



# A synthetic likelihood approach for intractable markov random fields

Wanchuang Zhu<sup>1,2</sup> · Yanan Fan<sup>3,4</sup>

Received: 2 September 2021 / Accepted: 17 June 2022 / Published online: 9 July 2022  
© The Author(s) 2022

## Abstract

We propose a new scalable method to approximate the intractable likelihood of the Potts model. The method decomposes the original likelihood into products of many low-dimensional conditional terms, and a Monte Carlo method is then proposed to approximate each of the small terms using their corresponding (exact) Multinomial distribution. The resulting tractable synthetic likelihood then serves as an approximation to the true likelihood. The method is scalable with respect to lattice size and can also be used for problems with irregular lattices. We provide theoretical justifications for our approach, and carry out extensive simulation studies, which show that our method performs at least as well as existing methods, whilst providing significant computational savings, up to ten times faster than the current fastest method. Finally, we include three real data applications for illustration.

**Keywords** Markov random field · Intractable normalizing constant · Potts model · Synthetic likelihood

---

✉ Wanchuang Zhu  
wanchuang.zhu@sydney.edu.au  
Yanan Fan  
y.fan@unsw.edu.au

<sup>1</sup> School of Mathematics and Statistics, The University of Sydney, Darlington, Sydney 2008, NSW, Australia

<sup>2</sup> Data Analytics for Resources and Environments (DARE), The University of Sydney, Darlington, Sydney 2008, NSW, Australia

<sup>3</sup> School of Mathematics and Statistics, University of New South Wales, Sydney 2052, NSW, Australia

<sup>4</sup> UNSW Data Science Hub, University of New South Wales, Sydney 2052, NSW, Australia

## 1 Introduction

The Potts model, a special type of Markov random fields (MRF) model, is often used to describe spatial dependence over a lattice. As such, it has been used extensively in image analysis problems, see for example Hurn et al. (2003); Pereyra et al. (2013); Pal and Pal (1993); Li and Singh (2009); Storath et al. (2015) and in social networks modelling (Everitt 2012). The Ising model, which is a two state Potts model, is also closely related to the Boltzmann Machine (Hinton 2014), a popular building block in deep learning algorithms (Goodfellow et al. 2016).

It is well known that the normalising constant in the Potts model is intractable even for moderately sized lattices (Zhu and Fan 2018). The considerable amount of statistical literature on how to overcome this problem can be approximately divided into two categories: approximation-based approaches and simulation-based approaches.

Approximation-based approaches aim to transform the intractable likelihood of the Potts model to a tractable form by imposing some assumptions on the model. The Pseudo likelihood (PL) method of Besag (1975) is one of the earliest attempts. PL approximates the intractable Potts likelihood by the product of full conditional probabilities. Under this approach, the normalizing constant is then trivial to compute. However, PL no longer a proper likelihood function, but can be considered a type of composite likelihood (Varin et al. 2011), and has been documented to be unstable when dependence is strong (Geyer and Thompson 1992). The method is popular in practice, particularly for large lattices due to its concise form. Cressie and Davidson (1998) proposed a partially ordered Markov models, taking advantage of the structure to avoid the need to compute the normalizing constant. However, the method is only applicable to a subset of MRFs, therefore not flexible enough for use in practice. Reeves and Pettitt (2004) proposed a method for general factorizable models. Although this approach is exact, it is only applicable to small sized lattices. Friel et al. (2009) extended the work of Reeves and Pettitt (2004) to larger lattices by sacrificing some dependencies. The full lattice is divided into many sublattices and the sublattices are assumed to be independent, they call this the reduced dependence approximation (RDA). The authors reported that the method can be efficiently applied to binary MRFs, but concluded that the extension to the  $q$ -state Potts models may not be computationally tractable. Another similar idea can be found in Bartolucci and Besag (2002), where a recursive algorithm using the product of conditional probabilities was presented, but again their method is only applicable to small lattices. More recently, Zhu and Fan (2018) proposed a method (RCoDA) that decomposes the large lattice into a sequence of nested subsets of smaller lattices, and used a sequence of much smaller Potts models to approximate each sublattice. The method scales well with the sizes of lattices, but can exhibit similar biases as those observed from the pseudo-likelihood model.

Simulation-based approaches calculate the intractable likelihood via simulation of the Potts models. Gelman and Meng (1998) proposed path sampling to approximate ratio of normalizing constants. Green and Richardson (2002)

advocated the thermodynamic integration (TDI) approach, which relies on Monte Carlo simulations. This approach computes a look-up table offline and can be used with many different posterior computation methods. Some other similar simulation-based methods can be found in Geyer and Thompson (1992); Gu and Zhu (2001); Liang (2007) and references therein. Another group of methods rely on generations of auxiliary variables. Such methods include Møller et al. (2006); Murray (2006, 2007); Liang (2010); Liang et al. (2016). As perfect simulation (Propp and Wilson 1996) step is usually required, these methods are computationally very expensive. Liang et al. (2016) extended the exchange algorithm of Murray (2006) to overcome the issue of obtaining perfect samples, using an importance sampling procedure coupled with a Markov chain running in parallel. Grelaud et al. (2009) adopted approximate Bayesian computation (ABC) to tackle the intractable normalizing constant. The method was established on the fact that the sufficient statistics of the Potts model can be readily computed. Atchadé and Liu (2010) adopted Wang-Landau algorithm (Wang and Landau 2001) to estimate the normalizing constant. Everitt (2012) developed a sequential Monte Carlo ABC method to deal with the same issue. For model selection, Moores et al. (2015) adopted ABC as a pre-process step to fit the function between summary statistics and spatial dependence parameter and then employed this function to learn the parameter of interest. Furthermore, Moores et al. (2020) proposed an indirect likelihood approach based on an approximation of the score function.

Often in applications of MRFs, particularly those found in image analyses, the size of the random field can be extremely large, the rows and columns of the lattices can sometimes be in the order of thousands (Moores et al. 2020). Generally speaking, simulation-based approaches tend to be less scalable than approximation-based approaches since simulation of large Potts models is time-consuming. Among the approximation-based methods, according to our experience, PL (Besag 1975) and RCoDA (Zhu and Fan 2018) are the most scalable methods with respect to lattice size. However, both methods can suffer from bias and/or coverage issues, see Zhu and Fan (2018) for further details.

In this article, we are aiming to propose an approach which is capable of handling very large lattices, and is also extendable to irregular lattices. We directly approximate the density function of the Potts model using a *synthetic likelihood*. All the components of the synthetic likelihood function, taken as the density function of a Multinomial distribution, are then obtained using Monte Carlo simulations. The simulation component in our approach is much less time consuming than existing simulation-based approaches. Our method integrates strong points of both approximation and simulation based approaches, and it is both scalable and extendable to non-regular lattices which are important for real applications. Our procedure also shares some connections with the literature on approximate Bayesian computation (Sisson et al. 2018), see for example Fan et al. (2014) and Grazian and Fan (2019), where the intractable likelihoods have been replaced by flexible density approximations. The Matlab code for our method is available at Github.

The paper is structured as follows. Section 2 presents the background on the Potts model and the proposed synthetic likelihood approximation. Section 3 provides theoretical results for the proposed method, such as unbiasedness. Section 4 performs

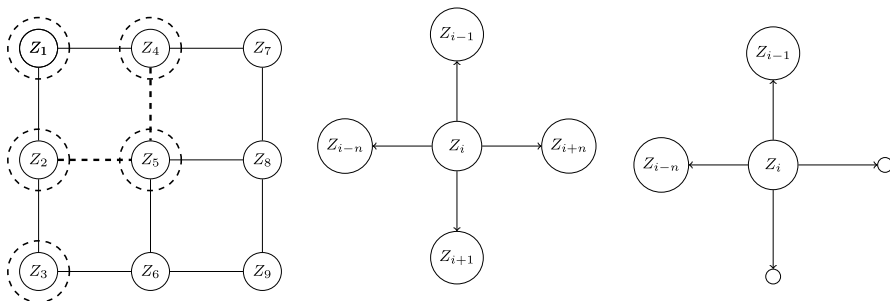
simulation studies in various scenarios, both small lattice and large lattice under different neighbourhood structures. Three real examples are demonstrated in Sect. 5 and we conclude with some discussions and summary in Sect. 6.

## 2 Synthetic likelihood with piecewise conditional distribution

The Potts model plays an important role in modelling spatially correlated data, it is widely used to describe spatial dependencies defined on a lattice. A Potts model is usually defined on a regular lattice. For example, in texture analysis (Bharati et al. 2004), a photograph is analysed to extract its texture. We demonstrate our method in the context of square lattice, such as the left panel in Fig. 1. However, our method is not restricted to this type of lattices. Let  $\mathbf{Z} = \{Z_i\}_{i=1}^{n^2}$  denote a set of hidden (or auxiliary) variables, the prior distribution  $\pi(\mathbf{Z} | \beta)$  is specified as the Potts model parameterized by  $\beta \geq 0$ , where the parameter  $\beta$  controls the strength of spatial correlation, higher values of  $\beta$  indicate stronger correlation. The  $Z_i$ s are discrete, and in a  $q$ -state Potts model, they take values from the set  $\{1, 2, \dots, q\}$ . Specifically, the likelihood of the Potts model  $\pi(\mathbf{Z} | \beta)$  is given as,

$$\pi(\mathbf{Z} | \beta) = \frac{1}{\mathcal{C}(\beta)} \exp \left\{ \beta \sum_{i \sim j} I(Z_i = Z_j) \right\}, \tag{1}$$

where  $i \sim j$  indicates that  $i$  and  $j$  are neighbours in some pre-defined neighbourhood structure,  $\beta \sum_{i \sim j} I(Z_i = Z_j)$  is the energy function 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 indicator function where  $I(Z_i = Z_j) = 1$  if  $Z_i = Z_j$ , otherwise  $I(Z_i = Z_j) = 0$ . The normalizing constant  $\mathcal{C}(\beta)$  is a function with respect to  $\beta$ , and therefore cannot be ignored in the Bayesian analyses. The calculation of normalizing constant  $\mathcal{C}(\beta)$  involves a summation over all possible values of  $\mathbf{Z}$ . For a  $q$ -state Potts model defined on a  $n \times n$  lattice, the number of possible realizations of  $\mathbf{Z}$  is  $q^{n^2}$ . Thus even for a moderate sized lattice, this number becomes too large to enumerate over. Wu (1982) provides more details on the properties of the Potts model.



**Fig. 1** Left: vectorization of  $\mathbf{Z} = (Z_1, \dots, Z_9)$  by column for a  $3 \times 3$  lattice. The dashed circles correspond to the structure of  $\pi(Z_i | Z_{1:(i-1)}, \beta)$ , and the dashed lines show the dependence structure of  $\pi(Z_i | Z_{1:(i-1)} \cap \partial i), \beta)$  for the first order neighbourhood,  $i = 5$ . Middle: the first order neighbourhood dependence structure for general  $Z_i$ . Right: partial conditional dependence of  $Z_i$  given  $Z_{i-1}$  and  $Z_{i-n}$

The Potts model can be employed as a hidden state model to model spatial correlation (Feng 2008), which is governed by the inverse temperature  $\beta$ . The spatial correlation in the Potts model is manifested via the form of Eq. (1). That is, the pixel’s value is conditionally independent given the values of pixels in its neighborhood. Under the setting of hidden state model, the posterior distribution of a  $q$ -component Potts spatial mixture model takes the form

$$\pi(\mathbf{Z}, \beta, \boldsymbol{\theta} \mid \mathbf{y}) \propto \left( \prod_{i=1}^{n^2} \pi(y_i \mid \boldsymbol{\theta}, Z_i) \right) \pi(\mathbf{Z} \mid \beta) \pi(\beta) \pi(\boldsymbol{\theta}), \tag{2}$$

where  $\pi(y_i \mid \boldsymbol{\theta}, Z_i)$  is the component distribution for the observed data  $y_i$  conditional on the model parameters  $\boldsymbol{\theta}$  and  $Z_i$ ,  $\pi(\boldsymbol{\theta})$  is the prior for the model parameters and  $\pi(\beta)$  denotes the prior for the parameters of the Potts model, often referred to as a hyper-prior. Equations of the form in (2) are often used in applications where the goal is to cluster or segment several (or  $q$ ) pieces of an image, and the component distribution can take the form of a Gaussian density. In some applications  $\mathbf{Z}$  may also be directly observed. In these cases, the posterior distribution will become

$$\pi(\beta \mid \mathbf{Z}) \propto \pi(\mathbf{Z} \mid \beta) \pi(\beta). \tag{3}$$

The goal is to make inference for the inverse temperature parameter  $\beta$ , which controls the degree of spatial dependence.

In both cases, the main problem associated with inference using the Potts model is the intractable normalizing constant  $\mathcal{C}(\beta)$  in the likelihood function. The intractability of the normalizing constant causes difficulties for the inference on  $\beta$ . Algorithm 1 provides a Markov chain Monte Carlo (MCMC) algorithm to sample from the posterior distribution of  $\beta$  in the Potts model. In step 3, two normalizing constants are involved in the calculation of  $r$ . The intractability of the normalizing constant leads to unknown  $r$  resulting in difficulties with the MCMC procedure. In what follows, we describe an approach involving an approximate likelihood function to overcome this issue.

---

**Algorithm 1** Posterior sampling of  $\beta$  via MCMC

---

- 1: Suppose we are at the  $t^{th}$  iteration of MCMC, and the current state is denoted as  $\beta^{(t)}$ .
  - 2: Let  $t = t + 1$ . Draw a candidate parameter  $\beta^*$  via random walk proposal,  $\beta^* \sim \mathcal{N}(\beta^{(t)}, \sigma^2)$ . If  $\beta^*$  is not in the constraint space of  $\beta$ ,  $\beta^{(t+1)} = \beta^{(t)}$  and repeat Step 2. Otherwise go to 3.
  - 3: Calculate the acceptance probability of  $\beta^*$ :  $r = \min(1, \frac{\pi(\mathbf{Z} \mid \beta^*) \pi(\beta^*)}{\pi(\mathbf{Z} \mid \beta^{(t)}) \pi(\beta^{(t)})})$ .
  - 4: Draw a sample  $u$  from the standard Uniform distribution. If  $r > u$ , let  $\beta^{(t+1)} = \beta^*$ . Otherwise,  $\beta^{(t+1)} = \beta^{(t)}$ .
  - 5: Go to step 2, until termination criteria is satisfied.
-

## 2.1 Decomposing the likelihood function

Consider a regular lattice for  $\mathbf{Z}$ , which is vectorized by column. See an example in Fig. 1 (left panel), where a  $3 \times 3$  lattice is labelled as  $Z_1, \dots, Z_9$  by column. Then in general, for an  $n \times n$  lattice, we have  $\mathbf{Z} = (Z_1, Z_2, \dots, Z_{n^2-1}, Z_{n^2})$  and without loss of generality, write the likelihood function of the Potts model as

$$\pi(\mathbf{Z} | \beta) = \pi(Z_1 | \beta) \prod_{i=2}^{n^2} \pi(Z_i | Z_{1:(i-1)}, \beta), \quad (4)$$

where  $Z_{1:(i-1)}$  denotes the set  $\{Z_1, Z_2, \dots, Z_{i-1}\}$ .

We make two approximations to Eq. (4). Firstly, we propose to omit  $\pi(Z_1 | \beta)$  from the likelihood function  $\pi(\mathbf{Z} | \beta)$ . Although it is possible to approximate  $\pi(Z_1 | \beta)$  using simulation approaches similar to the rest of this paper, it is computationally heavy. As  $n$  increases, the contribution to the likelihood function  $\pi(\mathbf{Z} | \beta)$  is asymptotically approaching 0. Secondly, we use  $\pi(Z_i | (Z_{1:(i-1)} \cap \partial i), \beta)$  to approximate each of the terms  $\pi(Z_i | Z_{1:(i-1)}, \beta)$ , where  $\partial i$  denotes the full neighbourhood of  $Z_i$ . Since we know that the Potts model has a Markovian property where the conditional distribution of  $Z_i$  given  $\{Z_1, \dots, Z_{i-1}, Z_{i+1}, \dots, Z_{n^2}\}$  only depends on its neighbourhood  $\partial i$ . With these changes, we now work with the composite likelihood  $\pi_C(\mathbf{Z} | \beta)$  with the following form,

$$\begin{aligned} \pi_C(\mathbf{Z} | \beta) &\triangleq \prod_{i=2}^{n^2} \pi(Z_i | (Z_{1:(i-1)} \cap \partial i), \beta) \approx \pi(Z_1 | \beta) \prod_{i=2}^{n^2} \pi(Z_i | Z_{1:(i-1)}, \beta) \\ &= \pi(\mathbf{Z} | \beta). \end{aligned} \quad (5)$$

The middle panel in Fig. 1 illustrates the standard first order neighbourhood of  $Z_i$ , which is given by  $\{Z_{i-n}, Z_{i-1}, Z_{i+1}, Z_{i+n}\}$  and the set  $Z_{1:(i-1)} \cap \partial i$  is demonstrated in the right panel of Fig. 1. The full extent of the approximation for each piece centered on  $i$ , is illustrated in the left panel of Fig. 1, for  $i = 5$ , the original dependence on  $Z_1, \dots, Z_4$  reduces to only  $Z_2$  and  $Z_4$  after we take the neighbourhood into consideration. The impact of such an approximation will be discussed in the Sect. 3 where some theoretical properties of the proposed method will be provided.

## 2.2 Approximation using monte carlo simulation

The term  $\pi(Z_i | (Z_{1:(i-1)} \cap \partial i), \beta)$  in the Eq. (5) is typically challenge to compute. In a  $q$ -state Potts model, each  $Z_i$  can take any value  $1, 2, \dots, q$ . It is possible to consider all possible outcomes for the combinations of  $Z_i, Z_{i-n}$  and  $Z_{(i-1)}$ . But as  $q$  gets large or number of conditional item increases, this quickly becomes cumbersome. For example, there are  $q^l$  different combinations of the conditional items given  $l$  conditional items. Fortunately, the sufficient statistics in the Potts model, which is simply the number of pairs with the same values between neighbouring observations, i.e.,  $\sum_{i \sim j} I(Z_i = Z_j)$ , vastly reduces the dimension of the approximation without any loss

of information with respect to the inference of the parameters of interest. Note that this also makes the approximation independent of the value of  $Z_i$ , as the physical value of  $Z_i$  is irrelevant. What matters is whether the neighbouring values are the same or not. For the first order neighbourhood dependence structure, we can use the following

$$\pi_C(\mathbf{Z} \mid \beta) = \prod_{i=2}^{n^2} \pi(Z_i \mid (Z_{1:(i-1)} \cap \partial i), \beta) \propto \prod_{i=2}^{n^2} \pi(S_i \mid Z_{i-1}, Z_{i-n}, \beta). \tag{6}$$

where  $S_i \triangleq I(Z_i = Z_{i-1}) + I(Z_i = Z_{i-n})$  denotes the sufficient statistics whose distribution will be approximated. In the case under the first order neighbourhood, the distribution is conditioned on only two cases:  $Z_{i-1} = Z_{i-n}$  and  $Z_{i-1} \neq Z_{i-n}$ . The sufficient statistics  $S_i$  takes possible values  $\{0, 1, 2\}$ , so we need to approximate  $\pi(S_i = 0 \mid \beta)$ ,  $\pi(S_i = 1 \mid \beta)$  and  $\pi(S_i = 2 \mid \beta)$  under cases where  $Z_{i-1} = Z_{i-n}$  and  $Z_{i-1} \neq Z_{i-n}$  respectively.

The posterior distribution attached to Eq. (6) is given as,

$$\pi_C(\beta \mid \mathbf{Z}) \propto \pi(\beta) \prod_{i=2}^{n^2} \pi(S_i \mid Z_{i-1}, Z_{i-n}, \beta). \tag{7}$$

The distribution of  $S_i$  is clearly a Multinomial distribution with three parameters, the respective probabilities which we denote as  $p_{i1}(\beta)$ ,  $p_{i2}(\beta)$  and  $p_{i3}(\beta)$ , which depends on the value of  $\beta$ . Hereafter we use  $\mathcal{M}(\cdot)$  to denote the Multinomial distribution. These parameters are unknown for now. According to the homogeneity of the Potts model Feng et al. (2012), the parameters can be suppressed to be  $p_1(\beta)$ ,  $p_2(\beta)$  and  $p_3(\beta)$  and the density can be denoted as  $\pi(S \mid Z_{i-1}, Z_{i-n}, \beta)$  for simplicity.

Here we propose to obtain an estimator of  $\pi(S_i \mid Z_{i-1}, Z_{i-n}, \beta)$  via Monte Carlo simulation. For any fixed value of  $\beta$ , an  $n \times n$  Potts model can be simulated using a number of algorithms, see for example the Swendsen and Wang algorithm (Swendsen and Wang 1987) and Wolff’s algorithm (Wolff 1989). The samples  $Z_i, i = 1, \dots, n^2$  and their respective  $\{Z_{i-1}, Z_{i-n}\}$  can then be used to estimate the parameters in the Multinomial distribution. We provide theoretical support for the use of the Multinomial distributions as part of the synthetic likelihood in Theorem 1, where it is shown that as the number of Monte Carlo samples increase, the synthetic likelihood using Multinomial pieces converges pointwise to  $\pi_C(\mathbf{Z} \mid \beta)$ . Together with Theorem 2, we have that  $\pi_C(\mathbf{Z} \mid \beta)$  converges pointwise in  $\beta$ , to the true Potts likelihood for large lattices  $\pi(\mathbf{Z} \mid \beta)$ .

Note that pixels on the boundaries of the lattice have different structures as they are truncated by the boundary, these truncated distributions can be estimated separately in a similar fashion. Algorithm 2 provides detailed description of the Monte Carlo based approach which provides unbiased estimates for  $p_j(\beta), j = 1, \dots, 3$ .

---

**Algorithm 2** Monte Carlo estimation of  $\pi(S \mid Z_{i-1}, Z_{i-n}, \beta)$

---

- 1: Given  $\beta$ , generate  $N$  independent  $n \times n$  lattice realizations  $\mathbf{Z}^1, \dots, \mathbf{Z}^N$  from the Potts model using the Gibbs sampler (see Feng (2008)).
  - 2: Over all the realizations, for each set  $Z_i, Z_{i-1}, Z_{i-n}$  following the dependence structure of Figure 1 (right panel).
    - 2.1 For  $Z_{i-1} = Z_{i-n}$ . Count the number of sets with 0 pairs ( $S = 0$ ), 1 pair ( $S = 1$ ) and 2 pairs ( $S = 2$ ). The normalised proportions form unbiased estimators  $\hat{p}_1^1(\beta), \hat{p}_2^1(\beta), \hat{p}_3^1(\beta)$  where  $S \mid Z_{i-1} = Z_{i-n}, \beta \sim \mathcal{M}(\hat{p}_1^1(\beta), \hat{p}_2^1(\beta), \hat{p}_3^1(\beta))$ .
    - 2.2 For  $Z_{i-1} \neq Z_{i-n}$ , obtain  $\hat{p}_1^2(\beta), \hat{p}_2^2(\beta), \hat{p}_3^2(\beta)$  where  $S \mid Z_{i-1} \neq Z_{i-n}, \beta \sim \mathcal{M}(\hat{p}_1^2(\beta), \hat{p}_2^2(\beta), \hat{p}_3^2(\beta))$ , similarly to Step 2.1.
  - 3: Repeat Steps 1-3 for a dense grid of  $\beta$  values.
- 

When the above procedure is repeated for different  $\beta$ , it will produce a look-up table for a dense grid of  $\beta$  values. Let  $\beta_\epsilon = (\beta_l, \tilde{\beta}_1, \dots, \tilde{\beta}_{M-1}, \beta_u)$  denote the grid of  $\beta$ , where  $\beta_l$  and  $\beta_u$  are the lower and upper bound respectively,  $\epsilon = \frac{\beta_u - \beta_l}{M}$  is the step size of the grid. With the Monte Carlo estimation of  $\pi(S \mid Z_{i-1}, Z_{i-n}, \beta)$ , a grid likelihood function is defined as below, for  $\beta \in [\tilde{\beta}_k - \frac{\epsilon}{2}, \tilde{\beta}_k + \frac{\epsilon}{2}]$

$$\pi_{CG}(\mathbf{Z} \mid \beta) \propto \prod_{i=2}^{n^2} \mathcal{M}(\hat{p}_1(\tilde{\beta}_k), \hat{p}_2(\tilde{\beta}_k), \hat{p}_3(\tilde{\beta}_k) \mid Z_{i-1}, Z_{i-n}). \tag{8}$$

Note that Gibbs sampling rather than perfect sampling is adopted in Step 1 of Algorithm 2. The reason is mainly computational, because perfect sampling is computationally expensive. The posterior distribution for  $\beta$  arising from the Monte Carlo approximation is a step function. We analyse the error introduced to the posterior distribution due to the use of the grid approach in Theorem 2. And note that this error diminishes with  $\mathcal{O}(M^{-2})$  but at the cost of increased computational effort.

Our procedure to estimate the Potts model involves two main ingredients. The first is the breaking down of the large spatial model into smaller, more manageable pieces. The second is the use of data simulation to approximate the distributions of these smaller pieces. The second ingredient in our procedure is similar in spirit to the so-called synthetic likelihood of Wood (2010), or density estimation approximate Bayesian computation (ABC) in the ABC literature (Fan et al. 2014; Papamakarios et al. 2019; Alsing et al. 2019), where either parametric or non-parametric synthetic likelihoods are used in conjunction with Monte Carlo simulations of the data. In our situation, the Multinomial distribution is exact for the distribution of the summary statistics, so a nonparametric approach is not needed. We refer to our approximation procedure as Synthetic Likelihood with Piecewise Conditional Distribution (SLPCD).



### 2.3 Bayesian inference

Given the composite grid likelihood function, posterior distribution of  $\beta$  is derived as,

$$\pi_{CG}(\beta | \mathbf{Z}) \propto \pi_{CG}(\mathbf{Z} | \beta)\pi(\beta). \tag{9}$$

Posterior samples of  $\beta$  can be drawn by Algorithm 3 via MCMC targeting Equation (9).

---

**Algorithm 3** Posterior sampling of  $\beta$  from posterior distribution (9)

---

- 1: Suppose we are at the  $t^{th}$  iteration of MCMC. And the current value is denoted by  $\beta^{(t)}$ .
  - 2: Let  $t = t + 1$ . Draw a candidate parameter  $\beta' \sim \mathcal{N}(\beta^{(t)}, \sigma^2)$ . Find the closest grid value  $\tilde{\beta}_k$  among  $\beta_c$  to  $\beta'$ . Let  $\beta' = \tilde{\beta}_k$ . If  $\beta'$  is not in the constraint space of  $\beta$ ,  $\beta^{(t+1)} = \beta^{(t)}$  and repeat Step 2. Otherwise go to next step.
  - 3: Calculate the acceptance rate of  $\beta'$ :  $r = \min(1, \frac{\pi_{CG}(\mathbf{Z}|\beta')\pi(\beta')}{\pi_{CG}(\mathbf{Z}|\beta^{(t)})\pi(\beta^{(t)})})$ . Both  $\pi_{CG}(\mathbf{Z} | \beta')$  and  $\pi_{CG}(\mathbf{Z} | \beta^{(t)})$  can be computed via Equation (8) with a look-up table containing Monte Carlo estimation of  $p_1(\beta), p_2(\beta), p_3(\beta)$ .
  - 4: Draw a sample  $u$  from standard Uniform distribution. If  $r > u$ , let  $\beta^{(t+1)} = \beta'$ . Otherwise,  $\beta^{(t+1)} = \beta^{(t)}$ .
  - 5: Go to step 2, until termination criteria is satisfied.
- 

We make the conjecture that the individual conditional densities can be estimated from a Potts model of any size. That is, the strength of spatial dependence  $\beta$  is invariant to the size of the lattice, at least locally the subsets  $(Z_{1:(i-1)} \cap \partial i)$  are not effected by observations far away and hence the size of the lattice. As far as we know, the literature does not touch upon this aspect. The theoretical clue for the conjecture is that all the pixels in the lattice are homogeneous regardless the size of the lattice. Therefore, the conditional distribution of  $p(Z_i | \partial i)$  is homogeneous. Moreover, we observe empirically that this appears to be a reasonable assumption. Table 1 shows the estimated Multinomial probabilities  $\hat{\pi}(S | Z_{i-1} = Z_{i-n}, \beta)$  for the Ising model, using three different lattice sizes  $32 \times 32, 128 \times 128$  and  $512 \times 512$ . The differences are negligible between the three lattice sizes. Similarly, Table 2 demonstrates the estimated Multinomial probabilities  $\hat{\pi}(S | Z_{i-1} = Z_{i-n}, \beta)$  for the Potts model with  $q = 4$ . In this and subsequent simulations, we have found that the size of the lattice indeed has little impact on the approximation of  $\pi(S | Z_{i-1}, Z_{i-n}, \beta)$ . Consequently, we only need to simulate the Potts model on small lattices in order to approximate Potts model of any size. This provides a huge computational saving, as there is no need to re-run simulations every time a lattice of a different size is presented. This is a computational advantage compared to the well known method of thermodynamic integration (TDI) proposed in Green and Richardson (2002), which is harder to

**Table 1** The estimations of  $\hat{p}_1^1(\beta), \hat{p}_2^1(\beta), \hat{p}_3^1(\beta)$  in  $\pi(S | Z_{i-1} = Z_{i-n}, \beta)$  for Ising model

$\beta$	32 × 32			128 × 128			512 × 512		
	$\hat{p}_1^1(\beta)$	$\hat{p}_2^1(\beta)$	$\hat{p}_3^1(\beta)$	$\hat{p}_1^1(\beta)$	$\hat{p}_2^1(\beta)$	$\hat{p}_3^1(\beta)$	$\hat{p}_1^1(\beta)$	$\hat{p}_2^1(\beta)$	$\hat{p}_3^1(\beta)$
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	0.376	0.000	0.624
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	0.327	0.000	0.673
0.400	0.302	0.000	0.698	0.303	0.000	0.697	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	0.232	0.000	0.768
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	0.062	0.000	0.938	0.063	0.000	0.937
0.950	0.037	0.000	0.963	0.046	0.000	0.954	0.046	0.000	0.954
1.000	0.030	0.000	0.970	0.035	0.000	0.965	0.035	0.000	0.965

Simulations using different sized lattices, 32 × 32, 128 × 128 and 512 × 512

use in larger lattices because a Monte Carlo estimate of the total number of pairs is needed, requiring repeated simulation of entire lattices.

Actually, the two main ingredients of SLPCD correspond to the two categories of approaches in the literature. The breaking-down step of SLPCD corresponds to composite likelihood approaches. It decomposes the intractable likelihood function into some manageable small pieces. The Monte-Carlo step of SLPCD corresponds to simulation-based approaches. It estimates parameters via Monte Carlo simulation. Therefore, SLPCD combines two categories of approaches into one, and avoids the limitations of each category. That is, SLPCD can be scalable with lattice size because each ingredient of it is scalable with lattice size.

### 2.4 Generalization to the higher order neighbourhood

The SLPCD approach described in previous sections can be generalized to the higher order neighbourhood Potts models. Here we consider the second order neighbourhood structure (Besag 1974; Hurn et al. 2003). Formally, the second order neighbours of  $Z_{i,j}$  includes the eight nearest neighbours, i.e.,  $Z_{i,(j+1)}, Z_{i,(j-1)}, Z_{(i-1),j}, Z_{(i+1),j}, Z_{(i-1),(j-1)}, Z_{(i+1),(j-1)}, Z_{(i-1),(j+1)}$  and  $Z_{(i+1),(j+1)}$ . Figure 2

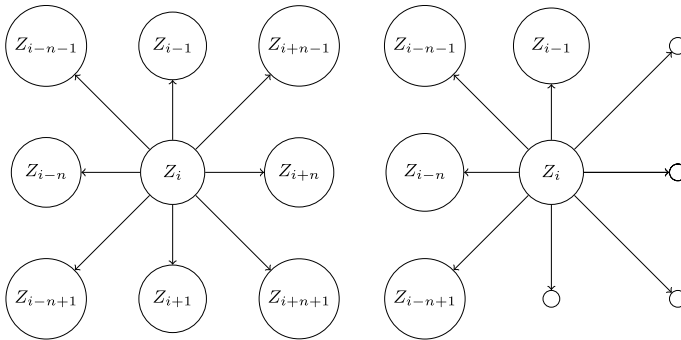
**Table 2** The estimations of  $\hat{p}_1^1(\beta), \hat{p}_2^1(\beta), \hat{p}_3^1(\beta)$  in  $\pi(S | Z_{i-1} = Z_{i-n}, \beta)$  for Potts model with  $q = 4$

$\beta$	$32 \times 32$			$128 \times 128$			$512 \times 512$		
	$\hat{p}_1^1(\beta)$	$\hat{p}_2^1(\beta)$	$\hat{p}_3^1(\beta)$	$\hat{p}_1^1(\beta)$	$\hat{p}_2^1(\beta)$	$\hat{p}_3^1(\beta)$	$\hat{p}_1^1(\beta)$	$\hat{p}_2^1(\beta)$	$\hat{p}_3^1(\beta)$
0.050	0.891	0.000	0.109	0.891	0.000	0.109	0.891	0.000	0.109
0.100	0.880	0.000	0.120	0.880	0.000	0.120	0.880	0.000	0.120
0.150	0.869	0.000	0.131	0.869	0.000	0.131	0.869	0.000	0.131
0.200	0.856	0.000	0.144	0.856	0.000	0.144	0.857	0.000	0.143
0.250	0.844	0.000	0.156	0.844	0.000	0.156	0.843	0.000	0.157
0.300	0.829	0.000	0.171	0.829	0.000	0.171	0.829	0.000	0.171
0.350	0.812	0.000	0.188	0.812	0.000	0.188	0.812	0.000	0.188
0.400	0.795	0.000	0.205	0.795	0.000	0.205	0.795	0.000	0.205
0.450	0.775	0.000	0.225	0.775	0.000	0.225	0.775	0.000	0.225
0.500	0.754	0.000	0.246	0.754	0.000	0.246	0.754	0.000	0.246
0.550	0.731	0.000	0.269	0.731	0.000	0.269	0.731	0.000	0.269
0.600	0.705	0.000	0.295	0.705	0.000	0.295	0.705	0.000	0.295
0.650	0.677	0.000	0.323	0.677	0.000	0.323	0.677	0.000	0.323
0.700	0.646	0.000	0.354	0.646	0.000	0.354	0.646	0.000	0.354
0.750	0.612	0.000	0.388	0.612	0.000	0.388	0.612	0.000	0.388
0.800	0.575	0.000	0.425	0.575	0.000	0.425	0.575	0.000	0.425
0.850	0.534	0.000	0.466	0.534	0.000	0.466	0.534	0.000	0.466
0.900	0.488	0.000	0.512	0.488	0.000	0.512	0.488	0.000	0.512
0.950	0.437	0.000	0.563	0.437	0.000	0.563	0.437	0.000	0.563
1.000	0.378	0.000	0.622	0.378	0.000	0.622	0.378	0.000	0.622

Simulations using different sized lattices,  $32 \times 32$ ,  $128 \times 128$  and  $512 \times 512$

shows the second order neighbourhood structure (the left panel) and the corresponding piecewise conditional dependence structure (the right panel).

The difference between the second order and the first order is that there are four rather than two conditional elements in each conditional dependence structure, which are now  $\{Z_{i-1}, Z_{i-n+1}, Z_{i-n}, Z_{i-n-1}\}$ . This results in more conditional distributions to estimate. For example, we need to consider when all four dependent observations are the same, or when two of them are the same and the other two are the same as each other but different to the first two, and so on. However, the main procedure remains the same as the first order neighbourhood structure. As a consequence, the details for the second order neighbourhood structure will be omitted. In both the first order and the second order neighbourhood structures, we only described the typical terms (Right panels of Figs. 1 and 2).



**Fig. 2** The second order neighbourhood structure for  $Z_i$  (left panel ) and the corresponding conditional dependence structure (right panel)

### 3 Theoretical property of SLPCD

Theorem 1 deals with the validity of using a synthetic likelihood in the form of Multinomial distribution, where the Multinomial distribution is estimated via Monte Carlo. Theorem 2 studies the error introduced by the lookup table for  $\beta$ .

**Theorem 1** For any fixed  $\beta \geq 0$ ,  $\tilde{\pi}_\beta(\mathbf{S} | \hat{\mathbf{P}})$  be the Multinomial synthetic likelihood with parameters  $\hat{\mathbf{P}} = (\hat{p}_1, \dots, \hat{p}_K)$  as estimated according to Algorithm 2, and  $\pi_\beta(\mathbf{S} | \mathbf{P}_0)$  denotes the Multinomial likelihood under the true parameters  $\mathbf{P}_0$ , then

$$d_{KL}(\tilde{\pi}_\beta(\mathbf{S} | \hat{\mathbf{P}}), \pi_\beta(\mathbf{S} | \mathbf{P}_0)) \xrightarrow{N \rightarrow \infty} 0,$$

where  $N$  is the number of Monte Carlo samples.

A proof is given in the Appendix A. Theorem 1 says that when the Multinomial distribution is used to model each piece of the likelihood function in 7, and if an unbiased estimate for the parameters in the Multinomial distribution is used to approximate the density at each  $\beta$ , then as the number of Monte Carlo samples,  $N$ , increases, the estimated density approaches to the density under the true parameter.

Then, we study the effect of using  $\beta$  over a grid in a lookup table. Theorem 2 describes the error bound between the posterior samples drawn from posterior distribution (9) using grid values of  $\beta$ , and posterior distribution attached to Eq. (7).

**Theorem 2** For all  $\beta \in [\beta_l, \beta_u]$ , assume that the first derivative of the posterior distribution is bounded and continuous:  $|\pi'_C(\beta | \mathbf{Z})| \leq K_1 < \infty$ . Define  $F(\beta) \triangleq \beta \pi_C(\beta | \mathbf{Z})$  and assume the second derivative of  $F(\beta)$  is bounded and continuous,  $|F''(\beta)| \leq K_2 < \infty$ . Then as  $N \rightarrow \infty$ , such that for some constant  $K$  which is a function of  $K_1$  and  $K_2$ ,

$$\lim_{M \rightarrow \infty} |E_{CG}(\beta) - \beta| = 0$$

where  $M$  is the number of equal-spaced grid points in  $[\beta_l, \beta_u]$  and  $\mathbb{E}_{CG}(\beta)$  is the expectation of  $\beta$  derived from the Algorithm 3. Further, define  $G(\beta) \triangleq \beta^2 \pi_C(\beta | \mathbf{Z})$  and assume the second derivative of  $G(\beta)$  is continuous and bounded  $|G''(\beta)| \leq K_3 < \infty$ . Then for some constant  $K'$  which is a function of  $K_1, K_2$  and  $K_3$ ,

$$\lim_{M \rightarrow \infty} |\mathbb{V}_{CG}(\beta) - \mathbb{V}_C(\beta)| = 0,$$

where  $\mathbb{V}_{CG}(\beta)$  is the variance of  $\beta$  obtained from the Algorithm 3.

Theorem 2 demonstrates that the error of expectation and the variance between posterior distribution (9) approximated using  $M$  grid values of  $\beta$  and the composite posterior distribution in (7) are bounded, and vanish as  $M \rightarrow \infty$ . A proof is given in the Appendix B.

### 4 Simulation study

In this section, simulation results for the first and second order neighbourhood lattices using different methods are presented. Various scenarios are included to evaluate the performance: different values of  $q$  and different lattice sizes. Approximation-based approaches, such as the pseudo-likelihood (PL) method of Besag (1975), the reduced dependence approximation (RDA) of Friel et al. (2009) and the conditional subset method (RCoDA) of Zhu and Fan (2018) will be compared to SLPCD. At the same time, simulation-based approaches, such as the thermodynamic integration method of Green and Richardson (2002) (TDI) (when computationally feasible, this method can be taken as the gold standard, ignoring the sampling error of the Potts samples) will be demonstrated as well. These methods represent existing methods that are flexible enough to cope with large lattices. To make all the approaches comparable, we adopt all the above likelihood functions to derive corresponding posterior distributions and draw samples from them.

The root mean squared error (RMSE) measure, defined as

$$RMSE = \sqrt{\frac{1}{T} \sum_{i=1}^T (\beta - \hat{\beta}_i)^2},$$

is selected to evaluate the quality of estimations, where  $\hat{\beta}_i$  is taken as the mean of the posterior samples produced by the MCMC algorithm using the  $i$ th simulated data set.

It is well known that the Potts models exhibit the phenomenon of phase transition. That is, there exists a critical point  $\beta_{crit}$  in the parameter space, where for  $\beta > \beta_{crit}$ , the Potts models will transit from a disordered to an ordered pattern or phase. The sites in the Potts model will eventually all be in a single state as  $\beta$  increases. The property of  $\beta_{crit}$  has been widely discussed in the literature (Potts 1952). For a general  $q$ -state model, the precise value of the critical value is difficult to determine. Under the first order neighbourhood lattice structure, for  $q \in \{2, 3, 4\}$ , 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 consider the set of values 0.1, 0.2, ..., 1 for  $\beta$ . Under the scenario of  $q = 3$ , we have to omit the results of the method RDA, which cannot be easily extended to the case for higher values of  $q$ .

For the simulation study, three different lattice sizes are considered, including  $32 \times 32$ ,  $128 \times 128$  and  $256 \times 256$ . We used  $T = 200$  simulated datasets across a grid of  $\beta$  values. Using the lookup tables produced from Algorithm 2, we used a Random walk Metropolis-Hastings algorithm to sample the posterior distribution  $\pi(\beta | \mathbf{Z})$ , where we take the prior for  $\beta \sim U(0, 1)$ . We used a Gaussian random walk proposal distribution, setting the variance/tuning parameter to be  $\sigma^2 = 0.015^2$ , and ran the MCMC for 6000 iterations, keeping the final 4000 iterations for inference.

Table 3 shows the resulting RMSE and the standard error of the estimates. Both the SLPCD and TDI perform consistently well compared to the other methods across different lattice size when  $\beta$  is under or slightly larger than the critical values. Overall, RMSE reduces as the size of the lattice increases for all the methods, with negligible differences between the methods for large lattices. In terms of spatial correlation strength  $\beta$ , SLPCD and TDI perform better as  $\beta$  increases, whereas PL and RCoDA have the opposite behavior. Overall, the most performance gain in SLPCD (and TDI) over PL and RCoDA is seen for higher values of  $\beta$  in smaller lattices. The performance seems to be consistent across  $q$ . Note that, when the true  $\beta$  is significantly larger than the critical value, the performance of both TDI and SLPCD deteriorate noticeably. The reason is that both these approaches rely on the MCMC simulation of the Potts model. The simulation of Potts model at large  $\beta$  values becomes unstable and extremely time-consuming, where convergence can be difficult to monitor, (see for example Møller et al. (2006)). We suggest not to use simulation-based methods when there is evidence of phase transition.

Phase transition also exists in the second order neighbourhood Potts models. The critical values for the second order Potts models cannot be obtained in closed form. Zhu and Fan (2018) 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  $\beta < 0.5$ . For more details about critical values of the second order Potts models, see Zhu and Fan (2018).

Under the similar conditions adopted in the simulation study of the first order neighbourhood, Table 4 shows RMSE for all the feasible methods under scenarios of  $q = 2$  and  $q = 3$ . Overall, SLPCD outperforms other methods with respect to RMSE. Again, the most marked differences in performance is observed at higher values of  $\beta$  and in smaller fields. All the methods improve their performance as size increases. However, it can be seen that SLPCD continues to outperform competing methods in the larger fields. SLPCD appears to work better under the second order neighbourhood structure compared to the first order neighbourhood structures. Overall, SLPCD outperforms all other methods considered under the second order neighbourhood structure. This might be because in the second order neighbourhood structure, the partial dependence now includes more observations around  $Z_i$  compared to the first order, hence retaining more dependencies.

**Table 3** RMSE ( $\times 10^{-2}$ ) and the standard errors (the values in the brackets) of  $\beta$  for the Potts models with the first order neighbourhood structure. Bold fonts highlight the smallest RMSE

	$q$	$\beta$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	
32 <sup>2</sup>	2	RCoDA	3.9 (3.8)	4.7 (4.6)	4.8 (4.7)	5.3 (5.1)	5.3 (5.1)	5.7 (5.2)	5.7 (4.9)	5.1 (4.5)	6.5 (2.5)	7.6 (7.5)	
		PL	4.3 (4.0)	4.6 (4.6)	4.4 (4.4)	4.9 (4.5)	4.8 (4.6)	4.6 (4.5)	4.6 (4.3)	5.3 (4.9)	3.5 (3.4)	5.7 (5.6)	
		TDI	4.0 (3.7)	<b>4.2</b> (4.1)	<b>4.2</b> (4.1)	4.3 (4.1)	<b>3.8</b> (3.8)	3.7 (3.1)	3.6 (3.2)	3.2 (3.2)	2.6 (2.5)	<b>4.8</b> (4.8)	
		RDA	4.0 (3.9)	4.3 (4.3)	4.2 (4.2)	4.3 (4.3)	3.8 (3.8)	3.7 (3.7)	3.6 (3.6)	3.2 (3.2)	<b>1.0</b> (0.1)	5.1 (0.1)	
		SLPCD	<b>3.7</b> (3.7)	4.6 (4.5)	4.4 (4.4)	<b>4.3</b> (4.2)	3.9 (3.9)	<b>3.7</b> (3.7)	<b>3.1</b> (3.0)	<b>2.8</b> (2.7)	2.8 (2.7)	5.5 (5.5)	
		3	RCoDA	3.9 (3.6)	4.7 (4.6)	5.1 (4.7)	5.1 (4.8)	5.3 (4.4)	4.9 (4.4)	4.9 (4.8)	4.6 (4.1)	4.9 (2.0)	5.5 (5.4)
	PL	4.4 (4.2)	4.6 (4.6)	4.7 (4.6)	4.9 (4.4)	5.1 (4.2)	4.4 (4.2)	4.2 (4.0)	4.7 (4.4)	3.6 (3.6)	4.2 (4.2)		
	TDI	4.0 (3.8)	<b>4.4</b> (4.0)	<b>4.5</b> (4.1)	4.5 (4.1)	4.5 (4.1)	3.9 (3.4)	3.4 (3.4)	3.4 (3.1)	<b>2.4</b> (2.4)	<b>2.5</b> (1.3)		
	SLPCD	<b>3.9</b> (3.9)	4.5 (4.5)	4.6 (4.6)	<b>4.3</b> (4.3)	<b>3.9</b> (3.9)	<b>3.8</b> (3.8)	<b>3.2</b> (3.2)	<b>3.4</b> (3.4)	2.7 (2.5)	5.1 (2.8)		
	128 <sup>2</sup>	2	RCoDA	1.1 (1.0)	1.2 (1.2)	1.1 (1.1)	1.4 (1.2)	1.2 (1.2)	1.4 (1.3)	1.6 (1.3)	1.7 (1.3)	3.0 (1.1)	2.6 (1.7)
			PL	1.1 (1.0)	1.1 (1.1)	1.1 (1.1)	1.2 (1.1)	1.1 (1.1)	1.1 (1.1)	1.2 (1.1)	1.2 (1.1)	1.1 (1.1)	<b>1.4</b> (1.3)
			TDI	<b>1.1</b> (1.1)	<b>1.1</b> (1.1)	<b>1.0</b> (1.0)	<b>1.0</b> (1.0)	0.9 (0.9)	<b>0.9</b> (0.8)	<b>0.8</b> (0.7)	0.7 (0.7)	0.9 (0.9)	4.1 (0.7)
SLPCD			1.1 (1.0)	1.2 (1.2)	1.1 (1.0)	1.1 (1.0)	<b>0.9</b> (0.9)	1.0 (0.9)	0.9 (0.8)	<b>0.6</b> (0.6)	<b>0.9</b> (0.8)	4.2 (1.1)	
3			RCoDA	1.1 (1.1)	1.2 (1.1)	1.1 (1.1)	1.1 (1.1)	1.1 (1.1)	1.2 (1.1)	1.2 (1.1)	1.3 (1.1)	1.7 (0.7)	2.0 (1.3)
PL			1.1 (1.1)	1.2 (1.1)	1.1 (1.1)	1.1 (1.1)	1.1 (1.1)	1.0 (1.0)	1.1 (1.1)	1.1 (1.1)	1.0 (1.0)	<b>1.0</b> (1.0)	
TDI		<b>1.1</b> (1.1)	<b>1.1</b> (1.1)	<b>1.0</b> (1.1)	<b>1.0</b> (1.1)	1.0 (0.9)	0.9 (0.8)	<b>0.8</b> (0.8)	0.8 (0.6)	<b>0.6</b> (0.6)	1.2 (0.3)		
SLPCD		1.2 (1.2)	1.1 (1.1)	1.2 (1.1)	1.1 (1.1)	<b>0.9</b> (0.9)	<b>0.9</b> (0.9)	1.0 (0.9)	<b>0.8</b> (0.8)	0.7 (0.6)	3.3 (1.0)		

**Table 3** (continued)

	$q$	$\beta$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	
256 <sup>2</sup>	2	RCoDA	0.6	0.5	0.6	0.6	0.6	0.8	0.9	1.3	2.6	2.4	
			(0.6)	(0.5)	(0.6)	(0.6)	(0.6)	(0.7)	(0.6)	(0.6)	(0.7)	(0.8)	
		PL	0.6	0.5	0.6	0.6	0.5	0.6	0.6	0.6	0.6	0.5	<b>0.6</b>
			(0.6)	(0.5)	(0.5)	(0.6)	(0.5)	(0.5)	(0.5)	(0.5)	(0.5)	(0.5)	(0.6)
		TDI	<b>0.5</b>	<b>0.5</b>	0.6	0.5	0.5	<b>0.4</b>	<b>0.4</b>	0.4	<b>0.4</b>	4.1	
			(0.5)	(0.5)	(0.5)	(0.5)	(0.5)	(0.4)	(0.4)	(0.4)	(0.3)	(0.6)	
		SLPCD	0.7	0.6	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	0.5	0.5	<b>0.4</b>	0.9	4.3	
			(0.6)	(0.6)	(0.5)	(0.5)	(0.5)	(0.4)	(0.4)	(0.3)	(0.3)	(0.4)	
	3	RCoDA	0.6	0.6	0.5	0.7	0.6	0.6	0.7	0.7	1.0	1.8	
				(0.6)	(0.6)	(0.5)	(0.6)	(0.6)	(0.6)	(0.7)	(0.6)	(0.4)	(0.7)
		PL	0.6	<b>0.5</b>	0.6	0.6	0.5	0.6	0.6	0.6	0.6	0.5	<b>0.5</b>
				(0.6)	(0.5)	(0.6)	(0.5)	(0.5)	(0.5)	(0.5)	(0.5)	(0.6)	(0.5)
TDI		<b>0.6</b>	0.6	<b>0.5</b>	<b>0.5</b>	0.5	0.5	<b>0.4</b>	0.4	0.3	1.1		
			(0.6)	(0.6)	(0.5)	(0.5)	(0.5)	(0.5)	(0.4)	(0.4)	(0.3)	(0.2)	
SLPCD		0.6	0.6	0.6	0.6	<b>0.5</b>	<b>0.5</b>	0.6	<b>0.4</b>	<b>0.3</b>	3.4		
			(0.6)	(0.6)	(0.5)	(0.6)	(0.5)	(0.5)	(0.5)	(0.4)	(0.2)	(0.7)	

The results are based on 200 simulated data sets for each 32<sup>2</sup>, 128<sup>2</sup> and 256<sup>2</sup> lattices. Both scenarios under  $q = 2$  and  $q = 3$  are shown

### 4.1 Computation time

All the four algorithms RCoDA, PL, TDI and SLPCD were implemented by the current authors in Matlab. The detailed information of the CPU of our machine is as follow: Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz. RDA was implemented in C language since the code was kindly provided by the authors (modified for our purposes). Table 5 shows a comparison of computation times for the different algorithms. All methods were implemented for the first order neighbourhood structure Ising models.

From Table 5, generally, PL, TDI and SLPCD are the fastest methods. While only SLPCD and TDI are scalable with respect to the lattice size. One of the reasons of the fast computation is that both TDI and SLPCD require look-up tables to be pre-calculated. At the moment, computational time for look-up tables are ignored in Table 5. The advantage of SLPCD over TDI lies in that the look-up table of SLPCD takes much less time than TDI.

SLPCD is more efficient than TDI in two aspects. First of all, as we mentioned before, one  $n \times n$  Potts model sample provides approximate  $n^2$  samples in the SLPCD approach, whereas it is only one sample in TDI. In this sense, SLPCD is



**Table 4** RMSE ( $\times 10^{-2}$ ) and standard errors (in bracket) of  $\beta$  for a second order neighbourhood dependence. Bold fonts highlight the smallest RMSE

		$\beta = 0.1$		$\beta = 0.2$		$\beta = 0.3$		$\beta = 0.4$	
		$q = 2$	$q = 3$	$q = 2$	$q = 3$	$q = 2$	$q = 3$	$q = 2$	$q = 3$
32 <sup>2</sup>	RCoDA	3.9 (3.0)	3.9 (3.2)	4.7 (3.0)	4.7 (3.1)	4.8 (2.9)	5.1 (2.9)	3.2 (2.2)	2.6 (2.5)
	PL	4.3 (2.6)	4.4 (3.0)	4.6 (2.4)	4.6 (2.6)	4.4 (2.0)	4.7 (2.3)	2.6 (1.8)	2.0 (1.8)
	TDI	4.0 (3.0)	4.0 (2.8)	4.2 (2.6)	4.4 (2.5)	4.2 (2.2)	4.5 (2.1)	2.4 (1.0)	1.2 (1.2)
	SLPCD	<b>3.1</b> (2.7)	<b>2.9</b> (2.9)	<b>3.1</b> (2.2)	<b>2.6</b> (2.5)	<b>2.0</b> (1.5)	<b>2.1</b> (2.0)	0.9 (0.8)	<b>1.2</b> (1.1)
128 <sup>2</sup>	RCoDA	1.1 (0.8)	1.1 (0.9)	1.2 (0.8)	1.2 (0.8)	1.1 (0.6)	1.1 (0.7)	0.6 (0.6)	0.7 (0.6)
	PL	1.1 (0.6)	1.1 (0.8)	1.1 (0.6)	1.2 (0.7)	1.1 (0.5)	1.1 (0.6)	0.6 (0.4)	0.5 (0.4)
	TDI	1.1 (0.9)	1.1 (0.8)	1.1 (0.6)	1.1 (0.6)	1.0 (0.5)	1.0 (0.5)	0.5 (0.4)	0.3 (0.3)
	SLPCD	<b>0.9</b> (0.7)	<b>0.8</b> (0.8)	<b>0.7</b> (0.5)	<b>0.6</b> (0.6)	<b>0.8</b> (0.4)	<b>0.5</b> (0.5)	0.5 (0.4)	<b>0.3</b> (0.3)
256 <sup>2</sup>	RCoDA	0.6 (0.4)	0.6 (0.5)	0.5 (0.4)	0.6 (0.4)	0.6 (0.3)	0.5 (0.4)	0.5 (0.3)	0.5 (0.3)
	PL	0.6 (0.3)	0.6 (0.4)	0.5 (0.3)	0.5 (0.3)	0.6 (0.2)	0.6 (0.3)	0.3 (0.2)	0.2 (0.2)
	TDI	0.5 (0.5)	0.6 (0.4)	0.5 (0.3)	0.6 (0.3)	<b>0.6</b> (0.2)	0.5 (0.2)	0.2 (0.2)	0.2 (0.1)
	SLPCD	<b>0.4</b> (0.4)	<b>0.4</b> (0.4)	<b>0.5</b> (0.3)	<b>0.3</b> (0.3)	0.8 (0.2)	<b>0.2</b> (0.2)	0.4 (0.2)	<b>0.1</b> (0.1)

The results are based on 200 simulated data sets for each 32<sup>2</sup>, 128<sup>2</sup> and 256<sup>2</sup> lattices. Both scenarios under  $q = 2$  and  $q = 3$  are shown

**Table 5** Computation times in seconds per iteration of MCMC

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	–	–
SLPCD	0.00006	0.00006	0.00007	0.00007

RDA was not implemented for the large lattice

$n^2$  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 under SLPCD

whereas TDI has to generate a different look-up table for each different size of the Potts models. Hence, SLPCD outperforms TDI overall and thus all other methods in terms of computational time.

Clearly, it is unfair to ignore pre-calculation times in TDI and SLPCD as shown in Table 5. As we just mentioned, TDI needs to calculate the look-up tables whenever the size of the Potts model changes. Therefore, the pre-calculation time must be taken into account when considering computational efficiency. On the other hand, SLPCD can apply its look-up table to any lattice size. Thus, the time consumed on computing look-up tables can be ignored. Thus, SLPCD is scalable with respect to lattice size. There are two quantities that affect the computational cost to produce the look-up tables of SLPCD. Firstly, the denseness of the grid for  $\beta$  and secondly, the Monte Carlo sample size. In this paper, the step size of  $\beta$  was set to be 0.001 when the look-up tables were calculated. For the second factor, 500 small ( $32 \times 32$ ) Potts models were generated which provided 512,000 relevant samples in total. This means 512,000 samples are generated to approximate the Multinomial distributions for each  $\beta$ . It takes 18.7 seconds to generate 512,000 samples. There are 1000  $\beta$ 's for a grid of size 0.001 in the interval of  $[0, 1]$ . Therefore, the pre-calculation takes around 5 hours. However, since parallel computing is now easily accessible, and computation for each  $\beta$  can be done simultaneously, this will further reduce the pre-calculation time.

## 5 Real data examples

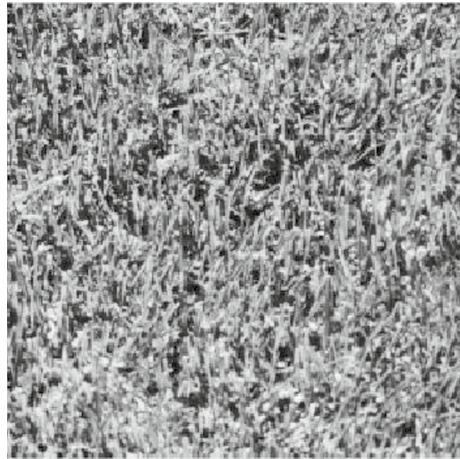
In this section, we present some real data examples of where Potts models are used. The first involve a texture analysis on a regular grid. In the second example, we demonstrate the use of SLPCD on an irregular lattice and a final example involves segmentation of a satellite image.

### 5.1 Texture analysis

Texture analysis is an important branch of computer vision and image segmentation (Bharati et al. 2004). Regions or patches on an image can be characterised by their texture content. Using measures such as gray scale on a photographic image, values at each pixel of the image can be used to segment the images into regions of similar textures. The Potts model is a natural choice for incorporating the spatial dependence when segmenting images (Haindl et al. 2012), which can then be used in applications such as texture synthesis where one may reproduce or enlarge the texture image. Another application is in object detection, where one may be interested in separating for example, weeds from grass in an automatic weed control systems (Watchareeruetai et al. 2006).

Here we consider the grass image available from the website <http://sipi.usc.edu/database/database.php?volume=textures>. The image was originally studied in Brodatz (1966) and was recently analyzed in Zhu and Fan (2018). Without loss of

**Fig. 3** A  $256 \times 256$  image of grass taken from Brodatz (1966)



generality, we take the first 256 rows and 256 columns as our data of interest, shown in Fig. 3.

Similarly to Zhu and Fan (2018), we use a two-component Gaussian mixture model of the form in Eq. (2), where  $\pi(y_i | \theta, Z_i)$  is taken as the Gaussian distribution for  $y_i$  conditional on the model parameters  $\theta = (\theta_1, \theta_2)$ , and  $\theta_j = (\mu_j, \sigma_j^2), j = 1, 2$  and  $Z_i, \pi(\theta)$  and  $\pi(\beta)$  denote the prior and hyper prior for the unknown parameters and  $y_i$  denotes the actual pixel value of the grass image. We set vague prior distributions for mean parameters and standard deviation parameters, that is  $\mu_j \sim N(0.5, 100^2)$  and  $\sigma_j^2 \sim IG(0.001, 0.001), j = 1, 2$ . Noninformative prior distribution is set on parameter  $\beta$ , that is  $\beta \sim \mathcal{U}[0, 4]$ . Given the posterior distribution as Eq. (2), the conditional distribution for each parameter is given as below,

$$\pi(Z_i | \cdot) \propto \pi(y_i | Z_i, \theta_{Z_i})\pi(Z_i | \beta, \partial Z_i), \text{ for } i = 1, \dots, n^2; \tag{10}$$

$$\pi(\beta | \cdot) \propto \pi_{CG}(\mathbf{Z} | \beta)\pi(\beta); \tag{11}$$

$$\pi(\theta_i | \cdot) \propto \prod_{j \in C_i} \pi(y_j | \theta_i), \text{ for } i = 1, 2, \tag{12}$$

where  $C_i$  is the set of index belonging to group  $i$ . A Gibbs sampler with Random Walk Metropolis-Hastings algorithm was adopted to draw posterior samples. In total, 6000 MCMC iterations was conducted, while keeping the latter 4000 iterations for further inference. It takes 0.59 seconds to run one iteration. The computation time is order of  $n^2$  in general. We fitted the Ising model with both the first order and the second order neighborhood structure respectively. The results are shown in Table 6.

Table 6 indicates that TDI and SLPCD obtain similar posterior distributions of all the parameters. This suggests that SLPCD performs very well in practice since

**Table 6** Posterior means and standard deviations (in brackets) for grass data for RCoDA, PL, TDI and SLPCD respectively

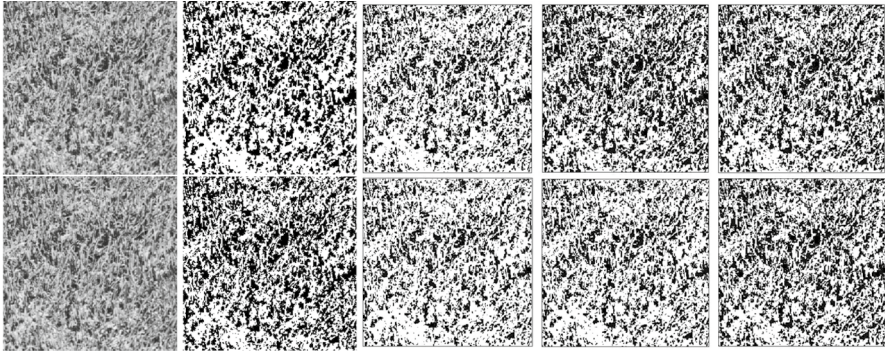
	$\mu_1$	$\mu_2$	$\sigma_1$	$\sigma_2$	$\beta$
RCoDA-(F)	0.265 (0.0028)	0.620 (0.0017)	0.014 (0.00037)	0.018 (0.00029)	1.280 (0.022)
PL-(F)	0.251 (0.0022)	0.609 (0.0015)	0.013 (0.00026)	0.019 (0.00027)	1.364 (0.017)
TDI-(F)	0.302 (0.0015)	0.650 (0.0010)	0.017 (0.00022)	0.013 (0.00014)	0.841 (0.0033)
SLPCD-(F)	0.301 (0.0016)	0.650 (0.0011)	0.017 (0.00023)	0.013 (0.00015)	0.853 (0.0032)
RCoDA-(S)	0.252 (0.0025)	0.611 (0.0016)	0.013 (0.00029)	0.019 (0.00030)	0.567 (0.0080)
PL-(S)	0.236 (0.0022)	0.599 (0.0015)	0.011 (0.00027)	0.021 (0.00029)	0.600 (0.0066)
TDI-(S)	0.303 (0.0015)	0.649 (0.0010)	0.017 (0.00021)	0.013 (0.00016)	0.373 (0.0013)
SLPCD-(S)	0.317 (0.0008)	0.662 (0.0012)	0.011 (0.00011)	0.019 (0.00022)	0.370 (0.0006)

(F) denotes first order neighbourhood structure. (S) denotes the second order neighbourhood structure

TDI is the only algorithm that computes the full dependence structure in the Potts model and can be considered as the benchmark. PL and RCoDA provides more similar results, while the mixture components parameters are not too different to those obtained by SLPCD and TDI, the estimates of  $\beta$  are higher from these other two approaches.

We proceed to assess the performance of the competing methods using the average in-sample posterior predictive mean squared error (MSE). Specifically, at each MCMC iteration, we generate the posterior predictive estimate of the image using the current estimate of parameters in the MCMC chain. We then compute the average of the squared pixel-by-pixel difference between the simulated image and the true image. The average posterior predictive error is then taken as the average over all the MCMC iterations. We obtained the value of 0.0150 for SLPCD, 0.0346 for PL, 0.0328 for RCoDA and 0.0296 for TDI in the first order model. This indicates that SLPCD outperforms the others. The corresponding average MSE for the second order models were 0.0151, 0.0368, 0.0343 and 0.0300 respectively, showing that SLPCD is again the best.

Although the above MSE is not an absolute measurement of goodness of fit, these values indicate the relative goodness of fit for all the approaches. SLPCD performs better than PL, RCoDA and TDI for this grass image. As an additional evaluation of our models, Fig. 4 provides visual representations of classifications based on the posterior estimates given by PL, RCoDA and TDI respectively. Visually, the differences can be difficult to discern and similar images were produced by all four methods. Results from PL appears to be most different from the other three methods.



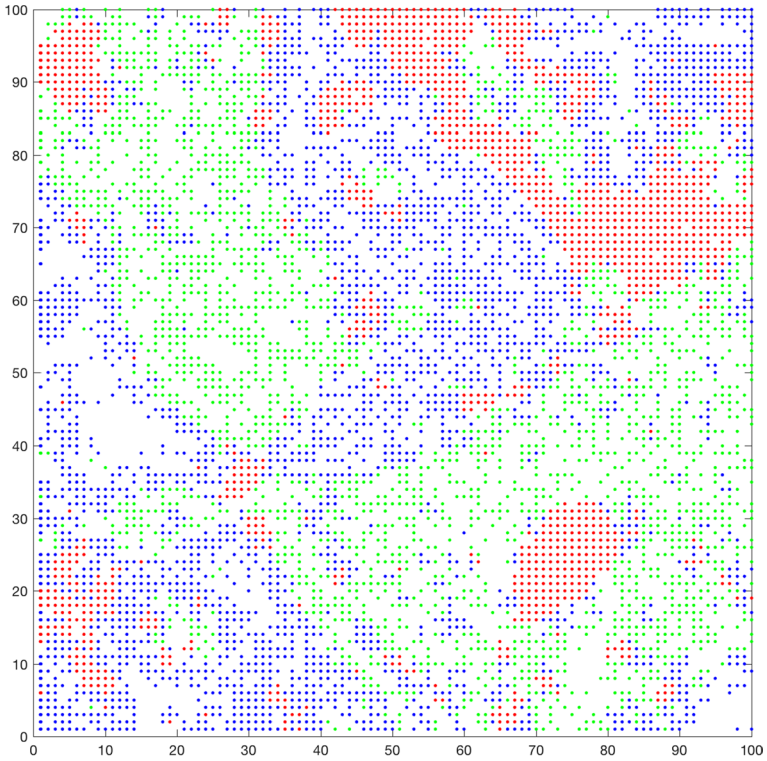
**Fig. 4** Results of the classifications. Respectively ground truth, SLPCD, PL, RCoDA and TDI from left to right. The first row correspond to first order neighbourhood structure, the second row correspond to second order neighbourhood structure

Figure 4 is also a showcase of how to generate a similar texture to the ground truth texture. In this process, a good estimator of  $\beta$ , which controls the interaction in the texture, would be beneficial.

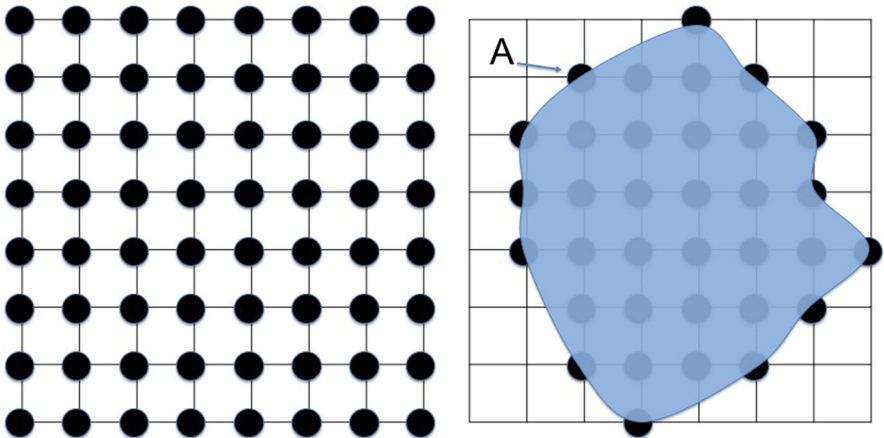
## 5.2 Lung cancer pathology example on an irregular lattice

Digital pathology imaging of tumor tissues capturing histological details in high resolution, is fast becoming a routine clinical procedure. Statistical methods can be developed to utilize these high-resolution images to assist patient prognosis and treatment planning. Researchers have designed a pipeline to predict cell types in pathology images, details of the project can be found at <https://qbrc.swmed.edu/projects/cnn/>. The predicted images of cell types can be further employed to study spatial pattern and interactions between different cell types. It is reported in Li et al. (2018) that the spatial pattern and interaction may reveal important information about tumor cell growth and its micro-environment. We consider the problem of modelling a pathology image with irregular locations involving three different types of cells: lymphocyte, stromal, and tumor cells. The pathology is pre-processed by ConvPath (<https://qbrc.swmed.edu/projects/cnn/>). The image of interest was collected by the Non-small-cell lung cancer (NSCLC) project (<https://biometry.nci.nih.gov/cdas/nlst/>). This image is a lung cancer pathology image. The original pre-processed image is of  $1000 \times 1000$ , and there are many empty pixels in the image. The original image is very sparse, it is not ideal to illustrate our approach. Thus, we downsampled the original image to  $100 \times 100$ , see Fig. 5. We employed a  $10 \times 10$  window to scan the original image and count the number of cells of each type. The  $10 \times 10$  window would be replaced by the most frequent cell type. This results in the downsampled image as shown by Fig. 5.

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



**Fig. 5** Lung cancer pathology image. Lymphocyte, stromal, and tumor cells are marked in red, green and blue, respectively. Patches of white can be seen in the image which correspond to no cell



**Fig. 6** An  $8 \times 8$  lattice. Left panel: regular lattice. Right panel: irregular lattice



lattice are demonstrated in Fig. 6 (left panel). An irregular lattice can always be included inside a regular lattice, as shown in the right panel of Fig. 6. 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 Fig. 6 has only two neighbours, whereas it would have four neighbours in a regular lattice.

In summary, the irregularity changes the boundary area, leading to extra irregular pixels which generates special conditional distributions in SLPCD. The special pixels can be handled naturally in the algorithm of SLPCD, see Sec. 2.4.

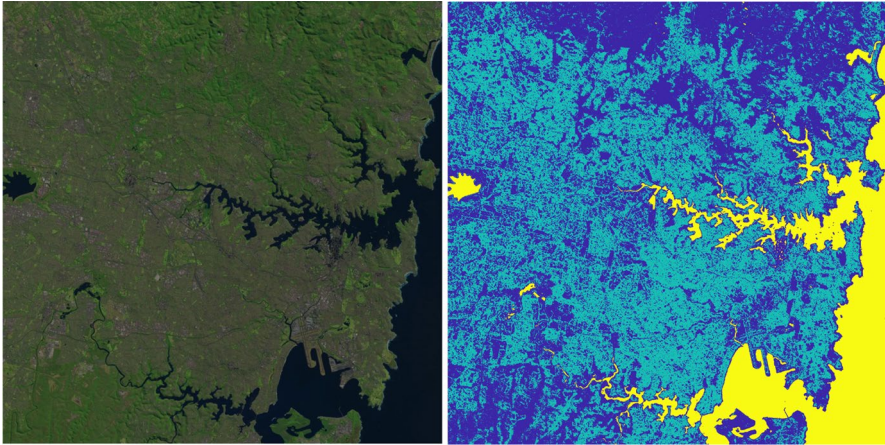
Figure 5 shows the distribution of different cell types in the lung cancer pathology image. We use a Potts model defined on irregular lattice with  $q = 3$  in order to model the different cell types. We aim to estimate spatial correlation of this Potts model for further clinical diagnosis. Li et al. (2018) proposed a Potts mixture model to fit this figure where they partition the lattice into several subregions, and in each region a spatial correlation is estimated. These estimated  $\beta$  are further used in survival analysis. Here we will estimate the spatial correlation for the entire region. The prior distribution of  $\beta$  is set to be uniform distribution over  $[0, 0.9]$ . The proposal distribution is chosen to be  $\beta^* \sim \mathcal{N}(\beta^t, 0.005^2)$ . In total, 6000 MCMC iterations were implemented and the first 2000 iteration were abandoned as burnin. It takes  $5 \times 10^{-3}$  for each iteration.

As other approaches, such as PL, RCoDA and TDI are not suitable for irregular lattices, only SLPCD can be used here to estimate  $\beta$ . Posterior mean of  $\beta$  is estimated to be 0.358 and standard deviation of  $\beta$  is estimated as  $5.8 \times 10^{-4}$ . Further work will be needed to extend the use of SLPCD in the lung cancer pathology analysis context and is beyond the scope of this paper.

### 5.3 Satellite remote sensing image

Massive satellite remote sensing image data are available nowadays. Such images can be utilized in various domain areas, for example in land use estimation (Small 2001), ocean color monitoring (McClain 2009) and vegetation mapping (Molnár et al. 2007). Due to the large size of the satellite images, automatic algorithms are always preferred to complete such tasks. In this example, we demonstrate via a real satellite remote sensing image, which will be classified into three clusters which denote different vegetation type. Such analyses can be used by environmental scientists to estimate global levels of plant biomass and monitor land use monitoring (Moores et al. 2020). We aim to classify all the pixels in the remote sensing image into different land types, such as water and forest.

The remote sensing images of Sydney was downloaded from from the USGS/EROS EarthExplorer website (<https://earthexplorer.usgs.gov/>). The downloaded image is shown in the left panel of Fig. 7. The size of this image is  $1600 \times 1600$ . We use hidden Potts model with  $q = 3$  to model this image and apply SLPCD to implement parameter inference. The prior distribution of  $\beta$  is set to be the uniform distribution over the interval of  $[0, 0.9]$ . The proposal distribution of  $\beta$  is chosen



**Fig. 7** Left: a satellite remote sensing image of Sydney. Right: the MAP estimate of the vegetation types for Sydney. The yellow color denotes water (ocean/river), blue color denotes forest and the light blue color denotes developed areas (industrial/residential)

to be  $\beta^* \sim \mathcal{N}(0, 0.002^2)$ . Totally, 2000 MCMC iterations have been conducted and 1000 iterations are thrown away as burnin. It takes 2.14 seconds to run one iteration, since the observation is a multivariate normal distribution. The posterior mean and standard deviation of  $\beta$  is 0.416 and  $5.7 \times 10^{-4}$  respectively. The right panel of Fig. 7 shows the MAP estimated image segmentation of vegetation types. Given the estimation, scientists can monitor land use over a specific area.

## 6 Discussion and conclusion

In this paper, we proposed a synthetic likelihood to approximate the likelihood function of the Potts model. Our method avoids calculation of the intractable normalizing constant. Instead, the likelihood function is transformed to be the product of many conditional density functions and the distribution of the corresponding summary statistics is approximated through Monte Carlo simulations.

SLPCD was proposed on account of three considerations. Firstly, it is fast. It takes much less time than comparable methods with and without counting in the calculation of the look-up tables, as discussed in Sect. 4.1. Our simulation studies show that the computational savings did not come at a cost of accuracy. The two other defining features of SLPCD are its ability to scale up to larger lattices, and ability to handle irregular lattices. These capacities are important in the context of modern images with higher and higher resolutions, and ever more complex.

Our simulation results show that SLPCD performs similarly to TDI, which is the only method that is exact, in the sense that there are no added approximations to the likelihood. In this sense, SLPCD has outperformed all other approximation methods. Our simulations suggest that the SLPCD approach is both reliable and very efficient.



It is noticeable that the look-up table can be utilized in different fashions. Throughout our simulation studies, a grid of  $\beta$ 's were considered due to the simplicity of univariate setting. As the step size of the chosen grid is 0.001, the granularity of estimation cannot be finer than 0.001. There are two alternatives to achieve finer granularity of estimation. The first approach is to utilize the current look-up table. For the values of  $\beta$  which are not in the look-up table, the corresponding PCD could be obtained by interpolation. This approach is as computationally efficient as our current implementation. However, as dimension of parameters increases, interpolation could be problematic. The second approach is not to use a look-up table at all. For any value of  $\beta$ , an online estimation of its corresponding PCD can be obtained via Monte Carlo simulation. This approach in theory can eliminate the errors caused by interpolation. The disadvantage is that it is computationally intensive for Markov chain Monte Carlo algorithms but maybe feasible using variational Bayesian inference where the number of likelihood evaluations are much smaller. This approach is particularly useful as the dimension of parameters increases.

### Appendix A Proof of Theorem 1

**Proof** Let  $S_i$  denote the sufficient statistic of  $\pi(Z_i | Z_{i-1}, Z_{i-n}, \beta)$ ,  $\hat{P}$  denote the parameters estimated from the Monte Carlo simulation and  $P_0$  denote the true parameters in the Multinomial distribution. As  $\hat{p}_{S_i}$  is a consistent estimator of  $p_{0,S_i}$ , we have that for  $\forall \epsilon, \delta > 0, \exists N^* > 0$  s.t. for  $\forall N > N^*$ ,

$$P(|\hat{p}_{S_i} - p_{0,S_i}| < \delta \cdot p_{0,S_i}) \geq 1 - \epsilon.$$

As  $p_{0,S_i} > 0$ , it is straightforward to see

$$P\left(|\frac{\hat{p}_{S_i}}{p_{0,S_i}} - 1| < \delta\right) \geq 1 - \epsilon. \tag{A1}$$

Dropping notational dependence on  $\beta$ , working with Kullback-Leibler (KL) divergence and the Multinomial distribution with  $K$  categories, we have

$$\begin{aligned} KL(\pi(S_i | P_0) || \pi(S_i | \hat{P})) &= \mathbb{E}_{\pi(S_i|P_0)} \left[ \log \frac{\pi(S_i|P_0)}{\pi(S_i|\hat{P})} \right] \\ &= \mathbb{E}_{\pi(S_i|P_0)} \left[ \log p_{0,S_i} - \log \hat{p}_{S_i} \right] \\ &= \sum_{S_i} \pi(S_i | P_0) \left[ \log p_{0,S_i} - \log \hat{p}_{S_i} \right] \\ &= - \sum_{S_i} \pi(S_i | P_0) \left[ \log \left( \frac{\hat{p}_{S_i}}{p_{0,S_i}} \right) \right]. \end{aligned}$$

Combining the above Equations with Eq. (A1), we can conclude that  $KL(\pi(S_i | P_0) \rightarrow 0$  as  $N \rightarrow \infty$ , where  $N$  is the sample size of Monte Carlo simulation. Since  $S$  is sufficient for  $Z$ , then we can replace each piece inside  $\pi_C(Z | \beta)$  with the Multinomial likelihood over  $S_i$  without loss of information (Cam 1964).  $\square$

### Appendix B Proof of Theorem 2

**Proof** According to Algorithm 3,  $\pi_{CG}(\beta | \mathbf{Z})$  is a step function where for some specified value  $M$ ,  $\pi_{CG}(\beta = \beta^* | \mathbf{Z}) = \pi_C(\beta_k | \mathbf{Z})$ , for  $\beta^* \in \left[ \beta_k - \frac{(\beta_u - \beta_l)/M}{2}, \beta_k + \frac{(\beta_u - \beta_l)/M}{2} \right]$ . In other words,  $\pi_{CG}(\beta | \mathbf{Z})$  is a discrete approximation of  $\pi_C(\beta | \mathbf{Z})$ .

Then we have,

$$\begin{aligned} \mathbb{E}_C(\beta) &= \int_{\beta_l}^{\beta_u} \beta \pi_C(\beta | \mathbf{Z}) d\beta, \\ \mathbb{E}_{CG}(\beta) &= \frac{1}{Q_\epsilon} \sum_{k=1}^M \beta_k \pi_C(\beta = \beta_k | \mathbf{Z}) = \frac{1}{Q_\epsilon} \mathbb{E}_{CG}^*(\beta), \end{aligned}$$

where  $Q_\epsilon = \sum_{k=1}^M \pi_C(\beta = \beta_k | \mathbf{Z})$  is the normalizing constant of  $\pi_{CG}(\beta | \mathbf{Z})$ . Obviously,  $Q_l \leq Q_\epsilon < \infty$  and  $Q_\epsilon$  is the numerical integration of  $\int_{\beta_l}^{\beta_u} \pi_C(\beta | \mathbf{Z}) d\beta$ . According to the error bound theory of numerical integration,

$$\left| Q_\epsilon - \int_{\beta_l}^{\beta_u} \pi_C(\beta | \mathbf{Z}) d\beta \right| = |Q_\epsilon - 1| \leq K_1 \frac{(\beta_u - \beta_l)^3}{24M^2}.$$

Similarly,  $\mathbb{E}_{CG}^*(\beta)$  is the numerical integration of  $\mathbb{E}_C(\beta)$  with mid-point rule. Thus,

$$|\mathbb{E}_{CG}^*(\beta) - \mathbb{E}_C(\beta)| \leq K_2 \frac{(\beta_u - \beta_l)^3}{24M^2}.$$

Combining the above, we have

$$\begin{aligned} &|\mathbb{E}_{CG}(\beta) - \mathbb{E}_C(\beta)| \\ &= \left| \frac{1}{Q_\epsilon} \mathbb{E}_{CG}^*(\beta) - \mathbb{E}_C(\beta) \right| \\ &= \frac{1}{Q_\epsilon} |(\mathbb{E}_{CG}^*(\beta) - \mathbb{E}_C(\beta)) + (1 - Q_\epsilon)\mathbb{E}_C(\beta)| \\ &\leq \frac{1}{Q_\epsilon} |(\mathbb{E}_{CG}^*(\beta) - \mathbb{E}_C(\beta))| + \frac{1}{Q_\epsilon} |(1 - Q_\epsilon)\mathbb{E}_C(\beta)| \\ &\leq \frac{K_2}{Q_l} \frac{(\beta_u - \beta_l)^3}{24M^2} + \frac{|1 - Q_\epsilon|}{Q_\epsilon} \mathbb{E}_C(\beta) \\ &\leq \frac{K_2}{Q_l} \frac{(\beta_u - \beta_l)^3}{24M^2} + \frac{E}{Q_l} K_1 \frac{(\beta_u - \beta_l)^3}{24M^2} \\ &= K \frac{(\beta_u - \beta_l)^3}{24M^2}. \end{aligned}$$

As reported in Varin et al. (2011), the estimator from the composite likelihood is unbiased, that is,  $\mathbb{E}_C(\beta) = \beta$ . Therefore,

$$\lim_{M \rightarrow \infty} |\mathbb{E}_{CG}(\beta) - \beta| = 0.$$

Similarly for the variance, we have

$$\begin{aligned} \mathbb{V}_C(\beta) &= \mathbb{E}_C(\beta^2) - (\mathbb{E}_C(\beta))^2 = \int_{\beta_l}^{\beta_u} G(\beta)d\beta - (\mathbb{E}_C(\beta))^2, \\ \mathbb{V}_{CG}(\beta) &= \mathbb{E}_{CG}(\beta^2) - (\mathbb{E}_{CG}(\beta))^2 = \frac{1}{Q_\epsilon} \sum_{k=1}^M \beta_k^2 \pi_C(\beta = \beta_k | \mathbf{Z}) - (\mathbb{E}_{CG}(\beta))^2. \end{aligned}$$

and

$$\begin{aligned} & | \mathbb{V}_{CG}(\beta) - \mathbb{V}_C(\beta) | \\ &= | \frac{1}{Q_\epsilon} \sum_{k=1}^M \beta_k^2 \pi_C(\beta = \beta_k | \mathbf{Z}) - (\mathbb{E}_{CG}(\beta))^2 - \int_{\beta_l}^{\beta_u} G(\beta)d\beta + (\mathbb{E}_C(\beta))^2 | \\ &\leq | \frac{1}{Q_\epsilon} \sum_{k=1}^M \beta_k^2 \pi_C(\beta = \beta_k | \mathbf{Z}) - \int_{\beta_l}^{\beta_u} G(\beta)d\beta | + | (\mathbb{E}_C(\beta))^2 - (\mathbb{E}_{CG}(\beta))^2 | \\ &\leq \frac{1}{Q_\epsilon} | \sum_{k=1}^M G(\beta_k) - \int_{\beta_l}^{\beta_u} G(\beta)d\beta | + \frac{|1 - Q_\epsilon|}{Q_\epsilon} \int_{\beta_l}^{\beta_u} G(\beta)d\beta + | (\mathbb{E}_C(\beta))^2 - (\mathbb{E}_{CG}(\beta))^2 | \\ &\leq \frac{K_3}{Q_l} \frac{(\beta_u - \beta_l)^3}{24M^2} + \frac{E'K_1}{Q_l} \frac{(\beta_u - \beta_l)^3}{24M^2} + 2EK \frac{(\beta_u - \beta_l)^3}{24M^2} \\ &= K' \frac{(\beta_u - \beta_l)^3}{24M^2}. \end{aligned}$$

Therefore,

$$\lim_{M \rightarrow \infty} | \mathbb{V}_{CG}(\beta) - \mathbb{V}_C(\beta) | = 0.$$

□

**Funding** Open Access funding enabled and organized by CAUL and its Member Institutions.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

**References**

Alsing J, Charnock T, Feeny S, Wand elt B (2019). Fast likelihood-free cosmology with neural density estimators and active learning. arXiv e-prints, page [arXiv:1903.00007](https://arxiv.org/abs/1903.00007)

- Atchadé YF, Liu JS (2010) The wang-landau algorithm in general state spaces: applications and convergence analysis. *Statistica Sinica*, 209–233
- Bartolucci F, Besag J (2002) A recursive algorithm for Markov random fields. *Biometrika* 89(3):724–730
- Besag J (1974) Spatial interaction and the statistical analysis of lattice systems. *J Royal Stat Soc. Series B (Methodological)* 36(2):192–236
- Besag J (1975) Statistical analysis of non-lattice data. *J Royal Stat Soc. Series D (The Statistician)* 24(3):179–195
- Bharati MH, Liu JJ, MacGregor JF (2004) Image texture analysis: methods and comparisons. *Chemomet Intelligent Lab Sys* 72(1):57–71
- Brodatz P (1966) *Textures: a photographic album for artists and designers*. Dover Publications, New York, Dover pictorial archives
- Cam LL (1964) Sufficiency and approximate sufficiency. *Ann Math Statist* 35(4):1419–1455
- Cressie N, Davidson JL (1998) Image analysis with partially ordered Markov models. *Comput Stat Data Analys* 29(1):1–26
- Everitt RG (2012) Bayesian parameter estimation for latent Markov random fields and social networks. *J Comput Graphic Stat* 21(4):940–960
- Fan Y, Nott DJ, Sission SA (2014) Approximate Bayesian computation via regression density estimation. *Stat* 2(1):34–48
- Feng D (2008) *Bayesian hidden Markov normal mixture models with application to MRI tissue classification*. PhD thesis, University of Iowa
- Feng D, Tierney L, Magnotta V (2012) MRI tissue classification using high-resolution Bayesian hidden Markov normal mixture models. *J Am Stat Assoc* 107(497):102–119
- Friel N, Pettitt A, Reeves R, Wit E (2009) Bayesian inference in hidden Markov random fields for binary data defined on large lattices. *J Comput Graphic Stat* 18(2):243–261
- Gelman A, Meng X-L (1998) Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. *Statistical Sci* 13(2):163–185
- Geyer CJ, Thompson EA (1992) Constrained Monte Carlo maximum likelihood for dependent data (with discussion). *J Royal Stat Soc. Series B. Methodological* 54(3):657–699
- Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>
- Grazian C, Fan Y (2019) A review of approximate bayesian computation methods via density estimation: Inference for simulator-models. *WIREs Computational Statistics*, n/a(n/a):e1486
- Green PJ, Richardson S (2002) Hidden Markov models and disease mapping. *J Am Stat Assoc* 97(460):1055–1070
- Grelaud A, Robert CP, Marin J-M, Rodolphe F, Taly J-F et al (2009) ABC likelihood-free methods for model choice in Gibbs random fields. *Bayesian Analysis* 4(2):317–335
- Gu MG, Zhu H-T (2001) Maximum likelihood estimation for spatial models by Markov chain Monte Carlo stochastic approximation. *J Royal Stat Soc: Series B (Statistical Methodology)* 63(2):339–355
- Haindl M, Remeš V, Havlíček V (2012) Potts compound Markovian texture model. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 29–32. IEEE
- Hinton G (2014) Boltzmann machines. In *Encyclopedia of Machine Learning and Data Mining*, Springer Science, New York
- Hurn MA, Husby OK, Rue H (2003) A tutorial on image analysis. In: Møller J (ed) *Spatial statistics and computational methods*, vol 173. *Lecture Notes in Statistics*. New York, Springer, pp 87–141
- Li Q, Wang X, Liang F, Yi F, Xie Y, Gazdar A, Xiao G (2018) A Bayesian hidden potts mixture model for analyzing lung cancer pathology images. *Biostatistics* 00(00):1–17
- Li SZ, Singh S (2009) *Markov random field modeling in image analysis*. *Advances in Computer Vision and Pattern Recognition*. Springer-Verlag London, London, 3 edition
- Liang F (2007) Continuous contour Monte Carlo for marginal density estimation with an application to a spatial statistical model. *J Comput Graphic Stat* 16(3):608–632
- Liang F (2010) A double Metropolis-Hastings sampler for spatial models with intractable normalizing constants. *J Stat Comput Simulat* 80(9):1007–1022
- Liang F, Jin IH, Song Q, Liu JS (2016) An adaptive exchange algorithm for sampling from distributions with intractable normalising constants. *J Am Stat Assoc* 111(513):377–393
- McClain CR (2009) A decade of satellite ocean color observations. *Annual Rev Marine Sci* 1:19–42
- Møller J, Pettitt AN, Reeves R, Berthelsen KK (2006) An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika* 93(2):451–458

- Molnár Z, Bartha S, Seregélyes T, Illyés E, Botta-Dukát Z, Tímár G, Horváth F, Révész A, Kun A, Bölöni J et al (2007) A grid-based, satellite-image supported, multi-attributed vegetation mapping method (méta). *Folia Geobotanica* 42(3):225–247
- Moores M, Nicholls G, Pettitt A, Mengersen K et al (2020) Scalable bayesian inference for the inverse temperature of a hidden potts model. *Bayesian Analysis* 15(1):1–27
- Moores MT, Drovandi CC, Mengersen K, Robert CP (2015) Pre-processing for approximate bayesian computation in image analysis. *Statistics and Computing* 25(1):23–33
- Murray I (2007) Advances in Markov chain Monte Carlo methods. PhD thesis, University of Cambridge
- Murray I, Ghahramani Z, MacKay DJC (2006) MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 359–366. AUAI Press
- Pal NR, Pal SK (1993) A review on image segmentation techniques. *Pattern Recognition* 26(9):1277–1294
- Papamakarios G, Sterratt DC, Murray I (2019) Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan.*, 89(00):00
- Pereyra M, Dobigeon N, Batatia H, Tourneret J (2013) Estimating the granularity coefficient of a potts-markov random field within a markov chain monte carlo algorithm. *IEEE Transac Image Proc* 22(6):2385–2397
- Potts RB (1952) Some generalized order-disorder transformations. In *Mathematical proceedings of the cambridge philosophical society*, volume 48, pages 106–109. Cambridge University Press
- Propp JG, Wilson DB (1996) Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random struct Algorith* 9(1–2):223–252
- Reeves R, Pettitt AN (2004) Efficient recursions for general factorisable models. *Biometrika* 91(3):751–757
- Sisson SA, Fan Y, Beaumont M (2018) *Handbook of Approximate Bayesian Computation*. Chapman & Hall, London
- Small C (2001) Estimation of urban vegetation abundance by spectral mixture analysis. *Int J Remote Sens* 22(7):1305–1334
- Storath M, Weinmann A, Friel J, Unser M (2015) Joint image reconstruction and segmentation using the potts model. *Inverse Problems* 31(2):025003
- Swendsen RH, Wang J-S (1987) Nonuniversal critical dynamics in Monte Carlo simulations. *Phys Rev Lett* 58(2):86
- Varin C, Reid NM, Firth D (2011) An overview of composite likelihood methods. *Statistica Sinica* 21(1):5–42
- Wang F, Landau DP (2001) Efficient, multiple-range random walk algorithm to calculate the density of states. *Phys Rev Lett* 86(10):2050
- Watchareeruetai U, Takeuchi Y, Matsumoto T, Kudo H, Ohnishi N (2006) Computer vision based methods for detecting weeds in lawns. *Machine Vision Appl* 17:287–296
- Wolff U (1989) Collective Monte Carlo updating for spin systems. *Phys Rev Lett* 62(4):361
- Wood SN (2010) Statistical inference for noisy nonlinear ecological dynamic systems. *Nature* 466(7310):1102–1104
- Wu F-Y (1982) The Potts model. *Rev Modern Phys* 54(1):235–268
- Zhu W, Fan Y (2018) A novel approach for Markov Random Field with intractable normalizing constant on large lattices. *J Comput Graphical Stat* 27(1):59–70