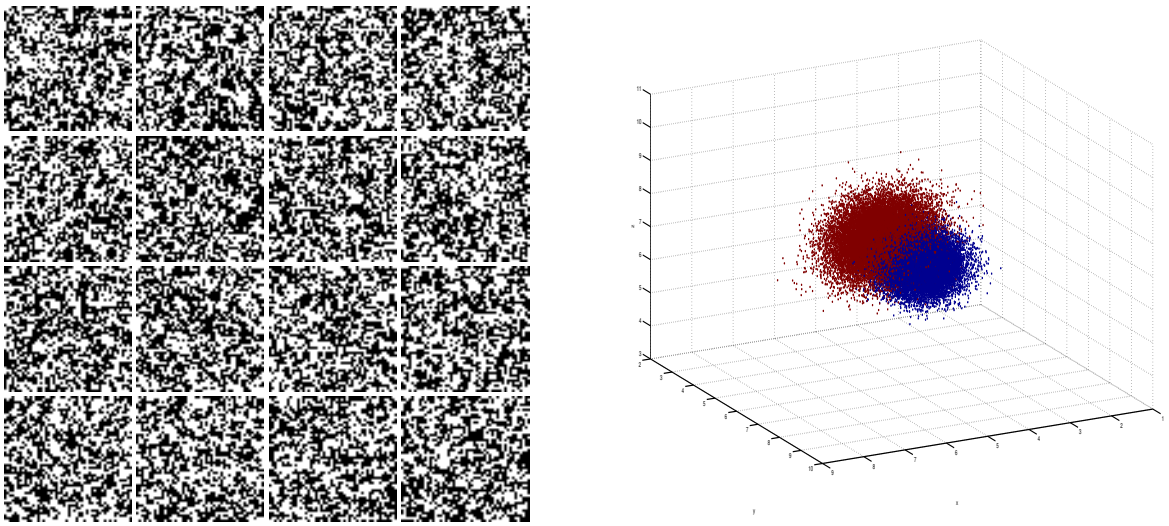


Figure 5.3: The true allocations shown slice by slice (left). The white points correspond to the component with $\mu = [4, 5, 6]$; and black ones denote the component with $\mu = [6, 7, 8]$ and, 3D scatter plot of the two components (right).

In the spatial clustering model, we have

$$p(x_i|\phi, z_i) = \sum_{k=1}^{K} f(x_i \mid \theta_k, z_i)p(z_i = k), \tag{5.9}$$

where the distribution of the allocation variables is given by the Potts model,

$$P(z|\kappa) = \frac{1}{c(\kappa)} \exp\{\kappa \sum_{i \in \delta_j} I(z_i = z_j)\}, \tag{5.10}$$

where δ_j denotes the neighbourhood of j, and κ denotes the strength of the spatial connectedness (c.f. Equation (5.3)). The normalising constant $c(\kappa)$ is intractable, and we follow Green and Richardson (2002) and Smith and Smith (2006) in precomputing these in a look -up table.

We set the prior for κ to be a truncated Normal(0.6,100) on the interval $[0, 1]$, and use conjugate priors for the mean parameter $\mu_k|\Sigma_k \sim N(0, 100 \times \Sigma_k)$. Covariance matrices follow an Inverse-Wishart distribution $\Sigma_k \sim IW(3, 1.5 \times I_{3\times3})$, for $k = 1, \ldots, K$. A hybrid Gibbs within Metropolis sampler can be constructed from the full conditional distributions, and convergence of the MCMC sampler is obtained after 10,000 iterations, discarding the initial 5,000 samples as burn in. In order to guarantee the presence of the label switching phenomenon, we manually switch the samples during simulation, see Papaspiliopoulos and Roberts (2008) and Jasra et al. (2005).

Table 5.6 provides the posterior mean estimates for the model parameters using different reordering schemes. Here all the methods performed well, as the two mixture components are fairly well separated in the example. Table 5.7 gives the comparative KL divergence, misclassification rates, total variance and computing time. Due to the moderate large size of the data, small differences in the posterior parameter estimates for the method of Celeux et al translate into a relatively large KL measure. The other methods are all comparable in terms of misclassification and total variance. In terms of computational time, with the allocation based methods taking

longer than the full parameter based methods. This example illustrates that parameter based algorithms scale up better when the size of the observation increases, since the corresponding increase in the number of allocation variables needed do not affect the efficiency in the relabelling algorithms. On the other hand, if the dimension of the parameter space increase, i.e, as the dimension of the multivariate Normals increase, we would expect the allocation based relabelling algorithms to be more efficient.

	$\hat{\mu}_k$	$\hat{\sigma}_k^2$	
True	$\begin{pmatrix} 4.00 & 5.00 & 6.00 \\ 6.00 & 7.00 & 8.00 \end{pmatrix}$	$\begin{pmatrix} 0.50 & 0.40 & 0.32 \\ 0.40 & 0.50 & 0.40 \\ 0.32 & 0.40 & 0.50 \end{pmatrix}$	$\begin{pmatrix} 0.50 & 0.25 & 0.13 \\ 0.25 & 0.50 & 0.25 \\ 0.13 & 0.25 & 0.50 \end{pmatrix}$
Celeux et al	$\begin{pmatrix} 4.05 & 5.05 & 6.05 \\ 5.97 & 6.97 & 7.97 \end{pmatrix}$	$\begin{pmatrix} 0.51 & 0.41 & 0.32 \\ 0.41 & 0.52 & 0.41 \\ 0.32 & 0.41 & 0.51 \end{pmatrix}$	$\begin{pmatrix} 0.50 & 0.25 & 0.13 \\ 0.25 & 0.50 & 0.25 \\ 0.13 & 0.25 & 0.50 \end{pmatrix}$
Früwirth-Schnatter	$\begin{pmatrix} 4.01 & 5.01 & 6.01 \\ 6.01 & 7.01 & 8.01 \end{pmatrix}$	$\begin{pmatrix} 0.51 & 0.41 & 0.33 \\ 0.41 & 0.52 & 0.41 \\ 0.33 & 0.41 & 0.51 \end{pmatrix}$	$\begin{pmatrix} 0.50 & 0.25 & 0.11 \\ 0.25 & 0.49 & 0.24 \\ 0.11 & 0.24 & 0.50 \end{pmatrix}$
Cron and West	$\begin{pmatrix} 4.01 & 5.01 & 6.01 \\ 6.01 & 7.01 & 8.01 \end{pmatrix}$	$\begin{pmatrix} 0.51 & 0.41 & 0.33 \\ 0.41 & 0.52 & 0.41 \\ 0.33 & 0.41 & 0.51 \end{pmatrix}$	$\begin{pmatrix} 0.50 & 0.25 & 0.12 \\ 0.25 & 0.50 & 0.24 \\ 0.12 & 0.24 & 0.51 \end{pmatrix}$
Marin et al	$\begin{pmatrix} 4.01 & 5.01 & 6.01 \\ 6.01 & 7.01 & 8.01 \end{pmatrix}$	$\begin{pmatrix} 0.51 & 0.41 & 0.33 \\ 0.41 & 0.52 & 0.41 \\ 0.33 & 0.41 & 0.51 \end{pmatrix}$	$\begin{pmatrix} 0.50 & 0.25 & 0.12 \\ 0.25 & 0.49 & 0.24 \\ 0.12 & 0.24 & 0.50 \end{pmatrix}$
Papastamoulis et al	$\begin{pmatrix} 4.01 & 5.01 & 6.01 \\ 6.01 & 7.01 & 8.01 \end{pmatrix}$	$\begin{pmatrix} 0.51 & 0.41 & 0.33 \\ 0.41 & 0.52 & 0.41 \\ 0.33 & 0.41 & 0.51 \end{pmatrix}$	$\begin{pmatrix} 0.50 & 0.25 & 0.12 \\ 0.25 & 0.49 & 0.24 \\ 0.12 & 0.24 & 0.50 \end{pmatrix}$
Minimum Variance	$\begin{pmatrix} 4.01 & 5.01 & 6.01 \\ 6.01 & 7.01 & 8.01 \end{pmatrix}$	$\begin{pmatrix} 0.51 & 0.41 & 0.33 \\ 0.41 & 0.52 & 0.41 \\ 0.33 & 0.41 & 0.51 \end{pmatrix}$	$\begin{pmatrix} 0.50 & 0.25 & 0.12 \\ 0.25 & 0.50 & 0.24 \\ 0.12 & 0.24 & 0.50 \end{pmatrix}$

Table 5.6: Posterior mean estimates of the two-components multivariate spatial mixture model, for the six different methods.

5.4.3 Further comparison of Computational time

In this section, we further study the effect on computational time, in terms of the number of mixture components K, the sample size of the data N and the number of parameters in the model q. Table 5.8 provides the CPU time used to compute each of the relabelling algorithms under our study. Note that the algorithms of Cron

	KL	misclassification	Total Variance	Time (sec)
Celeux et al	4300.80	7.51%	0.48	3.00
Früwirth-Schnatter	38.87	7.50%	$8.005 * 10^{-4}$	0.39
Marin et al	38.03	7.50%	$8.008 * 10^{-4}$	0.48
Cron and West	37.80	7.50%	$8.008 * 10^{-4}$	113.26
Papastamoulis et al	25.93	7.50%	$8.008 * 10^{-4}$	18.46
Minimum Variance	38.83	7.50%	$8.008 * 10^{-4}$	1.39

Table 5.7: Comparison of KL divergence, misclassification rates, total variance and computing time for the six different methods. The multivariate spatial mixture model.

and West and Papastamoulis et al were performed using the authors own codes, we do not compare the absolute computational time between algorithms, but rather the scalability within each algorithm.

To study the effect of K on computational time, we used a univariate spatial mixture model of Equation (5.9), with K=3, 5,7, and $N = 100$, $q = 3K$. The results show that Celeux et al, Marin et al and Minimum variance are the worst in terms of scalability of K. This is unsurprising, as these algorithms relied on the simple permutations on the parameter space, and the computational burden increases with the number of permutations in the order of $K!$. The k-means algorithm of Früwirth-Schnatter, and the allocation space relabelling algorithms of Cron and West and Papastamoulis et al all performed better, although all these algorithms still experience large increases in computational time as K increases.

To study the effect of sample size N, we fix $K = 3$, and use a tri-variate mixture model, so $q = 30$, and with varying values of $N = 4608, 36864, 52488$. Here it is clear that all the parameter based algorithms perform well, since they are invariant to the increases in sample size. Similarly the k-means algorithm. However, the allocation based algorithms will increase in computational time as N dramatically increases.

Finally, we study the effect of increases in q, here we fix $K = 3$ and $N = 36864$, and consider 2, 3, 4 dimensional mixtures. Thus the corresponding $q = 18, 30, 60$. Here all the algorithms appear to scale relatively well to the increase in q. While we

would expect the parameter based algorithms to perform worse with increasing q, it appears that this effect is relatively small. The k-means algorithms is again very efficient under this case. The allocation based algorithms are also very stable under increases in q, however, their overall computational time appears to be much higher than the others.

	K=3	K=5	K=7	N=4608	N=36864	N=52488	q=18	q=30	q=60
Celeux et al	3.26	7.12	62.38	7.14	7.17	7.29	2.70	3.02	3.42
Früwirth-Schnatter	2.40	4.73	28.35	1.73	1.80	2.24	0.92	1.21	1.77
Marin et al	2.94	6.70	124.3	6.62	6.81	6.64	2.21	2.50	2.85
Cron and West	4.44	6.47	12.50	59.03	130.6	191.0	172.8	147.4	154.0
Papastamoulis et al	2.69	4.18	7.08	14.8	15.62	21.42	15.81	16.61	18.51
Minimum Variance	4.14	8.50	64.53	4.94	4.97	5.40	3.29	3.64	5.07

Table 5.8: Time (in sec) used in different scenarios. For each column we fix other parameters.

5.5 Summary and conclusion

In this paper, we introduce a new algorithm based on a loss function argument. We also comprehensively compare the new algorithm with some existing relabelling algorithms, restricting our comparison to those algorithms which are scalable to large dataset N and large parameter space q. Where applicable, we compute KL divergence, misclassification rate, total variance of posterior parameter estimates and computing time, based on several examples including uni and multivariate spatial mixture models, as well as on a real dataset. Generally speaking, full parameter space relabelling algorithms can scale up well with both N and q. While allocation space relabelling algorithms can scale up well with q and perform relatively faster in large K.

We find that the method of Celeux et al (1998, 2000) can be very sensitive, and does not always perform well. The method of Frühwirth-Schnatter (2011) is generally very fast in large K, N and q, but can requires much more additional MCMC

sampling if the clusters are close together. Therefore, we do not recommend these two methods as a default approach for relabelling. The performance of the remaining four methods are similar, in terms of the criterions we use. All these methods have performed well, under the different conditions. However, all the methods give slightly different solutions.

In terms of performance, we can broadly group the method of Marin et al. (2005) and our proposed Minimum Variance algorithm together. Both are based on full parameter vectors, and show comparable performance in all the simulations we have considered. The other two, the method of Cron and West (2011) and Papastamoulis and Iliopoulos (2010), are based on allocation variables. Although all four methods produce similar results, the method of Papastamoulis and Iliopoulos (2010) tends to produce an underestimated variance parameter estimate, while the other three produced an overestimated variance. Broadly speaking, the full parameter methods are more efficient for large datasets and the allocation methods are more efficient when the parameter space is large and the number of components K is also moderately large. From a more theoretical perspective, while Marin et al. (2005) simply used the canonical scalar product as an optimisation criterion, the Minimum Variance algorithm minimises the expected posterior loss. But Cron and West (2011) minimises the misclassification matrix and the algorithm of Papastamoulis and Iliopoulos (2010) is justified by an equivalence class representation. Thus from a theoretical perspective, the Minimum Variance algorithm and Papastamoulis and Iliopoulos (2010) is more satisfying. We summarise the above discussion in Table 5.9. Note that we omit scalability of K for Cron and West and Papastamoulis et al, while these two perform better for large K, we do not consider it scalable for very large K.

Finally, we note that in practice, all methods can fail to find the correct labelling, see Cron and West (2011). In particularly in the presence of genuine multimodality, i.e., in the presence of multiple modes under any one mixture component, it has

	Optimisation criterion	Scalability	potential issues
Marin et al	scalar product	N &q	overestimation of variance
Cron and West	misclassification	q	overestimation of variance
Papastamoulis et al	equivalence class	q	underestimation of variance
Minimum Variance	expected squared loss	N & q	overestimation of variance

Table 5.9: Summary of the main points for the four methods, Marin et al, Cron and West, Papastamoulis et al and Minimum Variance.

been noted that different algorithms will give very different results, while the results will be broadly similar otherwise (Jasra et al. 2005). Running several of the above (time efficient) algorithms will easily allow us to identify potential problems. Thus, under such problematic situations, we recommend careful investigation to the cause of the problem and the application of more problem specific methods. For instance, under genuine multimodality, Grün and Leisch (2009) developed methods specifically for such situations, although the more sophisticated methods can be very time consuming to compute.

Our simulated comparisons highlights the difficulty in distinguishing a clearly superior algorithm. From a practical perspective, we find four of the algorithms (including a novel approach introduced in this article) have similar performance, and the user may base their choice on computational considerations. Computational time is an important factor when choosing the algorithms. Especially, when all the algorithms can perform similarly, computational time becomes more important. For the applications where N and q are large, our method should be one option to be considered. Because it can scale up well with N and q, resulting in reduction in computational time. Our method is of great significance in the applications of medical imaging in terms of computational time.

Appendix

Proof of Proposition 1

Let $V_m^*(\phi_{\nu^*}^{[m]}) = \sum_{i=1}^q \widehat{\text{var}}(\phi_{\nu^*,i}^{[m]})$ denote the minimum total variance of the parameters $\phi_{\nu^*,i}^{[m]}$ with corresponding optimal permutations ν^*, based on m samples. Suppose we have an additional sample $m+1$, then

$$\begin{aligned} V_{m+1}(\phi_{\nu}^{[m+1]}) &= \sum_{i=1}^{q} \widehat{\text{var}}(\phi_{\nu,i}^{[m+1]}) \\ &= \sum_{i=1}^{q} \left[\frac{m-1}{m} \widehat{\text{var}}(\phi_{\nu,i}^{[m]}) + \frac{1}{m+1} (\phi_{\nu,i}^{(m+1)} - \bar{\phi}_{\nu,i}^{[m]})^2 \right] \end{aligned}$$

Then the first term inside the bracket is minimised at $\nu = \nu^*$. In addition, since we assume that, $\bar{\phi}_{\nu^{**},i}^{[m]} \approx \bar{\phi}_{\nu^*,i}^{[m]}$, where ν^{**} denote the optimal ordering of the $m+1$ samples. That is, since we assume that the component means do not change much at successive iterations, we can minimise the second term by minimising $(\phi_{\nu,i}^{(m+1)} - \bar{\phi}_{\nu^*,i}^{[m]})^2$. Consequently, to minimize $V_{m+1}(\phi_{\nu}^{[m+1]})$, we only need to minimize the variance with respect to the permutations of the vector $\phi^{(m+1)}$.

Chapter 6

Conclusion and future work

Motivated by the use of spatial mixture models in PET image analyses, the statistical estimation aspects of the modelling were developed in this thesis, including some solutions to the normalizing constant problem and label switching in MCMC.

The Bayesian spatial mixture model was employed to estimate kinetic parameters in compartmental model of the myocardium. Our results suggested that Bayesian inference can provide more robust estimations than the conventional methods. In addition, Bayesian inference naturally provided uncertainty estimations for the parameters. The uncertainty estimations are particularly important due to the extremely noisy nature of the data. The spatial dependence between voxels was incorporated by employing the Potts model as the prior in the spatial mixture model where TDI was utilized to solve the inferential problems related to the spatial correlation.

To deal with large sized Potts models, existing methods can either be computationally expensive (such as TDI and exact sampling methods) or have restrictive assumptions (such as PL). Therefore, RCoDA and MCAPCD were proposed to overcome the normalizing constant problem. RCoDA balanced computational efficiency and inferential accuracy. In other words, RCoDA achieved good inferential results without losing much computational efficiency compared to PL. In MCAPCD, the in-

tractability of the Potts model was tackled by decomposing the intractable density function of Potts model into a series of conditional distributions which were then approximated by Monte Carlo simulations of the corresponding summary statistics. Precomputed look-up tables are needed under this approach and they can be reused in the inference about Potts models of different sizes. It has been shown that MCAPCD is computationally efficient and in the meantime can achieve very accurate inferential outcomes.

The label switching problem has been commonly encountered in the inference of mixture models. Various relabelling algorithms were reviewed and their scalability was evaluated with respect to different factors in mixture models, such as the number of observation N, the number of clusters k and q in the Potts models. In order to deal with label switching problem, an algorithm which is based on a loss function interpretation was suggested. The proposed algorithm can scale up well with N and q.

The RCoDA and MCAPCD methods were developed for the Potts models which were widely used in the spatial mixture models. Only the basic forms of the Potts models were demonstrated in this thesis. However, there are many other forms of the Potts models, such as the autologistic models and the Potts models with external fields. Multivariate parameterization of the Potts models can be used to model more complicated data. In the future, our work will focus on generalization to other forms of the Potts models, as well as other types of MRFs, such as Gaussian Markov random fields. In addition, the assumptions on the form of decay in RCoDA can be relaxed. More flexible forms can be employed to capture the change of spatial dependence in each split.

References

Abend, K., T. Harley, and L. N. Kanal (1965). Classification of binary random patterns. *Information Theory, IEEE Transactions on 11*(4), 538–544.

Ahmed, M., S. Yamany, N. Mohamed, A. Farag, and T. Moriarty (2002). A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data. *Medical Imaging, IEEE Transactions on 21*(3), 193 –199.

Aizenman, M., J. Chayes, L. Chayes, and C. Newman (1988). Discontinuity of the magnetization in one-dimensional $1/|x-y|^2$ Ising and Potts models. *Journal of Statistical Physics 50*(1-2), 1–40.

Alessio, A. and P. Kinahan (2006). PET image reconstruction. In R. E. Henkin (Ed.), *Nuclear medicine*, Volume 2. Elsevier, Philadelphia, USA.

Alexander, C. (2004). Normal mixture diffusion with uncertain volatility: Modelling short-and long-term smile effects. *Journal of Banking & Finance 28*(12), 2957–2980.

Alpert, N., Y.-H. D. Fang, and G. El Fakhri (2012). Single-scan rest/stress imaging 18f-labeled flow tracers. *Medical physics 39*(11), 6609–6620.

Aristophanous, M., B. C. Penney, M. K. Martel, and C. A. Pelizzari (2007). A Gaussian mixture model for definition of lung tumor volumes in positron emission tomography. *Medical Physics 34*(11), 4223.

Ashburner, J. and K. J. Friston (2005). Unified segmentation. *NeuroImage 26*(3),

839–851.

Ashburner, J., J. Haslam, C. Taylor, V. J. Cunningham, and T. Jones (1996). A cluster analysis approach for the characterization of dynamic PET data. In B. D. Myers Ralph, Cunningham Vin and J. Terry (Eds.), *Quantification of Brain Function using PET*, Chapter 59. Academic Press, San Diego.

Attias, H. (2000). A variational Bayesian framework for graphical models. *Advances in neural information processing systems 12*(1-2), 209–215.

Banfield, J. D. and A. E. Raftery (1993). Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics 49*(3), pp. 803–821.

Barkema, G. and J. de Boer (1991). Numerical study of phase transitions in Potts models. *Physical Review A 44*(12), 8000–8005.

Bartolucci, F. and J. Besag (2002). A recursive algorithm for Markov random fields. *Biometrika 89*(3), 724–730.

Beaumont, M. A. (2010). Approximate Bayesian Computation in Evolution and Ecology. *Annual review of ecology, evolution, and systematics 41*, 379–406.

Beaumont, M. A., W. Zhang, and D. J. Balding (2002). Approximate Bayesian Computation in Population Genetics. *Genetics 162*(4), 2025–2035.

Belhassen, S. and H. Zaidi (2010). A novel fuzzy C-means algorithm for unsupervised heterogeneous tumor quantification in PET. *Medical Physics 37*(3), 1309.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological) 36*(2), 192–236.

Besag, J. (1975). Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society. Series D (The Statistician) 24*(3), 179–195.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological) 48*(3), 259–302.

Bezdek, J., R. Hathaway, M. Sobin, and W. Tucker (1987). Convergence theory for fuzzy C-means: counterexamples and repairs. *IEEE Trans. Syst. Man Cybern. 17*(5), 873–877.

Biernacki, C., G. Celeux, and G. Govaert (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence 22*(7), 719–725.

Boudraa, A. E., J. Champier, L. Cinotti, J. C. Bordet, F. Lavenne, and J. J. Mallet (1996). Delineation and quantitation of brain lesions by fuzzy clustering in positron emission tomography. *Comput Med Imaging Graph 20*(1), 31–41.

Brazey, D. and B. Portier (2014). A new spherical mixture model for head detection in depth images. *SIAM Journal on Imaging Sciences 7*(4), 2423–2447.

Brigo, D. and F. Mercurio (2002). Lognormal-mixture dynamics and calibration to market volatility smiles. *International Journal of Theoretical and Applied Finance 5*(04), 427–446.

Brodatz, P. (1966). *Textures: a photographic album for artists and designers*. Dover pictorial archives. New York: Dover Publications.

Cardy, J. L. (1986). Effect of boundary conditions on the operator content of two-dimensional conformally invariant theories. *Nuclear Physics B 275*(2), 200–218.

Celeux, G. (1998). Bayesian inference for mixtures: the label switching problem. In R. Payne and P. Green (Eds.), *Proceedings of XIII symposium on computational statistics*, pp. 227–232. Physica-Verlag.

Celeux, G., F. Forbes, and N. Peyrard (2003). EM procedures using mean field-like approximations for Markov model-based image segmentation. *Pattern recognition 36*(1), 131–144.

Celeux, G. and G. Govaert (1995). Gaussian parsimonious clustering models. *Pattern Recognition 28*(5), 781 – 793.

Celeux, G., M. Hurn, and C. P. Robert (2000). Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association 95*, 957 – 970.

Cerqueira, M. D., N. J. Weissman, V. Dilsizian, A. K. Jacobs, S. Kaul, W. K. Laskey, D. J. Pennell, J. A. Rumberger, T. Ryan, M. S. Verani, et al. (2002). Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart a statement for healthcare professionals from the cardiac imaging committee of the council on clinical cardiology of the american heart association. *Circulation 105*(4), 539–542.

Chang, S. C. and R. Shrock (2015). Exact partition functions for the q -state potts model with a generalized magnetic field on lattice strip graphs. *Journal of Statistical Physics 161*(4), 915–932.

Chen, S. C. and D. Q. Zhang (2004). Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics 34*(4), 1907–1916.

Chen, W., M. L. Giger, and U. Bick (2006). A fuzzy C-means (FCM)-based approach for computerized segmentation of breast lesions in dynamic contrast-enhanced MR Images. *Academic radiology 13*(1), 63–72.

Chuang, K.-S., H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen (2006). Fuzzy C-means clustering with spatial information for image segmentation. *Computerized Medical Imaging and Graphics 30*(1), 9–15.

Clark, M. C., L. O. Hall, D. B. Goldgof, L. P. Clarke, R. P. Velthuizen, and M. S. Silbiger (1994). MRI segmentation using fuzzy clustering techniques. *Engineering in Medicine and Biology Magazine, IEEE 13*(5), 730–742.

Coke, G. and M. Tsao (2010). Random effects mixture models for clustering electrical load series. *Journal of Time Series Analysis 31*(6), 451–464.

Cressie, N. and J. L. Davidson (1998). Image analysis with partially ordered Markov models. *Computational statistics & data analysis 29*(1), 1–26.

Cressie, N. A. C. (1993). *Statistics for spatial data*, Volume 298 of *Wiley series in probability and mathematical statistics: Applied probability and statistics*. New York: J. Wiley.

Cron, A. J. and M. West (2011). Efficient classification-based relabeling in mixture models. *The American Statistician 65*(1), 16–20.

DiCiccio, T. J., R. E. Kass, A. Raftery, and L. Wasserman (1997). Computing Bayes factors by combining simulation and asymptotic approximations. *Journal of the American Statistical Association 92*(439), 903–915.

Dunn, J. C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics 3*(3), 32–57.

El Fakhri, G., A. Sitek, B. Guérin, M. F. Kijewski, M. F. Di Carli, and S. C. Moore (2005). Quantitative dynamic cardiac 82Rb PET using generalized factor and compartment analyses. *Journal of nuclear medicine 46*(8), 1264–1271.

El Fakhri, G., A. Sitek, R. E. Zimmerman, and J. Ouyang (2006). Generalized five-dimensional dynamic and spectral factor analysis. *Medical physics 33*(4), 1016–1024.

Evans, M. and T. Swartz (1995). Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems. *Statistical Science 10*(3), 254–272.

Everitt, R. G. (2012). Bayesian parameter estimation for latent Markov random fields and social networks. *Journal of Computational and graphical Statistics 21*(4), 940–960.

Feng, D. (2008). *Bayesian hidden Markov normal mixture models with application to MRI tissue classification*. Ph. D. thesis, University of Iowa.

Feng, D., L. Tierney, and V. Magnotta (2012). MRI tissue classification using high-resolution Bayesian hidden Markov normal mixture models. *Journal of the American Statistical Association 107*(497), 102–119.

Fernandez, C. and P. J. Green (2002). Modelling spatialy correlated data via mixtures: A Bayesian approach. *Journal of Royal Statistical Society B 64*(4), 805–826.

Fisher, M. E. and M. N. Barber (1972). Scaling theory for finite-size effects in the critical region. *Physical Review Letters 28*(23), 1516.

Formisano, E., F. De Martino, and G. Valente (2008). Multivariate analysis of fmri time series: classification and regression of brain responses using machine learning. *Magnetic Resonance Imaging 26*(7), 921–934.

Friel, N., A. Pettitt, R. Reeves, and E. Wit (2009). Bayesian inference in hidden Markov random fields for binary data defined on large lattices. *Journal of Computational and Graphical Statistics 18*(2), 243–261.

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov switching models* (1 ed.). Springer Series in Statistics. Springer-Verlag New York.

Frühwirth-Schnatter, S. (2011). Dealing with label switching under model uncertainty. In K. L. Mengersen, C. P. Robert, and D. M. Titterington (Eds.), *Mixtures: Estimation and Applications*. Wiley.

Gaffney, S. and P. Smyth (1999). Trajectory clustering with mixtures of regression models. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, USA, pp. 63–72. ACM.

Gaffney, S. and P. Smyth (2003). Curve clustering with random effects regression mixtures. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*.

Gelman, A. and X.-L. Meng (1998). Simulating normalizing constants: from

importance sampling to bridge sampling to path sampling. *Statistical science 13*(2), 163–185.

Gelman, A. and D. B. Rubin (1992). Inference from iterative simulation using multiple sequences. *Statistical science 7*(4), 457–472.

Geman, S. and D. Geman (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (6), 721–741.

Geyer, C. J. and E. A. Thompson (1992). Constrained Monte Carlo maximum likelihood for dependent data (with discussion). *Journal of the Royal Statistical Society. Series B. Methodological 54*(3), 657–699.

Gordon, C. L., C. E. Webber, J. D. Adachi, and N. Christoforou (1996). In vivo assessment of trabecular bone structure at the distal radius from high-resolution computed tomography images. *Physics in medicine and biology 41*(3), 495.

Green, P. J. and S. Richardson (2002). Hidden Markov models and disease mapping. *Journal of the American statistical association 97*(460), 1055–1070.

Grelaud, A., C. P. Robert, J.-M. Marin, F. Rodolphe, J.-F. Taly, et al. (2009). ABC likelihood-free methods for model choice in Gibbs random fields. *Bayesian Analysis 4*(2), 317–335.

Grün, B. and F. Leisch (2009). Dealing with label switching in mixture models under genuine multimodality. *Journal of multivariate analysis 100*, 851–861.

Gu, M. G. and H.-T. Zhu (2001). Maximum likelihood estimation for spatial models by Markov chain Monte Carlo stochastic approximation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*(2), 339–355.

Guillot, G., A. Estoup, F. Mortier, and J. F. Cosson (2005). A spatial statistical model for landscape genetics. *Genetics 170*(3), 1261–1280.

Gunn, R. N., A. A. Lammertsma, S. P. Hume, and V. J. Cunningham (1997). Parametric imaging of ligand-receptor binding in PET using a simplified reference region model. *Neuroimage 6*(4), 279–287.

Haindl, M., V. Remeš, and V. Havlíček (2012). Potts compound markovian texture model. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 29–32. IEEE.

Hartvig, N. V. and J. L. Jensen (2000). Spatial mixture modeling of fMRI data. *Human Brain Mapping 11*(4), 233–248.

Hatt, M., C. Cheze le Rest, A. Turzo, C. Roux, and D. Visvikis (2009). A fuzzy locally adaptive Bayesian segmentation approach for volume determination in PET. *IEEE Trans Med Imaging 28*(6), 881–93.

Held, K., E. R. Kops, B. J. Krause, W. M. Wells III, R. Kikinis, and H.-W. Muller-Gartner (1997). Markov random field segmentation of brain MR images. *Medical Imaging, IEEE Transactions on 16*(6), 878–886.

Hongler, C. and K. Kytölä (2013). Ising interfaces and free boundary conditions. *Journal of the American Mathematical Society 26*(4), 1107–1189.

Huang, S.-C. and Y. Zhou (1998). Spatially-coordinated regression for image-wise model fitting to dynamic PET data for generating parametric images. *Nuclear Science, IEEE Transactions on 45*(3), 1194–1199.

Huang, Y., K. B. Englehart, B. Hudgins, and A. D. Chan (2005). A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses. *Biomedical Engineering, IEEE Transactions on 52*(11), 1801–1811.

Hudson, H. M. and R. S. Larkin (1994). Accelerated image reconstruction using ordered subsets of projection data. *Medical Imaging, IEEE Transactions on 13*(4), 601–609.

Hurn, M., A. Justel, and C. P. Robert (2003). Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics 12*(1), 55–79.

Hurn, M. A., O. K. Husby, and H. Rue (2003). A tutorial on image analysis. In J. Møller (Ed.), *Spatial statistics and computational methods*, Volume 173 of *Lecture Notes in Statistics*, pp. 87–141. New York: Springer.

Ibáñez, M. V. and A. Simó (2003). Parameter estimation in Markov random field image modeling with imperfect observations. A comparative study. *Pattern recognition letters 24*(14), 2377–2389.

James, G. M. and C. A. Sugar (2003). Clustering for sparsely sampled functional data. *Journal of the American Statistical Association 98*, 397–408.

Janssen, M. H., H. J. Aerts, M. C. Ollers, G. Bosmans, J. A. Lee, J. Buijsen, D. De Ruysscher, P. Lambin, G. Lammering, and A. L. Dekker (2009). Tumor delineation based on time activity curve differences assessed with dynamic fluorodeoxyglucose positron emission tomography computed tomography. *Int J Radiat Oncol Biol Phys 73*(2), 456–65.

Jasra, A. (2005). *Bayesian inference for mixture models via Monte Carlo*. Ph. D. thesis, Imperial College London.

Jasra, A., C. C. Holmes, and D. A. Stephens (2005). Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modelling. *Statistical Science 20*(1), 50–67.

Jiang, W. Z. T. (2004). Automation segmentatio of PET image for brain tumors. *Nuclear Science Symposium Conference Record*.

Jiechang Wen, Dan Zhang, Y.-m. C. H. L. X. Y. (2012). A batch rival penalized expectation-maximization algorithm for gaussian mixture clustering with automatic model selection. *Computational & Mathematical Methods in Medicine 2012*(1), 94–106.

KaewTraKulPong, P. and R. Bowden (2002). An improved adaptive background mixture model for real-time tracking with shadow detection. In P. Remagnino, G. A. Jones, N. Paragios, and C. S. Regazzoni (Eds.), *Video-based surveillance systems*, pp. 135–144. Springer US.

Kamasak, M. E., C. A. Bouman, E. D. Morris, and K. Sauer (2005). Direct reconstruction of kinetic parameter images from dynamic PET data. *Medical Imaging, IEEE Transactions on 24*(5), 636–650.

Kimura, Y., H. Hsu, H. Toyama, M. Senda, and N. M. Alpert (1999). Improved signal-to-noise ratio in parametric images by cluster analysis. *Neuroimage 9*(5), 554–61.

Knorr-Held, L. and H. Rue (2002). On block updating in Markov random field models for disease mapping. *Scandinavian Journal of Statistics 29*(4), 597–614.

Kosterlitz, J. (1974). The critical properties of the two-dimensional xy model. *Journal of Physics C: Solid State Physics 7*(6), 1046–1060.

Lammertsma, A. A. and S. P. Hume (1996). Simplified reference tissue model for pet receptor studies. *Neuroimage 4*(3), 153–158.

Lawson, A. B. and A. Clark (2002). Spatial mixture relative risk models applied to disease mapping. *Statistics in medicine 21*(3), 359–370.

Leahy, R. M. and J. Qi (2000). Statistical approaches in quantitative positron emission tomography. *Statistics and Computing 10*(2), 147–165.

Li, S. Z. (2012). *Markov random field modeling in computer vision* (1 ed.). Computer Science Workbench. Springer Japan.

Li, S. Z. and S. Singh (2009). *Markov random field modeling in image analysis* (3 ed.). Advances in Computer Vision and Pattern Recognition. London: Springer-Verlag London.

Liang, F. (2007). Continuous contour Monte Carlo for marginal density estimation with an application to a spatial statistical model. *Journal of Computational and Graphical Statistics 16*(3), 608–632.

Liang, F. (2010). A double Metropolis-Hastings sampler for spatial models with intractable normalizing constants. *Journal of Statistical Computation and Simulation 80*(9), 1007–1022.

Liang, F. and I. H. Jin (2013). A monte carlo metropolis-hastings algorithm for sampling from distributions with intractable normalizing constants. *Neural Computation 25*(8), 2199–2234.

Liang, F., I. H. Jin, Q. Song, and J. S. Liu (2016). An adaptive exchange algorithm for sampling from distributions with intractable normalising constants. *Journal of the American Statistical Association 111*(513), 377–393.

Lichstein, J. W., T. R. Simons, S. A. Shriner, and K. E. Franzreb (2002). Spatial autocorrelation and autoregressive models in ecology. *Ecological monographs 72*(3), 445–463.

Liew, A. W. and H. Yan (2003). An adaptive spatial fuzzy clustering algorithm for 3-D MR image segmentation. *IEEE Trans Med Imaging 22*(9), 1063–75.

Liew, A. W. C., S. H. Leung, and W. H. Lau (2000). Fuzzy image clustering incorporating spatial continuity. *IEE Proceedings-Vision, Image and Signal Processing 147*(2), 185–192.

Lin, Y., J. Haldar, Q. Li, P. Conti, and R. Leahy (2014, Jan). Sparsity Constrained Mixture Modeling for the Estimation of Kinetic Parameters in Dynamic PET. *Medical Imaging, IEEE Transactions on 33*(1), 173–185.

Lindsay, B. G. (1988). Composite likelihood methods. *Contemporary Mathematics 80*(1), 221–39.

Liptrot, M., K. H. Adams, L. Martiny, L. H. Pinborg, M. N. Lonsdale, N. V. Olsen, S. Holm, C. Svarer, and G. M. Knudsen (2004). Cluster analysis in kinetic modelling of the brain: a noninvasive alternative to arterial sampling. *Neuroimage 21*(2), 483–93.

Liu, X. and M. Rattray (2010). Including probe-level measurement error in robust mixture clustering of replicated microarray gene expression. *Statistical Applications in Genetics & Molecular Biology 9*(9), 1–25.

Luijten, E. and H. W. Blöte (1995). Monte carlo method for spin models with long-range interactions. *International Journal of Modern Physics C 6*(03), 359–370.

L.Wahl, Y. A. M. T. (1999). Head and neck cancer: Detection of recurrence with three-dimensional principal components analysis at dynamic FDG PET. *Radiology 212*(1), 285–290.

Lyne, A.-M., M. Girolami, Y. Atchad'e, H. Strathmann, and D. Simpson (2015). On Russian roulette estimates for Bayesian inference with doubly-intractable likelihoods. *Statistical science 30*(4), 443–467.

Marin, J. M., K. L. Mengersen, and C. P. Robert (2005). Bayesian modelling and inference on mixtures of distributions. In D. Dey and C. R. Rao (Eds.), *Handbook of statistics*. Elsevier.

Marin, J.-M., P. Pudlo, C. P. Robert, and R. J. Ryder (2012). Approximate Bayesian computational methods. *Statistics and Computing 22*(6), 1167–1180.

Martinez-Möller, A., M. Souvatzoglou, G. Delso, R. A. Bundschuh, C. Chefd'hotel, S. I. Ziegler, N. Navab, M. Schwaiger, and S. G. Nekolla (2009). Tissue classification as a potential approach for attenuation correction in whole-body PET/MRI: evaluation with PET/CT data. *Journal of nuclear medicine 50*(4), 520–526.

McGrory, C. A., D. Titterington, R. Reeves, and A. N. Pettitt (2009). Variational

bayes for estimating the parameters of a hidden potts model. *Statistics and Computing 19*(3), 329–340.

McLachlan, G. and D. Peel (2004). *Finite mixture models*. Wiley & Sons.

McLachlan, G. J. and K. E. Basford (1988). *Mixture models. Inference and applications to clustering*. Statistics: Textbooks and Monographs. New York: Dekker, 1988.

McLachlan, G. J., R. Bean, and D. Peel (2002). A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics 18*(3), 413–422.

Medvedovic, M., K. Y. Yeung, and R. E. Bumgarner (2004). Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics 20*(8), 1222–1232.

Mohy-ud Din, H., N. A. Karakatsanis, M. A. Lodge, J. Tang, and A. Rahmim (2014). Parametric myocardial perfusion PET imaging using physiological clustering. In *SPIE Medical Imaging*, Volume 9038, pp. 90380P–90380P. International Society for Optics and Photonics.

Møller, J., A. N. Pettitt, R. Reeves, and K. K. Berthelsen (2006). An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika 93*(2), 451–458.

Monahan, J. F. and D. D. Boos (1992). Proper likelihoods for Bayesian analysis. *Biometrika 79*(2), 271–278.

Moores, M. T., A. N. Pettitt, and K. Mengersen (2015). Scalable Bayesian inference for the inverse temperature of a hidden Potts model. *arXiv preprint arXiv:1503.08066*.

Morris, E. D., C. J. Endres, K. C. Schmidt, B. T. Christian, R. F. Muzic Jr., and R. E. Fisher (2004). Kinetic modelling in positron emission tomography. In M. N. Wernick and J. N. Aarsvold (Eds.), *Emission Tomography*, pp. 499–540. Elsevier,San Diego.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the soceity of industrial and applied mathematics 5*, 32–38.

Murray, I. (2007). *Advances in Markov chain Monte Carlo methods*. Ph. D. thesis, University of Cambridge.

Murray, I., Z. Ghahramani, and D. J. C. MacKay (2006). MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pp. 359–366. AUAI Press.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing 11*(2), 125–139.

Norets, A. (2010). Approximation of conditional densities by smooth mixtures of regressions. *The Annals of statistics 38*(3), 1733–1766.

Norouzi, A., M. S. M. Rahim, A. Altameem, T. Saba, A. E. Rad, A. Rehman, and M. Uddin (2014). Medical image segmentation methods, algorithms, and applications. *IETE Technical Review 31*(3), 199–213.

Nott, D. J. and T. Rydén (1999). Pairwise likelihood methods for inference in image models. *Biometrika 86*(3), 661–676.

Nye, J. A., J. R. Votaw, N. Jarkas, D. Purselle, V. Camp, J. D. Bremner, C. D. Kilts, C. B. Nemeroff, and M. M. Goodman (2008). Compartmental modeling of 11C-HOMADAM binding to the serotonin transporter in the healthy human brain. *Journal of Nuclear Medicine 49*(12), 2018–2025.

O'Hagan, A. and J. J. Forster (2004). *Kendall's Advanced Theory of Statistics, volume 2B: Bayesian Inference, second edition*. Arnold.

Onoma, D. P., S. Ruan, I. Gardin, G. A. Monnehan, R. Modzelewski, and P. Vera (2012). 3D random walk based segmentation for lung tumor delineation in PET imaging. In *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1260–1263. IEEE.

O'Sullivan, F. (1993). Imaging radiotracer model parameters in PET: a mixture analysis approach. *IEEE Trans Med Imaging 12*(3), 399–412.

O'Sullivan, F. and A. Saha (1999). Use of ridge regression for improved estimation of kinetic constants from PET data. *Medical Imaging, IEEE Transactions on 18*(2), 115–125.

Pal, N. R. and S. K. Pal (1993). A review on image segmentation techniques. *Pattern recognition 26*(9), 1277–1294.

Papaspiliopoulos, O. and G. O. Roberts (2008). Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika 95*(1), 169–186.

Papastamoulis, P. (2014). Handling the label switching problem in latent class models via the ECR algorithm. *Communications in Statistics-Simulation and Computation 43*(4), 913–927.

Papastamoulis, P. and G. Iliopoulos (2010). An artificial allocations based solution to the label switching problem in Bayesian analysis of mixtures of distributions. *Journal of Computational and Graphical Statistics 19*(2), 313–331.

Papastamoulis, P. and M. P. Papastamoulis (2013). R Package 'label. switching'.

Pappas, T. N. (1992). An adaptive clustering algorithm for image segmentation. *Signal Processing, IEEE Transactions on 40*(4), 901–914.

Pauli, F., W. Racugno, and L. Ventura (2011). Bayesian composite marginal likelihoods. *Statistica Sinica 21*(1), 149–164.

Pearson, K. (1894). Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A 185*, 71–110.

Pelosi, M., M. Alfò, F. Martella, E. Pappalardo, and A. Musarò (2015). Finite mixture clustering of human tissues with different levels of igf-1 splice variants mrna transcripts. *Bmc Bioinformatics 16*(1), 1–17.

Permuter, H., J. Francos, and I. H. Jermyn (2003). Gaussian mixture models of texture and colour for image database retrieval. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP'03). 2003 IEEE International Conference on*, Volume 3, pp. III–569. IEEE.

Petibon, Y., J. Ouyang, X. Zhu, C. Huang, T. Reese, S. Y. Chun, Q. Li, and G. El Fakhri (2013). Cardiac motion compensation and resolution modeling in simultaneous PET-MR: a cardiac lesion detection study. *Physics in medicine and biology 58*(7), 2085.

Pham, D. L. (2001). Spatial models for fuzzy clustering. *Computer Vision and Image Understanding 84*(2), 285–297.

Pham, D. L. and J. L. Prince (1999). Adaptive fuzzy segmentation of magnetic resonance images. *IEEE Trans Med Imaging 18*(9), 737–52.

Pham, D. L., C. Xu, and J. L. Prince (2000). Current methods in medical image segmentation 1. *Annual review of biomedical engineering 2*(1), 315–337.

Potts, R. B. (1952). Some generalized order-disorder transformations. In *Mathematical proceedings of the cambridge philosophical society*, Volume 48, pp. 106–109. Cambridge Univ Press.

Propp, J. G. and D. B. Wilson (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random structures and Algorithms 9*(1-2), 223–252.

Puolamaki, K. and S. Kaski (2009). Bayesian solutions to the label switching problem. In *Advances in Intelligent Data Analysis VIII*, Volume 5772 of *Lecture Notes in Computer Science*, pp. 381–392. Springer.

Rajapakse, J. C., J. N. Giedd, and J. L. Rapoport (1997). Statistical approach to segmentation of single-channel cerebral MR images. *Medical Imaging, IEEE Transactions on 16*(2), 176–186.

Reeves, R. and A. N. Pettitt (2004). Efficient recursions for general factorisable models. *Biometrika 91*(3), 751–757.

Richardson, S. and P. J. Green (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of Royal Statistical Society B 59*(4), 731–792.

Royle, J. A. (2004). N-mixture models for estimating population size from spatially replicated counts. *Biometrics 60*(1), 108–115.

Saad, A., B. Smith, G. Hamarneh, and T. Möller (2007). Simultaneous segmentation, kinetic parameter estimation, and uncertainty visualisation of dynamic PET images. In N. Ayache, S. Ourselin, and A. Maeder (Eds.), *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, pp. 726–733. Springer, Berlin Heidelberg.

Samé, A., F. Chamroukhi, G. Govaert, and P. Aknin (2011). Model-based clustering and segmentation of time series with changes in regime. *Advances in Data Analysis and Classification 5*(4), 301–321.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics 6*(2), 461–464.

Singleton, H. R. and G. M. Pohost (1997). Automatic cardiac MR image segmentation using edge detection by tissue classification in pixel neighborhoods. *Magnetic resonance in medicine 37*(3), 418–424.

Slifstein, M., B. Kolachana, E. Simpson, P. Tabares, B. Cheng, M. Duvall, W. G. Frankle, D. Weinberger, M. Laruelle, and A. Abi-Dargham (2008). COMT genotype predicts cortical-limbic D1 receptor availability measured with [11C] NNC112 and PET. *Molecular psychiatry 13*(8), 821–827.

Smith, D. and M. Smith (2006). Estimation of binary Markov random fields using Markov chain Monte Carlo. *Journal of Computational and Graphical Statis-*

tics 15(1), 207–227.

Sperrin, M., T. Jaki, and E. Wit (2010). Probabilistic relabelling strategies for the lable switching problem in Bayesian mixture models. *Statistics and Computing 20*, 357–366.

Steele, R. J. and A. E. Raftery (2009). Performance of Bayesian model selection criteria for Gaussian mixture models. In M.-H. Chen, P. Müller, D. Sun, K. Ye, and D. Dey (Eds.), *Frontiers of Statistical Decision Making and Bayesian Analysis*, pp. 113–155. Springer New York.

Stephens, M. (1997a). *Bayesian methods for mixtures of normal distributions*. Ph. D. thesis, University of Oxford.

Stephens, M. (1997b). Discussion on 'on Bayesian analysis of mixtures with an unknown number of components (with discussion)'. *Journal of Royal Statistical Society B 59*(4), 768–769.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the royal statistical society, series B 26*(4), 795–809.

Stigler, S. M. (1986). *The history of statistics: The measurement of uncertainty before 1900*. Harvard University Press.

Sujaritha, M. and S. Annadurai (2011). A new modified gaussian mixture model for color-texture segmentation. *Journal of Computer Science 7*(2), 279–283.

Swendsen, R. H. and J.-S. Wang (1987). Nonuniversal critical dynamics in Monte Carlo simulations. *Physical review letters 58*(2), 86.

Titterington, D. (1997). Mixture distributions (update). *Encyclopedia of statistical sciences*.

Titterington, D. M., A. F. Smith, and U. E. Makov (1985). *Statistical analysis of finite mixture distributions*. Wiley Series in Probability and Statistics - Applied Probability and Statistics Section. Wiley.

Toga, A. W. and J. C. Mazziotta (2002). *Brain mapping: the methods*. Academic press.

Ullah, I., P. Parviainen, and J. Lagergren (2015). Species tree inference using a mixture model. *Molecular Biology and Evolution 32*(9), 2469–2482.

Van Leemput, K., F. Maes, D. Vandermeulen, and P. Suetens (1999). Automated model-based tissue classification of MR images of the brain. *Medical Imaging, IEEE Transactions on 18*(10), 897–908.

Varin, C., N. M. Reid, and D. Firth (2011). An overview of composite likelihood methods. *Statistica Sinica 21*(1), 5–42.

Wang, G. and J. Qi (2009). Generalized algorithms for direct reconstruction of parametric images from dynamic PET data. *Medical Imaging, IEEE Transactions on 28*(11), 1717–1726.

Wang, J., J. Kong, Y. Lu, M. Qi, and B. Zhang (2008). A modified FCM algorithm for MRI brain image segmentation using both local and non-local spatial constraints. *Computerized Medical Imaging and Graphics 32*(8), 685–698.

Watchareeruetai, U., Y. Takeuchi, T. Matsumoto, H. Kudo, and N. Ohnishi (2006). Computer vision based methods for detecting weeds in lawns. *Machine vision and applications 17*, 287–296.

Weiss, Y. and E. H. Adelson (1996). A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pp. 321–326. IEEE.

Wernick, J. G. B. P. G. Y. N. (2003). Segmentation of dynamic PET or fMRI images based on a similarity metric. *IEEE TRANSACTIONS ON NUCLEAR SCIENCE 50*(5), 5.

White, S. R., T. Kypraios, and S. P. Preston (2015). Piecewise approximate bayesian computation: fast inference for discretely observed markov models using a

factorised posterior distribution. *Statistics & Computing 25*(2), 289–301.

Wilkinson, D. J. (2005). Parallel Bayesian computation. In E. J. Kontoghiorghes (Ed.), *Handbook of Parallel Computing and Statistics*, Volume 184 of *Statistics: A Series of Textbooks and Monographs*, Chapter 16. Chapman and Hall/CRC.

Winkler, G. (2003). *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction*, Volume 27 of *Stochastic Modelling and Applied Probability*. Verlag Berlin Heidelberg: Springer.

Wolff, U. (1989). Collective Monte Carlo updating for spin systems. *Physical Review Letters 62*(4), 361.

Wong, K. P., D. G. Feng, S. R. Meikle, and M. J. Fulham (2002). Segmentation of dynamic PET images using cluster analysis. *IEEE Transactions on Nuclear Science 49*(1), 200–207.

Wood, S. N. (2010). Statistical inference for noisy nonlinear ecological dynamic systems. *Nature 466*(7310), 1102–1104.

Woolrich, M. W. and T. E. Behrens (2006). Variational bayes inference of spatial mixture models for segmentation. *Medical Imaging, IEEE Transactions on 25*(10), 1380–1391.

Woolrich, M. W., T. E. Behrens, C. F. Beckmann, and S. M. Smith (2005). Mixture models with adaptive spatial regularization for segmentation with an application to fMRI data. *Medical Imaging, IEEE Transactions on 24*(1), 1–11.

W.Segars (2000). *Development of a new dynamic NURBS-based cardiac torso (NCAT) phantom*. Ph. D. thesis, University of North Carolina.

Wu, F.-Y. (1982). The Potts model. *Reviews of modern physics 54*(1), 235–268.

Xu, D. and J. Knight (2013). Stochastic volatility model under a discrete mixture-of-normal specification. *Journal of Economics & Finance 37*(2), 216–239.

Yao, W. (2012). Model based labeling for mixture models. *Statistics and Computing 22*(2), 337 – 347.

Yao, W. and L. Li (2014). An online Bayesian mixture labeling method by minimizing deviance of classification probabilities to reference labels. *Journal of statistical computation and simulation 84*(2), 310–323.

Yao, W. and B. G. Lindsay (2009). Bayesian mixture labeling by highest posterior density. *Journal of the American Statistical Association 104*, 758 – 767.

Zaidi, H., M. Diaz-Gomez, A. Boudraa, and D. O. Slosman (2002). Fuzzy clustering-based segmented attenuation correction in whole-body PET imaging. *Phys Med Biol 47*(7), 1143–60.

Zhou, Y., J. A. D. Aston, and A. M. Johansen (2013). Bayesian model comparison for compartmental models with applications in positron emission tomography. *Journal of Applied Statistics 40*, 993–1016.

Zhu, D. (2016). A two-component mixture model for density estimation and classification. *Journal of Interdisciplinary Mathematics 19*(2), 311–319.

Zhu, W. and Y. Fan (2016). Relabelling algorithms for mixture models with applications for large data sets. *Journal of Statistical Computation and Simulation 86*(2), 394–413.

Zhu, W., J. Ouyang, Y. Rakvongthai, N. Guehl, D. Wooten, G. El Fakhri, M. Normandin, and Y. Fan (2016). A Bayesian spatial temporal mixtures approach to kinetic parametric images in dynamic positron emission tomography. *Medical physics 43*(3), 1222–1234.