



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA

# **Structure Learning and Parameter Estimation of Probabilistic Context Neighborhoods**

**Débora de Freitas Magalhães**

Belo Horizonte, Brasil  
2021

**Débora de Freitas Magalhães**

# **Structure Learning and Parameter Estimation of Probabilistic Context Neighborhoods**

Thesis presented to the Graduate Program of Statistics at  
Universidade Federal de Minas Gerais (UFMG) in  
partial fulfillment of the requirements for the degree of  
Master in Statistics.

Advisor: Prof. Dr. Denise Duarte

Co-Advisor: Dr. Aline Piroutek

Belo Horizonte, Brasil

2021

Magalhães, Débora de Freitas.

M189s        Structure learning and parameter estimation of probabilistic  
context neighborhoods [manuscrito] / Débora de Freitas  
Magalhães. – 2021.  
60 f. il.

Orientadora: Denise Duarte Scarpa Magalhães.

Coorientadora: Aline Martines Piroutek.

Dissertação (mestrado) - Universidade Federal de Minas  
Gerais, Instituto de Ciências Exatas, Departamento de Estatística

Referências: f.49-53.

1. Estatística– Teses. 2. Markov, Campos aleatórios de –  
Teses. 3. Algoritmo de contexto – Teses. 4. Árvores  
probabilísticas de contexto – Teses. I. Magalhães, Denise Duarte  
Scarpa. II. Piroutek, Aline Martines. III. Universidade Federal de  
Minas Gerais, Instituto de Ciências Exatas, Departamento de  
Estatística. IV. Título.

CDU 519.2(043)



ATA DA DEFESA DE DISSERTAÇÃO DE MESTRADO DA ALUNA DÉBORA DE FREITAS MAGALHÃES, MATRICULADO, SOB O N° 2019.663.680, NO PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA, DO INSTITUTO DE CIÊNCIAS EXATAS, DA UNIVERSIDADE FEDERAL DE MINAS GERAIS, REALIZADA NO DIA 09 DE JULHO DE 2021.

Aos 09 dias do mês de Julho de 2021, às 13h30, em reunião pública virtual 261 (conforme orientações para a atividade de defesa de dissertação durante a vigência da Portaria PRPG nº 1819) no Instituto de Ciências Exatas da UFMG, <https://us02web.zoom.us/j/85351167306> reuniram-se os professores abaixo relacionados, formando a Comissão Examinadora homologada pelo Colegiado do Programa de Pós-Graduação em Estatística, para julgar a defesa de dissertação da aluna DÉBORA DE FREITAS MAGALHÃES, nº matrícula 2019.663.680, intitulada: "*Structure Learning and Parameter Estimation of Probabilistic Context Neighborhoods*", requisito final para obtenção do Grau de mestre em Estatística. Abrindo a sessão, a Senhora Presidente da Comissão, Profa. Denise Duarte Scarpa Magalhaes Alves, passou a palavra à aluna para apresentação de seu trabalho. Seguiu-se a arguição pelos examinadores com a respectiva defesa da aluna. Após a defesa, os membros da banca examinadora reuniram-se reservadamente sem a presença da aluna e do público, para julgamento e expedição do resultado final. Foi atribuída a seguinte indicação:

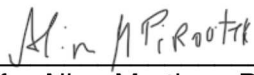
Aprovada.

Reprovada com resubmissão do texto em \_\_\_\_ dias.


Reprovada com resubmissão do texto e nova defesa em \_\_\_\_ dias.


Reprovada.

  
\_\_\_\_\_  
Profa. Denise Duarte Scarpa Magalhaes  
Alves – Orientadora (EST/UFMG)

  
\_\_\_\_\_  
Profa. Aline Martines Piroutek – Co-  
orientadora – (Doutora pela UFMG)

  
\_\_\_\_\_  
Prof. Marcos Oliveira Prates (EST/UFMG)

  
\_\_\_\_\_  
Prof. Rodrigo Lambert (FAMAT/UFU),

  
\_\_\_\_\_  
Caio Teodoro de Magalhaes Alves  
(Alfred Renyi Institute of Mathematics - Budapeste).

O resultado final foi comunicado publicamente à aluna pela Senhora Presidente da Comissão. Nada mais havendo a tratar, a Presidente encerrou a reunião e lavrou a presente Ata, que será assinada por todos os membros participantes da banca examinadora. Belo Horizonte, 09 de julho de 2021.

Observações:

1. No caso de aprovação da tese, a banca pode solicitar modificações a serem feitas na versão final do texto. Neste caso, o texto final deve ser aprovado pelo orientador da tese. O pedido de expedição do diploma do candidato fica condicionado à submissão e aprovação, pelo orientador, da versão final do texto.
2. No caso de reprovação da tese com resubmissão do texto, o candidato deve submeter o novo texto dentro do prazo estipulado pela banca, que deve ser de no máximo 6 (seis) meses. O novo texto deve ser avaliado por todos os membros da banca que então decidirão pela aprovação ou reprovação da tese.
3. No caso de reprovação da tese com resubmissão do texto e nova defesa, o candidato deve submeter o novo texto com a antecedência à nova defesa que o orientador julgar adequada. A nova defesa, mediante todos os membros da banca, deve ser realizada dentro do prazo estipulado pela banca, que deve ser de no máximo 6 (seis) meses. O novo texto deve ser avaliado por todos os membros da banca. Baseada no novo texto e na nova defesa, a banca decidirá pela aprovação ou reprovação da tese.

*À ciência brasileira*

# Agradecimentos

Em primeiro lugar, agradeço à minha família pelo apoio incondicional e por criarem condições para que eu pudesse me dedicar aos meus estudos. Especialmente aos meus pais, José Augusto e Graci, por investirem desde cedo na minha educação, e por me ensinarem o valor do trabalho e da dedicação. Às minhas irmãs, Bárbara e Vitória, obrigada pelos conselhos e por ouvirem meus desabafos. Ao Marco, agradeço o apoio à minha decisão de retornar ao Brasil para ficar perto da minha família e voltar à universidade. Obrigada também Tia Valéria, por todo carinho, paciência e orações.

Um agradecimento especial à minha orientadora Denise por sua calma, compreensão e confiança ao longo dessa jornada. Sou muito grata pelas nossas conversas e por seu incrível dom de me tranquilizar e apontar a direção a ser seguida. Agradeço também à minha co-orientadora Aline, por se disponibilizar a me ajudar a entender o seu trabalho e expandi-lo.

Em uma época de tanta tristeza e perdas em decorrência de uma pandemia global, não poderia deixar de agradecer à minha saúde física, mental e emocional, sem a qual seria impossível a conclusão desse trabalho. O meu “muito obrigada” ao SUS, aos pesquisadores, profissionais de saúde e pessoas da linha de frente no combate à COVID-19 por serem fontes de luz em tempos de escuridão.

Agradeço também aos meus colegas da Pós-Graduação em Estatística da UFMG, por criarem um ambiente positivo de companheirismo, onde colegas de trabalho se motivam, ajudam e torcem para o sucesso uns dos outros. Esse elemento foi fundamental para o aprendizado e conhecimento que adquiri nesse período.

Por fim, agradeço à CAPES pelo apoio financeiro concedido para realização dessa pesquisa.

# Resumo

As árvores probabilísticas de contexto oferecem uma representação mais eficiente para a dependência de uma Cadeia de Markov, tanto do ponto de vista computacional como em sua fácil interpretação. Essas vantagens permitiram que esses modelos fossem amplamente utilizados e suas propriedades, estudadas. A presente dissertação busca estudar a extensão desse modelo para reticulados em  $\mathbb{Z}^2$  introduzida por Piroutek (2013) e denominada modelo de contexto de vizinhança probabilística, ou em inglês, *probabilistic context neighborhood* (PCN). O modelo PCN propõe uma representação em forma de árvore para a dependência espacial de um campo aleatório de Markov bidimensional, permitindo que cada *site* dependa de uma vizinhança de tamanho variável, denominada *contexto*. Essa variação de campos aleatórios de Markov permite uma redução significativa dos parâmetros livres a serem estimados. No PCN, a estrutura de vizinhança é fixada em *frames*, diferentemente do trabalho feito em Csiszár e Talata (2006a), o que permite o cálculo da cardinalidade dos diferentes contextos de uma árvore e a proposta de um algoritmo que seleciona o melhor modelo baseado no critério PIC (*pseudo-Bayesian information criterion*). Nosso trabalho procura também validar o algoritmo PCN através de um estudo de simulações, além de exemplificar a aplicação do modelo para dados reais. Os resultados confirmam a adequação do algoritmo e sugerem que a cota do tamanho máximo da árvore permitida pode ser melhorada. Além disso, os resultados empíricos fornecem estimativas para as probabilidades de transição do processo.

**Palavras-Chave:** Campos aleatórios de Markov; Campos aleatórios de vizinhança variável; Algoritmo Contexto; Árvores probabilísticas de contexto; Seleção de modelos.

# Abstract

Probabilistic context trees offer a more efficient representation of the dependency of a Markov Chain, both in terms of the computational effort needed as well as its easy interpretability. This model has been extensively utilized and its properties have been studied by various authors. The present thesis aims to study an extension of the probabilistic context tree model to lattices in  $\mathbb{Z}^2$ , called probabilistic context neighborhood (PCN) model, introduced by Piroutek (2013). The PCN model proposes a tree representation for the spatial dependency of a two-dimensional Markov random field. It allows the sites of a region to depend on a variable neighborhood size, called *context*. This Markov random field variation is known in the literature as variable-neighborhood random field and it drastically reduces the number of free parameters to be estimated. In the PCN model, the neighborhood geometry is set to *frames* which allows us to calculate the cardinality of contexts of a given tree. Therefore, unlike the work of Csiszár and Talata (2006a), an algorithm is proposed to select the optimal model based on the pseudo-Bayesian information criterion (PIC). Our work seeks to validate the PCN algorithm through a simulation study. In addition, we exemplify the use of such methodology through a real-world data application. The results presented here confirm the adequacy of the algorithm, and suggest that the quota for the maximum depth of the tree could be further improved. Furthermore, an empirical study of the estimated transition probabilities indicate adequate estimates.

**Keywords:** Markov random fields; Variable-neighborhood random fields; Context algorithm, Probabilistic context trees; pseudo-Bayesian information criterion; Model selection.



# List of Abbreviations

<b>BIC</b>	Bayesian information criterion
<b>LB</b>	lower bound
<b>MCMC</b>	Markov chain Monte Carlo
<b>MODIS</b>	Moderate Resolution Imaging Spectroradiometer
<b>MPL</b>	maximum pseudo-likelihood
<b>MRF</b>	Markov random field
<b>PCN</b>	probabilistic context neighborhood
<b>PIC</b>	pseudo-Bayesian information criterion
<b>PST</b>	probabilistic suffix tree
<b>SFTP</b>	Secure File Transfer Protocol
<b>UB</b>	upper bound
<b>UFMG</b>	Universidade Federal de Minas Gerais
<b>VLMC</b>	variable length Markov chain
<b>VNRF</b>	variable-neighborhood random field

# Contents

<b>1</b>	<b>Introduction</b>	<b>10</b>
<b>2</b>	<b>Background and Motivation</b>	<b>12</b>
2.1	Markov Chains . . . . .	12
2.1.1	Variable Length Markov Chain (VLMC) . . . . .	12
2.1.2	Probabilistic Context Tree (PCT) Model . . . . .	13
2.1.3	Model Selection for PCTs . . . . .	14
2.2	Markov Random Fields (MRFs) . . . . .	15
2.2.1	Variable-neighborhood Random Field (VNRF) . . . . .	16
2.2.2	Model Selection for MRFs . . . . .	16
<b>3</b>	<b>Probabilistic Context Neighborhood (PCN) Model</b>	<b>18</b>
3.1	Definitions and Notations . . . . .	18
3.2	Illustrating a PCN $\mathcal{T}$ . . . . .	20
3.3	Main Results . . . . .	22
3.4	PCN algorithm . . . . .	24
<b>4</b>	<b>Simulation Study</b>	<b>28</b>
4.1	Generating samples . . . . .	28
4.2	Estimating a PCN $\mathcal{T}_0$ . . . . .	29
4.2.1	Simulation 1: First-order PCN $\mathcal{T}_0$ . . . . .	29
4.2.2	Simulation 2: Variable-neighborhood PCN $\mathcal{T}_0$ with $d(\mathcal{T}_0) = 2$ . . . . .	30
4.2.3	Simulation 3: Second-order PCN $\mathcal{T}_0$ . . . . .	34
<b>5</b>	<b>Spatial Dependency of Fires in the Pantanal Biome</b>	<b>38</b>
5.1	MODIS Data . . . . .	38
5.2	Data Treatment . . . . .	39
5.3	Results . . . . .	42
5.3.1	PCN $\hat{\mathcal{T}}$ . . . . .	42
5.3.2	Building Interval Estimates via Bootstrap . . . . .	43

<b>6 Conclusion</b>	<b>47</b>
<b>Bibliography</b>	<b>49</b>
<b>APPENDIX A - Proof of Proposition 3.9</b>	<b>54</b>
<b>APPENDIX B - Simulation 3 Results</b>	<b>55</b>

## CHAPTER 1

# Introduction

Markov random field (MRF) theory represents a broad class of models used to describe data interaction behavior. In the MRF framework, the probability of a random variable is conditioned on its neighbors, following the well-known Markovian property. Due to its generality, it has been utilized in a large variety of applications to model time dependence, spatial dependence or even space-time dependence.

One of the main applications of this methodology is in image analysis and remote sensing. The knowledge of pixel interactions can be used to recover images (Geman and Geman, 1984), to segment images (Kim and Yang, 1995), to correctly classify images (Subudhi et al., 2014; Zhang et al., 2017), and to synthesize images (Wu et al., 2016). But MRF models are, by no means, limited to computer vision and geostatistics applications.

In biology, MRF can be used to model the interaction of genes. In Wei and Li (2007), for example, an MRF-based procedure has been used to identify subnetworks related to breast metastasis or death from breast cancer. Lin et al. (2015), on the other hand, study brain development using MRF to understand how brain regions are affected by neighboring brain regions, as well as time.

In economics, MRF models can be used to study the interaction of individuals, households and financial institutions as it was done in Onural et al. (2021). Fahrmeir and Lang (2001) take a different approach and use MRF to study the spatial influence of districts in Germany in their unemployment rates.

Another field where MRF has become increasingly popular is in social networks. This methodology has been used to model person-to-person interactions taking into account sentiment analysis and the general social network structure in West et al. (2014). A more commercial application uses an MRF method in recommendation systems. In Peng et al. (2016), neighboring profiles are utilized to recommend new users or new items.

As it can be seen, the list of MRF applications is incredibly long. More details on this model and its applications can be seen in Kindermann and Snell (1980), while a more modern overview is given in Hernández-Lemus (2021). For the specific case of Gaussian Markov random fields, we direct the reader to Rue and Held (2005).

In our work, interest lies in studying the extent of spatial dependence of MRF processes

in lattices in  $\mathbb{Z}^2$ . More specifically, we study the probabilistic context neighborhood (PCN) model proposed by Piroutek (2013), which gives a consistent estimator for the tree dependency structure of an MRF. As shown in Frank and Strauss (1986), assumptions about the dependency structure of a graph can lead to various modeling strategies. However, unlike the graphs considered in their work, the graphs in Piroutek (2013) are not random. The PCN model evaluates the interaction of lattices that have a fixed structure of nodes and edges. The randomness lies on the tree dependency structure and its conditional probabilities.

The task of estimating parameters of an MRF is usually approached using *potentials*, but that is not the case for the PCN model, or the models proposed by Löcherbach and Orlandi (2011) and Csiszár and Talata (2006a). Instead, the MRF specification is given in terms of the probability of a site conditioned on its neighborhood configuration. The size of the neighborhood that determines the conditional probability of a site, will be called *context*, and it may vary from site to site (we will explain this definition in more detail later on).

In Löcherbach and Orlandi (2011), the authors find a consistent estimator for the radius of the smallest ball containing the *context*. Additionally, they yield the explicit upper bound for the probability of wrong estimation, and provide an algorithm to calculate this estimator.

In Csiszár and Talata (2006a), a model selection criterion called pseudo-Bayesian information criterion (PIC) is introduced. Using PIC, a consistent estimator is found for the minimal region that determines the conditional probability of an MRF. But despite the theoretical results, the authors leave open how to calculate the estimator in practice.

The PCN model accomplishes that exact task for lattices in  $\mathbb{Z}^2$ . Rather than directly estimating the minimal neighborhood, as it was done in Löcherbach and Orlandi (2011) and Csiszár and Talata (2006a), the PCN model gives a consistent estimator for the tree source of a sample. An algorithm is proposed by cleverly combining PIC and a modification of a pruning procedure for context trees in the one-dimensional case given in Csiszár and Talata (2006b). This algorithm provides the means for easy and relatively fast implementation of the PCN model.

The goal of the present thesis is to further study the PCN model, paying special attention to the application of the algorithm proposed. We conducted a simulation study, as well as a real-world application study of the dependency structure of a process, assuming the existence of an underlying MRF. The simulation results for black and white images confirm the adequacy of the algorithm proposed in recovering the tree structure generating the process. An empirical study of the estimated conditional probabilities also suggests that the PCN model can provide reasonable estimates.

Our work is organized as follows. In [Chapter 2](#), we briefly introduce important concepts and results that laid the foundations for the PCN model presented in [Chapter 3](#). A simulation study and its results are provided in [Chapter 4](#). In [Chapter 5](#), we show the study of the spatial dependency of fires in the Pantanal biome located in Brazil that occurred in September of 2020. Finally, we conclude with our final remarks in [Chapter 6](#).

## CHAPTER 2

# Background and Motivation

We present in this chapter a few methodologies that address (at some capacity) the problem of parameter estimation and model selection in the Markov framework. First, we consider the simpler case of one-dimensional data and explore some results that have been utilized in the literature for Markov chains. Then, we introduce the concept of a Markov random field and a few existing results related to it. Our aim is to show what has been proposed, but also the gaps left unresolved which the PCN model seeks to fill.

## 2.1 Markov Chains

Let us consider a stationary ergodic stochastic process  $\{Y_j : -\infty < j < +\infty\}$  with finite alphabet  $E$ . The cardinality of the alphabet is denoted by  $|E| < \infty$ . We use the capital letters  $Y$  to refer to the random variables, whereas the lowercase letters  $y$  for their fixed deterministic values. A string  $s = y_m y_{m+1} \dots y_n$  (with  $y_j \in E$ ,  $m \leq j \leq n$ ) is also denoted by  $y_m^n$ . The string's length is given by  $l(s) = n - m + 1$  and the concatenation of strings  $u$  and  $v$  is denoted by  $uv$ .

We say a process is a Markov chain of order  $k$  if

$$P(Y_0 = y_0 \mid Y_{-\infty}^{-1} = y_{-\infty}^{-1}) = P(Y_0 = y_0 \mid Y_{-k}^{-1} = y_{-k}^{-1})$$

for all  $y_0, y_{-1}, y_{-2}, \dots$ .

Thus, a  $k$ -order Markov chain depends on the previous  $k$  variables of the past, instead of the entire past history. Therefore, there are a total of  $|E|^k(|E| - 1)$  free parameters (transition probabilities) in a Markov chain of fixed order  $k$ . Clearly, as the order dependency  $k$  grows, the number of model parameters increases exponentially fast in  $k$ .

### 2.1.1 Variable Length Markov Chain (VLMC)

From the estimation point of view, Markov chains of fixed order can be problematic. For illustrative purposes, let us consider a sequence of nitrogenous bases in a string of RNA,  $E = \{A, C, G, U\}$  and  $|E| = 4$ . As shown in [Table 2.1](#), the dimensions of the problem can become

**TABLE 2.1** Number of free parameters in a Markov chain of order  $k$  for  $|E| = 4$ .

Order $k$	1	2	3	4	5	...	10
Number of parameters	12	48	192	768	3072	...	3145728

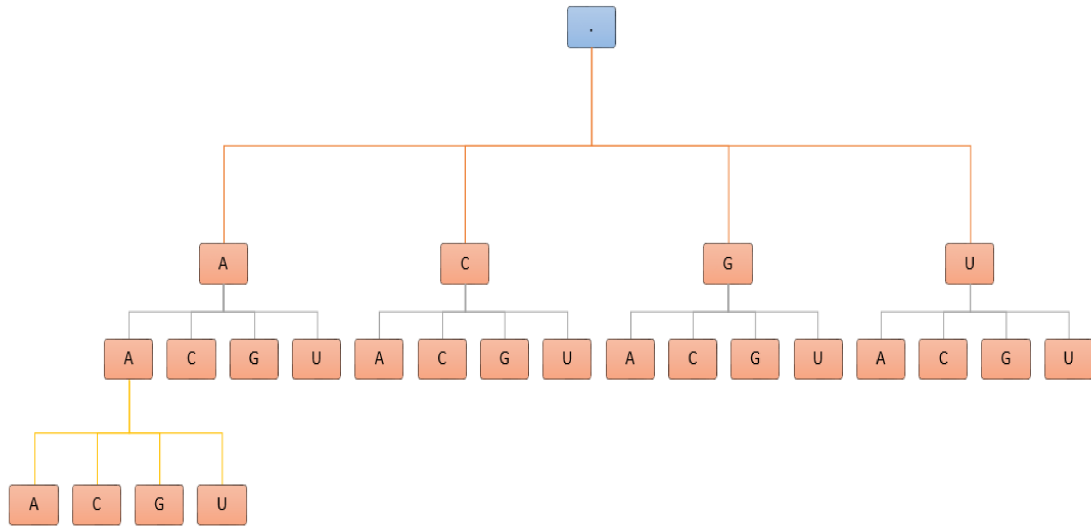
intractable as the order dependency increases. There is also a big jump in the number of parameters to be estimated from one order to another. A model called variable length Markov chain (VLMC) coined by Bühlmann and Wyner (1999) and initially proposed by Rissanen (1983), offers a more efficient representation of Markov dependency.

The VLMC model, as the name suggests, is a Markov chain that depends on a variable length of lagged values. The relevant past that influences the next outcome is called *context*. A context may be short or long depending on the length of the string needed to determine the conditional probability of the next symbol. By only storing the minimal states, there is a reduction on the number of parameters in a VLMC model compared to a full order Markov chain. Going back to the RNA example in Table 2.1, a model with 57 parameters is not possible when using a fixed order Markov chain. It is either a Markov chain of order 2 with 48 parameters, or a Markov chain of order 3 with 192 parameters. The VLMC model framework allows a number of parameters outside of this “all or nothing” approach. Processes belonging to the VLMC class are still Markovian but with memory of variable length, producing a class of models that is structurally larger and richer than Markov chains of fixed order. It can be easily seen that, if all variables  $Y_j$  ( $-\infty < j < +\infty$ ) depend on  $k$  prior values (which is equivalent to saying that all contexts have length  $k$ ), we have the general case of a full Markov chain of order  $k$ .

### 2.1.2 Probabilistic Context Tree (PCT) Model

The notion of a context was first introduced by Rissanen (1983) in information theory. In his work, the set of all contexts (allowed to be of variable length) was represented as the set of leaves of a rooted tree. This model will be addressed in this work as probabilistic context tree (PCT) model, but it is referred to in the literature in many ways, such as probabilistic suffix tree (PST), VLMC, finite memory sources, among other names.

Figure 2.1 exemplifies a PCT of order 3 in our RNA example where  $|E| = 4$ . It also shows that a tree representation offers easy interpretability of the dependency structure of a process. Clearly, a full Markov chain would require more parameters to accommodate the longer memory needed in one “direction”. In this example, only 4 contexts have length 3 while 15 other contexts have length 2, totaling 19 contexts. Completing the leaves for a full tree would result in a tree with 64 contexts. Evidently, the PCT model is very beneficial from a data compression standpoint, but other applications in biology (Bejerano and Yona, 2001; Busch et al., 2009) and linguistics (Galves et al., 2012) have shown the value of this methodology to real-life applications.



**FIGURE 2.1** Illustrative example of a probabilistic context tree model of order 3 for the dependency structure of nitrogenous bases in a string of RNA ( $|E| = 4$ ).

Besides the novel concept of only considering the relevant past, perhaps the biggest contributions of Rissanen’s work was the proposal of the *algorithm context* to estimate the true context tree given a finite sample. The true PCT, denoted by  $\mathcal{T}_0$ , contains the minimal set of strings needed in order to completely specify the probability of the next symbol. Several studies have built on this idea either improving the results of the original paper (Bühlmann and Wyner, 1999; Duarte et al., 2006; Garivier and Leonardi, 2011), or modifying the original algorithm (Willems et al., 1995; Martin et al., 2004). Finding the true PCT through information criteria was thought to be computationally infeasible by Bühlmann and Wyner (1999), because it would require the comparison of a very large number of hypothetical trees. The work of Csiszár and Talata (2006b) proves that it is indeed possible using the clever use of tree techniques.

### 2.1.3 Model Selection for PCTs

The Bayesian Information Criterion (BIC) of Schwarz (1978), was proven to be a consistent estimator for the order of a Markov chain in Csiszár and Shields (2000). Yet, it wasn’t until Csiszár and Talata (2006b) that BIC was proven to provide a strongly consistent estimator for  $\mathcal{T}_0$ . Finiteness and completeness of the true PCT were not required. By strong consistency, we mean that the estimated PCT denoted by  $\hat{\mathcal{T}}_{BIC}$  equals  $\mathcal{T}_0$  eventually almost surely as  $n \rightarrow \infty$ .

Model selection via information criteria, such as BIC, usually works by assigning scores to the different model possibilities. Then, the optimal model is chosen by minimizing the score (or maximizing, depending on the criterion formulation). As we have seen, the PCT class is very large and it would not be possible to calculate BIC over all possible tree configurations. In Csiszár and Talata (2006b), a consistent estimator was obtained by finding the tree that min-



imizes the BIC score, given in [Definition 2.1](#), over a set hypothetical PCTs allowed to grow with sample size  $n$  as  $o(\log n)$ . They also state that replacing  $\frac{(|E|-1)}{2}$  in Equation (2.1) by any  $c > 0$  does not affect the results. In addition to the providing the proof, the authors proposed an algorithm that computes this estimator in linear time.

**DEFINITION 2.1** Given a sample  $y_1^n$ , the Bayesian information criterion (BIC) of a feasible tree  $\mathcal{T}$  is

$$BIC_{\mathcal{T}}(y_1^n) = -\log ML_{\mathcal{T}}(y_1^n) + \frac{(|E|-1)|\mathcal{T}|}{2} \log n \quad (2.1)$$

where  $ML_{\mathcal{T}}$  is the maximum likelihood and  $(|E|-1)|\mathcal{T}|$  is the number of free parameters when the tree  $\mathcal{T}$  is complete. Logarithms are to the base  $e$ .

The findings in Csiszár and Talata (2006b) served as inspiration for the proposal of the PCN model and algorithm for lattices in  $\mathbb{Z}^2$  by Piroutek (2013). However, since the BIC formula (2.1) uses the maximum likelihood, BIC is considered inappropriate for high-dimensional problems where the likelihood function cannot be explicitly calculated. An alternative criterion will be introduced in the following section.

We refer the reader to Talata (2005) for a review of model selection using information criteria. For the problem of model selection specifically for PCTs, an earlier work addressing this issue was given by Bühlmann (2000). More recently, Garivier and Leonardi (2011) gave an overview of context tree selection and the different modifications of the algorithm context that were proposed in the literature.

## 2.2 Markov Random Fields (MRFs)

Let us now consider the general case of a  $d$ -dimensional lattice  $\mathbb{Z}^d$ . The points  $i \in \mathbb{Z}^d$  are called sites. The cardinality of a set  $\Delta \subset \mathbb{Z}^d$  is denoted as  $|\Delta|$ . We denote by  $\Subset$  and  $\subset$  the inclusion and strict inclusion, respectively. Subsets of  $\mathbb{Z}^d$  will be denoted by uppercase Greek letters. Thus, if  $\Lambda$  is a finite set of sites, then  $\Lambda \Subset \mathbb{Z}^d$ .

A random field is a family of random variables indexed by the site  $i$  of a lattice,  $\{X(i) : i \in \mathbb{Z}^d\}$ , where each  $X(i)$  is a random variable that takes values in a finite alphabet  $A$ . We denote the set of all configurations of the random field as  $\Omega = A^{\mathbb{Z}^d}$ . For realizations of  $X(\Delta)$ , we use the notation  $a(\Delta) = \{a(i) \in A : i \in \Delta\}$ .

The joint distribution of  $X(i)$  is given by:

$$Q(a(\Delta)) = P(X(\Delta) = a(\Delta)),$$

for  $\Delta \subset \mathbb{Z}^d$  and  $a(\Delta) \in A^{\Delta}$ .

And the conditional probability is defined by:

$$Q(a(\Delta) \mid a(\Phi)) = P(X(\Delta) = a(\Delta) \mid X(\Phi) = a(\Phi))$$

for all disjoint regions  $\Delta$  and  $\Phi$  where  $Q(a(\Phi)) > 0$ .

We say that the process is a Markov random field (MRF) if there exists a neighborhood  $\Gamma_i$ , satisfying for every  $i \in \mathbb{Z}^d$

$$P(X(i) = a(i) \mid X(\mathbb{Z}^d \setminus i) = a(\mathbb{Z}^d \setminus i)) = P(X(i) = a(i) \mid X(\Gamma_i) = a(\Gamma_i)), \quad (2.2)$$

where a neighborhood  $\Gamma_i$  (of the site  $i$ ) means a finite, central-symmetric set of sites with  $i \notin \Gamma_i$ .

### 2.2.1 Variable-neighborhood Random Field (VNRF)

If estimation of a Markov chain can be challenging as the order dependency grows, the problem of estimating parameters of a Markov random field is even more complicated. In an attempt to minimize this issue, the variable-neighborhood random field (VNRF) model was created in Löcherbach and Orlandi (2011), generalizing the concept of a VLMC to random fields in  $\mathbb{Z}^d$ .

Like the VLMC model explained in [Section 2.1.1](#), the VNRF model also works with the idea of *contexts*. Here, a context is the minimal neighborhood needed to determine the probability a site. The depth of the neighborhood changes according to the values in them. Hence, the VNRF model is defined by a family of conditional probabilities that do not depend on a fixed neighborhood depth. In Löcherbach and Orlandi (2011) the focus was on estimating the radius that contains the minimal neighborhood of a site. They do not address the problem of estimating the geometrical structure of the context as they claim it would introduce too many parameters. Similarly, Csiszár and Talata (2006a) offer a consistent estimator for the context neighborhood of a site. Their paper, however, is mainly concerned with the proposal of a model selection criterion for MRFs since penalized likelihood estimators cannot be used.

### 2.2.2 Model Selection for MRFs

Analogous to BIC, the pseudo-Bayesian information criterion (PIC) was proposed in Csiszár and Talata (2006a) to address the problem of model selection in MRFs. The likelihood in BIC was replaced by the pseudo-likelihood introduced by Besag (1975). Due to phase transition on multidimensional lattices, a unique invariant measure is not assured so a likelihood approach is not suitable. A similar criterion was proposed earlier by Ji and Seymour (1996) and recently, Pensar et al. (2017) introduced a small sample analytical version of PIC. The evaluation of the best model selection criteria for MRFs is beyond the scope of this work, we will only focus on the definition and results related to PIC.

**DEFINITION 2.2** Given a sample  $x(\Lambda_n)$ , the pseudo-Bayesian information criterion (PIC) of a neighborhood  $\Gamma$  is:

$$PIC_{\Gamma}(x(\Lambda_n)) = -\log MPL_{\Gamma}(x(\Lambda_n)) + |A|^{|\Gamma|} \log |\Lambda_n| \quad (2.3)$$

where  $MPL_{\Gamma}$  is the maximum pseudo-likelihood,  $\Lambda_n$  is the sample region, and  $n$  is the number of sites in the sample.

Csiszár and Talata (2006a) proved that minimizing PIC over a family of hypothetical neighborhoods resulted in an estimate that equaled the true context neighborhood eventually almost surely as  $n \rightarrow \infty$ . The radius of the possible neighborhoods were allowed to grow with the sample size as  $o((\log |\Lambda_n|)^{\frac{1}{2d}})$ . This result is unaffected by phase transition and non-stationarity of the joint distribution. Also, the result remains valid if the penalty term  $|A|^{|\Gamma|}$  in Equation (2.3) is replaced by any  $c > 0$ .

The problem, however, is that no algorithm was proposed to actually compute the PIC estimator  $\hat{\Gamma}_{PIC}$ . This happened for two reasons. First, no simple formula is available for  $|A|^{|\Gamma|}$  in Equation (2.3), because the candidate neighborhoods do not have a specific geometry. The only requirement is that the neighborhood of a site  $i$ , denoted by  $\Gamma_i$ , is a finite central-symmetric set of sites with  $i \notin \Gamma_i$ . The term  $|A|^{|\Gamma|}$  replaced half the “number of free parameters” in BIC’s Equation (2.1). The second reason is that, even if it could be calculated, the authors did not find a way to compute the PIC score for all possible neighborhood configurations without calculating them one by one. Consequently, they leave it open if the PIC estimator can be computed in a “clever way”, as it was done in the one-dimensional case.

That is precisely what the PCN model proposed by Piroutek (2013) does for lattices in  $\mathbb{Z}^2$ . By setting a fixed neighborhood geometry and representing the dependency structure as a tree, the PCN model is a two-dimensional version of a PCT. Consequently, the PCN algorithm is a modified version of the PCT algorithm in Csiszár and Talata (2006b), using PIC instead of BIC to find the optimal tree.

## CHAPTER 3

# Probabilistic Context Neighborhood (PCN) Model

The probabilistic context neighborhood model proposed by Piroutek (2013) offers a tree representation to an MRF process on lattices in  $\mathbb{Z}^2$ . The purpose of this model is to provide insight on the dependency of sites on their neighbors, through learning the dependency structure, as well as estimating the conditional probabilities that determine the value of a site.

### 3.1 Definitions and Notations

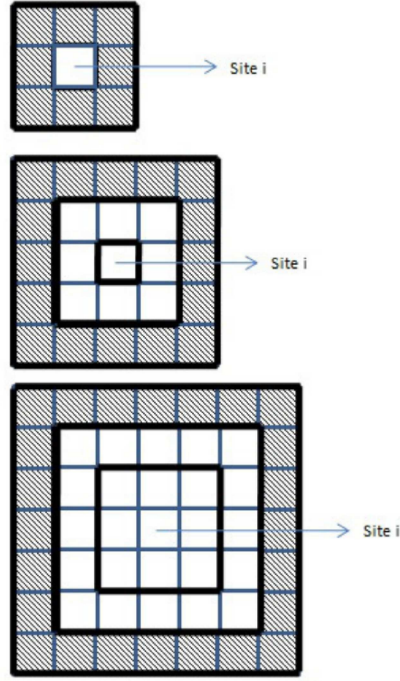
In this chapter, we will continue to use the notations and definitions introduced in [Section 2.2](#) for Markov random fields in lattices in  $\mathbb{Z}^d$  for the specific case where  $d = 2$ . However, an important aspect of the PCN model is that the neighborhood geometry  $\Gamma_i$  in Equation (2.2) is set to a *frame*, denoted by  $\partial_i^j$ , as defined in [Definition 3.1](#).

**DEFINITION 3.1** A frame  $\partial_i^j$ , with order  $j \in \mathbb{N}$ , is a particular type of neighborhood for a site  $i$ . It can be obtained by taking a square of side  $2j + 1$ , and removing a smaller square of side  $2j - 1$  contained within it, both centered on  $i$ .

[Figure 3.1](#) provides an example of frames of order 1, 2 and 3. Larger orders can be understood analogously. It can be easily seen that, for  $j = 1, 2, \dots, m$ , the frames  $\partial_i^j$  are nested sets.  $\bigcap_{j=1}^m \partial_i^j = \emptyset$  and  $\bigcup_{j=1}^m \partial_i^j$  is a square region of the lattice with side  $2m + 1$  and center on site  $i$ . Since the geometry of the neighborhood is fixed and to simplify the notation, we will write  $\partial^j$ , omitting the site  $i$  whenever it is clear.

We denote the union of frames  $\bigcup_{s=m}^n \partial^s = (\partial^m \partial^{m+1} \dots \partial^n)$  as  $\partial^{m, \dots, n}$ , with  $m < n$ . The length of a frame is represented as  $l(\partial^{m, \dots, n}) = n - m + 1$ . For simplicity, the concatenation of the first frame with all the higher order frames until the  $j^{\text{th}}$  frame, given by  $\partial^{1, \dots, j}$ , will be denoted as  $\mathcal{D}^j$ . The length of  $\mathcal{D}^j$  is  $l(\mathcal{D}^j) = j$  and equals the order of the neighborhood  $\mathcal{D}^j$ .

We say that a configuration  $a(\partial_i^j)$  is a realization of the process on the subset  $\partial_i^j$ . The concatenation of two configurations  $a(\partial^{1, \dots, k})$  and  $a(\partial^{m, \dots, n})$  is  $a(\partial^{1, \dots, n})$ , or  $a(\mathcal{D}^n)$ , and is only possible if  $m = k + 1$ . The cardinality of a neighborhood, denoted by  $|a(\mathcal{D}^n)|$ , indicates the



**FIGURE 3.1** Frame structure  $\partial_i^j$  for  $j = 1, 2$  and  $3$ , respectively.

number of sites within a neighborhood of order  $n$ .

**DEFINITION 3.2** A configuration  $a(\mathcal{D}^k)$  is a *suffix* of  $a(\mathcal{D}^n)$ ,  $k \leq n$ , if  $a(\mathcal{D}^n)$  is a concatenation of  $a(\partial^{1,\dots,k})$  and  $a(\partial^{k+1,\dots,n})$ . This induces an order in the space of configurations and we say that  $a(\mathcal{D}^n) \succeq a(\mathcal{D}^k)$ . If the cardinality  $|a(\partial^{k+1,\dots,n})| > 0$ , then  $a(\mathcal{D}^k)$  is a *proper suffix* of  $a(\mathcal{D}^n)$ .

A set of neighborhood configurations can be represented as a neighborhood tree  $\mathcal{T}$ . It has the root on top, characterizing the value of a site (identified as  $\emptyset$ ), and branches connected to it, growing downwards. The first set of nodes stemming from the root is the first-order neighborhood configurations  $\partial^1$ . The *children* of those nodes are the second-order neighborhood frames containing the *parent* neighborhood frame inside, that is  $\partial^{1,2}$  or simply  $\mathcal{D}^2$ . The third set of nodes are the children of the second order nodes, given by  $\mathcal{D}^3$ . The same logic is valid for higher-order nodes. A neighborhood configuration  $a(\mathcal{D}^j) \in \mathcal{T}$  represents a *leaf* of the neighborhood tree. The leaves correspond to the last nodes of each of the branches connected to the root. Therefore, an internal node of  $\mathcal{T}$  is a proper suffix of a leaf.

As stated in [Section 2.2](#), all possible configurations of a random field  $\{X(i), i \in \mathbb{Z}^2\}$ , that take values in a finite alphabet  $A$ , are given by  $\Omega = A^{\mathbb{Z}^2}$ . Therefore, the number of possible neighborhood configurations of order 1 in the PCN model is given by  $A^{|\mathcal{D}^1|}$ . The number of possible configurations of a neighborhood of order 2 is  $A^{|\mathcal{D}^2|}$  and so on. Hence, the formal definition of a neighborhood tree  $\mathcal{T}$  is given below.

**DEFINITION 3.3** A subset  $\mathcal{T} \subset \cup_{j=1}^{\infty} A^{|\mathcal{D}^j|}$  is called a neighborhood tree if no  $a(\mathcal{D}^k) \in \mathcal{T}$  is a suffix of any other  $a(\mathcal{D}^n) \in \mathcal{T}$ .

The depth of a neighborhood tree  $\mathcal{T}$  represents the maximum order of neighborhoods belonging to that tree and is denoted by  $d(\mathcal{T}) = \max_j \{a(\mathcal{D}^j) \in \mathcal{T}\}$ .

If not a single neighborhood  $a(\mathcal{D}^j)$  belonging to the neighborhood tree  $\mathcal{T}$  can be replaced by a proper suffix without violating the tree property, then the neighborhood tree is considered irreducible. The set of irreducible neighborhood trees is denoted by  $\mathcal{I}$ .

Although the neighborhood geometry is fixed in a frame format, the order of the neighborhood needed to determine the probability of a site can still vary. Thus, the PCN model utilizes the VNRF framework and the notion of contexts as specified in [Definition 3.4](#).

**DEFINITION 3.4** A finite configuration  $a(\mathcal{D}^j) \in A^{|\mathcal{D}^j|}$  is a context neighborhood of a Markov random field if  $Q(a(\mathcal{D}^j)) > 0$  and

$$\begin{aligned} P(X(i) = a(i) \mid X(\mathbb{Z}^2 \setminus i) = a(\mathbb{Z}^2 \setminus i)) &= P(X(i) = a(i) \mid X(\mathcal{D}^j) = a(\mathcal{D}^j)) \\ &= Q(a(i) \mid a(\mathcal{D}^j)) \end{aligned} \quad (3.1)$$

for every  $a(i) \in A$ , and no proper suffix of  $a(\mathcal{D}^j)$  has this property.

Therefore, if  $a(\mathcal{D}^j)$  is a context neighborhood of a site  $i$ , then the probability distribution of that site depends only on  $a(\mathcal{D}^j)$ . There is no need to inspect the entire lattice to acquire information about the value assumed by  $X(i)$ . We say that  $j$ , which is the number of frames in the configuration  $a(\mathcal{D}^j)$ , is the *order* of the context neighborhood.

Clearly, the set of all context neighborhoods of a process can be represented as a context neighborhood tree and we will denote it by  $\mathcal{T}_0$ . Let  $Q_0 = \{Q(a(i) \mid a(\mathcal{D}^j)) : a(i) \in A, a(\mathcal{D}^j) \in \mathcal{T}_0\}$  be the family of transition probabilities satisfying Equation (3.1). The pair  $(\mathcal{T}_0, Q_0)$  is called *probabilistic context neighborhood* or PCN.

The goal of the PCN model is, given a finite sample  $a(\Lambda_n)$  of a lattice in  $\mathbb{Z}^2$ , to estimate the PCN  $(\mathcal{T}_0, Q_0)$  that generated the sample. In order to do so, the PIC score of Csiszár and Talata (2006a) is used to compare a set of hypothetical PCNs  $(\mathcal{T}, Q)$  to reach the true PCN  $(\mathcal{T}_0, Q_0)$  that generated the sample under study.

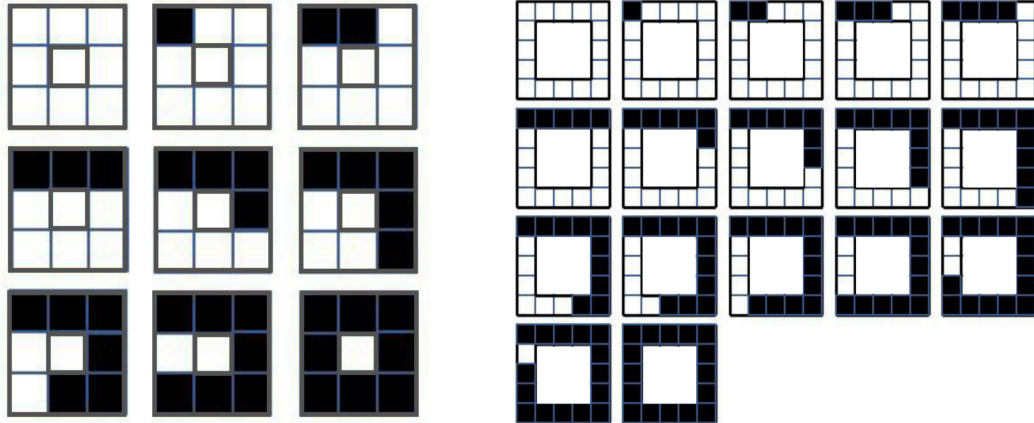
From now on, for simplicity, we refer to the PCN  $(\mathcal{T}, Q)$  only as  $\mathcal{T}$ .

## 3.2 Illustrating a PCN $\mathcal{T}$

This section is dedicated to exemplifying the concepts and ideas defined in [Section 3.1](#). We focus on the space of binary states due to its simplicity and because it allows the interesting study of black and white images. An extension to larger state spaces is straightforward.

Let  $A = \{-1, 1\}$ , where  $X(i) = -1$ , if the value of site  $i$  is white, and  $X(i) = 1$  if it is black.

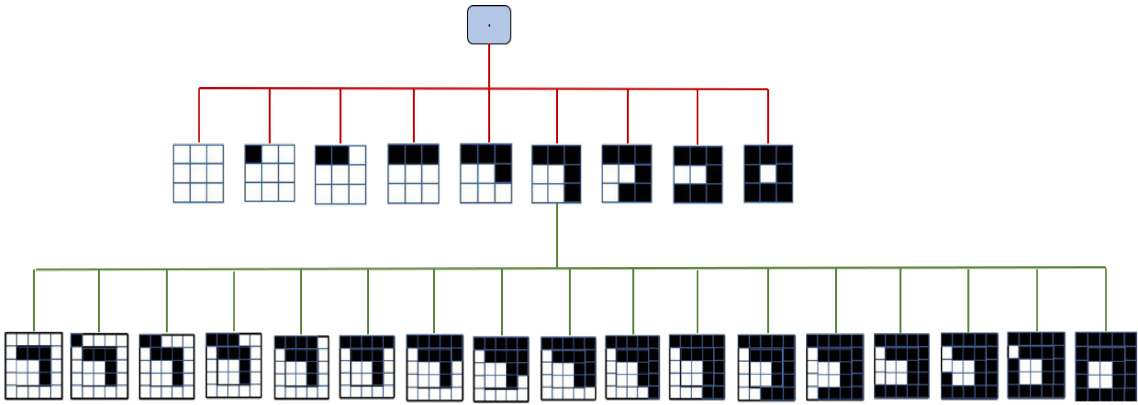
We consider two neighborhood configurations to be equivalent if each neighborhood con-



(a) All possible configurations of first-order frames  $a(\partial^1)$  for  $A = \{-1, +1\}$ .

(b) All possible configurations of second-order frames  $a(\partial^2)$  for  $A = \{-1, 1\}$ .

**FIGURE 3.2** All possible configurations of first and second-order frames for black and white images.



**FIGURE 3.3** Illustrative example of a PCN  $\mathcal{T}$  for  $|A| = 2$  and  $d(\mathcal{T}) = 2$ .

tains the same number of black and white sites, independently of their position.

Figure 3.2 shows the possible neighborhood configurations for frames of order 1 and 2, respectively. It can be seen that, a frame  $\partial^1$  is made of 8 sites, that is,  $|\partial^1| = 8$ . Therefore, in the case of black and white images, there are 9 total possible configurations of first-order frames. The first frame can have zero black sites, all the way up to 8 black sites. In the case of frames  $\partial^2$ , there are 16 sites within it ( $|\partial^2| = 16$ ), which translates into 17 possible second-order frame configurations (varying from zero black sites all the way up to 16 black sites). Generalizing, the  $j^{\text{th}}$ -order frame has a total of  $8j$  sites within it and  $8j + 1$  possible configurations.

The frame neighborhood geometry proposed by Piroutek (2013) makes it possible to represent the contexts of a MRF process in a tree format, similar to the PCT model introduced and exemplified in Section 2.1.2. A hypothetical PCN  $\mathcal{T}$  for  $A = \{-1, 1\}$  is shown in Figure 3.3.

The PCN root drawn on top of the tree represents the value of the site  $i$ . The first generation nodes (children) are drawn from the root down and represent the first-order neighborhoods. If



the information contained within the first-order frame is insufficient to provide a conditional probability for the site  $i$ , then the second-order neighborhood is drawn adding a frame of order 2 to this first-order neighborhood. The new neighborhood drawn is connected to the parent neighborhood. Each generation in the tree represents an added frame to the parent generation. The PCN tree continues to grow until all the context neighborhoods are added.

In the example showed in [Figure 3.3](#), the contexts of the PCN tree have variable neighborhood length. There are 8 contexts of order 1 and 17 contexts of order 2. For each context neighborhood, a conditional probability of the central site being black (or white) is assigned as in [Definition 3.4](#). All first-order frames are considered contexts, except for the first frame with 5 black sites in it. This means that, if we observe only one black site in the first-order neighborhood (or 0, 2, 3, 4, 6, 7 and 8 black sites), it will be sufficient to determine the probability of the site  $i$  being black. However, if there are 5 black sites in the first frame, we must continue “down” the PCN and look at the configurations of the second-order frame. All 17 child configurations of the first-frame with 5 black sites are considered contexts. In summary, this hypothetical PCN  $\mathcal{T}$  has depth  $d(\mathcal{T}) = 2$ , a total of 25 contexts neighborhoods (or leaves) and 1 internal node.

### 3.3 Main Results

We have explained and illustrated the neighborhood geometry and tree representation of an MRF process in the PCN model. This section will be focused on the main theoretical results of Piroutek’s work that led to the proposal of a consistent estimator for a PCN  $\mathcal{T}_0$  from a sample  $a(\Lambda_n)$  containing the  $n$  sites under study.

As it was explained in [Section 2.2.2](#), a likelihood approach is not suited for MRF problems. Therefore, we use the pseudo-Bayesian information criterion of Csiszár and Talata (2006a) to select the optimal PCN  $\mathcal{T}$ . This is achieved by replacing the likelihood by the pseudo-likelihood introduced in Besag (1975) and defined below.

**DEFINITION 3.5** Given a sample  $a(\Lambda_n)$ , the pseudo-likelihood function associated with a PCN  $(\mathcal{T}, Q)$  is defined by:

$$PL_{\mathcal{T}}(a(\Lambda_n)) = \prod_{a(\mathcal{D}^j) \in \mathcal{T}, N_n(a(\mathcal{D}^j)) \geq 1} \prod_{a(i) \in A} Q(a(i) | a(\mathcal{D}^j))^{N_n(a(\mathcal{D}^j, i))},$$

where

$$N_n(a(\mathcal{D}^j, i)) = |\{i \in a(\Lambda_n) : a(\mathcal{D}_i^j) \subset a(\Lambda_n), a(\mathcal{D}_i^j \cup i) = a(\mathcal{D}_i^j, i)\}|$$

represents the number of times that the configuration  $a(\mathcal{D}^j)$  is observed in the sample when the site  $i$  assumes the value  $a(i)$  and

$$N_n(a(\mathcal{D}^j)) = |\{i \in a(\Lambda_n) : a(\mathcal{D}_i^j) \subset a(\Lambda_n)\}|$$



is the number of occurrences of the configuration  $a(\mathcal{D}^j)$  in the sample  $a(\Lambda_n)$ .

According to Csiszár and Talata (2006a), the maximum pseudo-likelihood is obtained for:

$$\hat{Q}(a(i)|a(\mathcal{D}^j)) = \frac{N_n(a(\mathcal{D}^j, i))}{N_n(a(\mathcal{D}^j))}$$

Therefore, given a sample  $a(\Lambda_n)$ , the maximum pseudo-likelihood (MPL) for a PCN  $\mathcal{T}$  is:

$$MPL_{\mathcal{T}}(a(\Lambda_n)) = \prod_{a(\mathcal{D}^j) \in \mathcal{T}, N_n(a(\mathcal{D}^j)) \geq 1} \prod_{a(i) \in A} \left( \frac{N_n(a(\mathcal{D}^j, i))}{N_n(a(\mathcal{D}^j))} \right)^{N_n(a(\mathcal{D}^j, i))} \quad (3.2)$$

Since we are interested in estimating the PCN  $\mathcal{T}_0$ , instead of the neighborhood  $\Gamma$ , Piroutek (2013) modified the PIC formula in Equation (2.3) to be closer to the BIC formula for PCTs in Equation (2.1), replacing the maximum likelihood by the maximum pseudo-likelihood.

**DEFINITION 3.6** Given a sample  $a(\Lambda_n)$ , the pseudo-Bayesian information criterion (PIC) for a PCN  $\mathcal{T}$  is:

$$PIC_{\mathcal{T}}(a(\Lambda_n)) = -\log MPL_{\mathcal{T}}(a(\Lambda_n)) + \frac{(|A| - 1)|\mathcal{T}|}{2} \log |\Lambda_n| \quad (3.3)$$

An important difference between the definition above and [Definition 2.2](#) is the term that precedes  $\log |\Lambda_n|$ . Because the neighborhood structure in Csiszár and Talata (2006a) was not fixed, it was unfeasible to compute the term  $|A|^{|\Gamma|}$ . In the PCN model, however, the fixed frame geometry for the neighborhoods allows the computation of  $|\mathcal{T}|$ , which represents the number of leaves of a PCN tree or simply the number of neighborhood contexts  $a(\mathcal{D}^j) \in \mathcal{T}$ .

In our work, we obtained a closed formula for  $|\mathcal{T}|$  considering full trees instead of VNRFs, as given in [Proposition 3.7](#).

**PROPOSITION 3.7** *If frames are considered equivalent by having the same combination of elements in  $A$  within a frame, then  $|\mathcal{T}|$  is, at most,*

$$\prod_{k=1}^{d(\mathcal{T})} \binom{8k + |A| - 1}{|A| - 1}. \quad (3.4)$$

*Conversely, if we consider that the position of each site within the frame matters, then  $|\mathcal{T}|$  is, at most,*

$$|A|^{|\mathcal{D}^k|} = |A|^{\frac{8k(k+1)}{2}}, \text{ where } k = d(\mathcal{T}).$$

**Proof:** See Appendix A. ■

Consequently, the PCN model solves the first issued mentioned at the end of [Section 2.2.2](#). We are able to compute the PIC score for a given PCN  $\mathcal{T}$  since we can calculate the penalizing

term in Equation (3.3).

Given a sample  $a(\Lambda_n)$ , a feasible PCN  $\mathcal{T}$  is such that  $d(\mathcal{T}) \leq D(n)$ , where  $D(n)$  is an appropriate function of the sample size. Also, for every  $a(\mathcal{D}^j) \in \mathcal{T}$ ,  $N_n(a(\mathcal{D}^j)) \geq 1$ . We say that  $a(\mathcal{D}^k)$  is a suffix of some  $a(\mathcal{D}^j) \in \mathcal{T}$  if  $k \leq j$  and  $N_n(a(\mathcal{D}^k)) \geq 1$ . The family of feasible PCNs is denoted by  $\mathcal{F}_1(a(\Lambda_n), D(n))$ .

**DEFINITION 3.8** We define the PIC estimator for a PCN  $\mathcal{T}_0$  as

$$\hat{\mathcal{T}}_{PIC}(a(\Lambda_n)) = \arg \min_{\mathcal{T} \in \mathcal{F}_1(a(\Lambda_n), D(n)) \cap \mathcal{I}} PIC_{\mathcal{T}}(a(\Lambda_n)), \quad (3.5)$$

In other words, the PIC estimator for a PCN  $\mathcal{T}_0$  is the PCN  $\mathcal{T}$  that minimizes the PIC score among all feasible PCNs allowed to grow with the sample size.

**THEOREM 3.9** Let  $\{X(i) : i \in \mathbb{Z}^2\}$  be a PCN with finite tree  $\mathcal{T}_0$  such that  $\hat{Q}(a(i)|a(\mathcal{D}^j))$  is a consistent estimator of  $Q(a(i)|a(\mathcal{D}^j))$ . Then

$$\hat{\mathcal{T}}_{PIC}(a(\Lambda_n)) \rightarrow \mathcal{T}_0$$

almost surely as  $n \rightarrow \infty$ .

The consistency of the estimator  $\hat{\mathcal{T}}_{PIC}$  in [Theorem 3.9](#) was proven in Piroutek et al. provided that  $D(n) = (\log |\Lambda_n|)^{\frac{1}{4}}o$ . The mathematical proof is beyond the scope of this work. Instead, we will focus on the application of the PCN algorithm introduced in [Section 3.4](#).

Note that Csiszár and Talata (2006a) prove the consistency of the PIC estimator  $\hat{\Gamma}_{PIC}$  for any unstructured neighborhood  $\Gamma_i$  of site  $i$  as long as the neighborhood is a finite central-symmetric set of sites and it does not contain the site under evaluation. The authors also prove that the empirical estimator  $\hat{Q}(a(i)|a(\hat{\Gamma}_{PIC}))$  converges to the true conditional probability almost surely as  $n \rightarrow \infty$ . In their work, the question of “how to find the PIC estimator without computing the score for all possibilities?” was not answered.

### 3.4 PCN algorithm

Calculating PIC for all feasible PCNs  $\mathcal{T}$  would be impractical and time consuming. Since the PCN model represents the context neighborhoods of an MRF in a tree format, similar to the PCT model, Piroutek (2013) proposes a PCN algorithm similar to the one initially proposed by Csiszár and Talata (2006b) for the one-dimensional case. The pruning procedure in the PCN algorithm makes it possible to obtain  $\hat{\mathcal{T}}_{PIC}$ .

We are interested in obtaining the PIC estimator for PCN  $\mathcal{T}$ , given that [Theorem 3.9](#) proves that  $\hat{\mathcal{T}}_{PIC}$  converges to  $\mathcal{T}_0$  almost surely as  $n \rightarrow \infty$ .

Given a sample  $a(\Lambda_n)$ , all trees considered are denoted by  $\mathcal{F} = \mathcal{F}_1(a(\Lambda_n), D(n)) \cap \mathcal{I}$  and

using [Definitions 3.8](#) and [3.6](#), we have that:

$$\begin{aligned}
\hat{\mathcal{T}}_{PIC}(a(\Lambda_n)) &= \arg \min_{\mathcal{T} \in \mathcal{F}} PIC_{\mathcal{T}}(a(\Lambda_n)) \\
&= \arg \min_{\mathcal{T} \in \mathcal{F}} \left\{ -\log MPL_{\mathcal{T}}(a(\Lambda_n)) + \frac{(|A|-1)|\mathcal{T}|}{2} \log |\Lambda_n| \right\} \\
&= \arg \max_{\mathcal{T} \in \mathcal{F}} \left\{ \log MPL_{\mathcal{T}}(a(\Lambda_n)) - \frac{(|A|-1)|\mathcal{T}|}{2} \log n \right\} \\
&= \arg \max_{\mathcal{T} \in \mathcal{F}} \left\{ \log MPL_{\mathcal{T}}(a(\Lambda_n)) + \log n^{-\frac{(|A|-1)|\mathcal{T}|}{2}} \right\} \\
&= \arg \max_{\mathcal{T} \in \mathcal{F}} \left\{ \log \left[ n^{-\frac{(|A|-1)|\mathcal{T}|}{2}} MPL_{\mathcal{T}}(a(\Lambda_n)) \right] \right\} \\
&= \arg \max_{\mathcal{T} \in \mathcal{F}} \left\{ n^{-\frac{(|A|-1)|\mathcal{T}|}{2}} MPL_{\mathcal{T}}(a(\Lambda_n)) \right\}
\end{aligned}$$

We can factorize the maximum pseudo-likelihood function in Equation (3.2) as:

$$MPL_{\mathcal{T}}(a(\Lambda_n)) = \prod_{a(\mathcal{D}^j) \in \mathcal{T}} \tilde{P}_{MPL, \mathcal{D}^j}(a(\Lambda_n)),$$

where

$$\tilde{P}_{MPL, \mathcal{D}^j}(a(\Lambda_n)) = \begin{cases} \prod_{a(i) \in A} \left( \frac{N_n(a(\mathcal{D}^j, i))}{N_n(a(\mathcal{D}^j))} \right)^{N_n(a(\mathcal{D}^j, i))} & , \text{ if } N_n(a(\mathcal{D}^j)) \geq 1 \\ 1 & , \text{ if } N_n(a(\mathcal{D}^j)) = 0 \end{cases}$$

Hence, the PIC estimator  $\hat{\mathcal{T}}_{PIC}$  can be rewritten as:

$$\begin{aligned}
\hat{\mathcal{T}}_{PIC}(a(\Lambda_n)) &= \arg \max_{\mathcal{T} \in \mathcal{F}} \left\{ n^{-\frac{(|A|-1)|\mathcal{T}|}{2}} \prod_{a(\mathcal{D}^j) \in \mathcal{T}} \tilde{P}_{MPL, \mathcal{D}^j}(a(\Lambda_n)) \right\} \\
&= \arg \max_{\mathcal{T} \in \mathcal{F}} \left\{ \prod_{a(\mathcal{D}^j) \in \mathcal{T}} n^{-\frac{|A|-1}{2}} \tilde{P}_{MPL, \mathcal{D}^j}(a(\Lambda_n)) \right\} \\
&= \arg \max_{\mathcal{T} \in \mathcal{F}} \left\{ \prod_{a(\mathcal{D}^j) \in \mathcal{T}} \tilde{P}_{\mathcal{D}^j}(a(\Lambda_n)) \right\}, \tag{3.6}
\end{aligned}$$

where  $\tilde{P}_{\mathcal{D}^j}(a(\Lambda_n)) = n^{-\frac{|A|-1}{2}} \tilde{P}_{MPL, \mathcal{D}^j}(a(\Lambda_n))$ .

The steps of the PCN algorithm are as follows. First, we construct a tree using all neighborhood configurations observed within a sample  $a(\Lambda_n)$ , where the neighborhood length is at most  $D = D(n)$ . The set of nodes of this tree will be denoted by  $\mathcal{N}_D$ . Then,  $\tilde{P}_{\mathcal{D}^j}(a(\Lambda_n))$  is calculated for each neighborhood  $a(\mathcal{D}^j) \in \mathcal{N}_D$ . Using  $\tilde{P}_{\mathcal{D}^j}(a(\Lambda_n))$ , a value  $V_{\mathcal{D}^j}^D(a(\Lambda_n))$  is assigned to each node. Subsequently, this value is utilized to create a binary indicator denoted by  $\chi_{\mathcal{D}^j}^D(a(\Lambda_n))$ .

This assignment is recursive, starting from the leaves of the tree, moving to the parent nodes, all the way “up” to the root. The indicators  $\chi_{\mathcal{D}^j}^D(a(\Lambda_n))$  will stipulate where to prune the tree to arrive at the desired PIC estimator  $\hat{\mathcal{T}}_{PIC}$ .

**DEFINITION 3.10** Given a sample  $a(\Lambda_n)$ , each neighborhood  $a(\mathcal{D}^j) \in \mathcal{N}_D$  receives recursively, from the leaves of the tree, the value

$$V_{\mathcal{D}^j}^D(a(\Lambda_n)) = \begin{cases} \tilde{P}_{\mathcal{D}^j}(a(\Lambda_n)) & , \text{ if } j = D \\ \max \left\{ \tilde{P}_{\mathcal{D}^j}(a(\Lambda_n)) , \prod_{a(\mathcal{D}^{j+1}): N_n(a(\mathcal{D}^{j+1})) \geq 1} V_{\mathcal{D}^{j+1}}^D(a(\Lambda_n)) \right\} & , \text{ if } 0 \leq j < D \end{cases}$$

and the indicator

$$\chi_{\mathcal{D}^j}^D(a(\Lambda_n)) = \begin{cases} 0, & \text{if } j = D \\ 0, & \text{if } \tilde{P}_{\mathcal{D}^j}(a(\Lambda_n)) \geq \prod_{a(\mathcal{D}^{j+1}): N_n(a(\mathcal{D}^{j+1})) \geq 1} V_{\mathcal{D}^{j+1}}^D(a(\Lambda_n)) \text{ and } 0 \leq j < D \\ 1, & \text{if } \tilde{P}_{\mathcal{D}^j}(a(\Lambda_n)) < \prod_{a(\mathcal{D}^{j+1}): N_n(a(\mathcal{D}^{j+1})) \geq 1} V_{\mathcal{D}^{j+1}}^D(a(\Lambda_n)) \text{ and } 0 \leq j < D \end{cases}$$

where  $a(\mathcal{D}^{j+1})$  represents the children of the parent neighborhood  $a(\mathcal{D}^j)$ .

Based on the indicators  $\chi_{\mathcal{D}^j}^D(a(\Lambda_n))$ , a maximizing tree  $\mathcal{T}_{\mathcal{D}^j}^D(a(\Lambda_n))$  comprised of neighborhoods  $a(\mathcal{D}^u) \succeq a(\mathcal{D}^j)$  is assigned to each neighborhood  $a(\mathcal{D}^j) \in \mathcal{N}_D$ . **Lemma 3.13** will later clarify why the term “maximizing” was used.

**DEFINITION 3.11** Given  $a(\mathcal{D}^j) \in \mathcal{N}_D$ , let  $\mathcal{T}_{\mathcal{D}^j}^D(a(\Lambda_n))$  equal to

$$\begin{cases} a(\mathcal{D}^j) & , \text{ if } \chi_{\mathcal{D}^j}^D(a(\Lambda_n)) = 0 \\ \{a(\mathcal{D}^u) \in \mathcal{N}_D : \chi_{\mathcal{D}^u}^D(a(\Lambda_n)) = 0, \chi_{\mathcal{D}^v}^D(a(\Lambda_n)) = 1, \text{ for all } j \leq v < u\} & , \text{ if } \chi_{\mathcal{D}^j}^D(a(\Lambda_n)) = 1 \end{cases}$$

It follows that the maximizing neighborhood tree  $\mathcal{T}_{\mathcal{D}^j}^D(a(\Lambda_n))$  is irreducible unless it equals  $a(\mathcal{D}^j)$ .

The following proposition and lemma were stated and proved in Piroutek et al..

**PROPOSITION 3.12** *The probabilistic context neighborhood tree estimator  $\hat{\mathcal{T}}_{PIC}(a(\Lambda_n))$  equals the maximizing tree assigned to the root. That is,*

$$\hat{\mathcal{T}}_{PIC}(a(\Lambda_n)) = \mathcal{T}_{\emptyset}^D(a(\Lambda_n))$$

**LEMMA 3.13** *For any  $a(\mathcal{D}^j) \in \mathcal{N}_D$ ,*

$$V_{\mathcal{D}^j}^D(a(\Lambda_n)) = \max_{\mathcal{T} \in \mathcal{F}_1(a(\Lambda_n)|a(\mathcal{D}^j))} \prod_{a(\mathcal{D}^u) \in \mathcal{T}} \tilde{P}_{\mathcal{D}^u}(a(\Lambda_n)) = \prod_{a(\mathcal{D}^u) \in \mathcal{T}_{\mathcal{D}^j}^D(a(\Lambda_n))} \tilde{P}_{\mathcal{D}^u}(a(\Lambda_n))$$

where  $\mathcal{F}_1(a(\Lambda_n)|a(\mathcal{D}^j))$  is defined as the family of all trees  $\mathcal{T}$  of depth  $d(\mathcal{T}) \leq D$  consisting of configurations  $a(\mathcal{D}^u) \succeq a(\mathcal{D}^j)$  with  $N_n(a(\mathcal{D}^u)) \geq 1$ .

In other words, the maximizing tree assigned to the root  $\mathcal{T}_\emptyset^D(a(\Lambda_n))$  is the tree, among all the feasible trees, that maximizes the product  $\prod_{a(\mathcal{D}^j) \in \mathcal{T}} \tilde{P}_{\mathcal{D}^j}(a(\Lambda_n))$  in Equation (3.6). Therefore,  $\mathcal{T}_\emptyset^D(a(\Lambda_n)) = \hat{\mathcal{T}}_{PIC}$ .

The maximizing tree assigned to the root (or equivalently, the PIC estimator for PCN  $\mathcal{T}_\emptyset$ ) can be obtained by pruning the tree containing all configurations that belong to the sample  $a(\Lambda_n)$ , as determined by [Definition 3.11](#). Unlike the assignment of values  $V_{\mathcal{D}^j}^D(a(\Lambda_n))$  and indicators  $\chi_{\mathcal{D}^j}^D(a(\Lambda_n))$ , the pruning procedure is done starting from the root of the tree and moving “down” the branches. The indicator  $\chi_{\mathcal{D}^j}^D(a(\Lambda_n))$  determines where to prune the tree. If an indicator equals to zero, we keep that specific node and exclude the children configurations connected to it. Alternatively, if the indicator of a node equals one, we continue “down” to the children configurations until we observe an indicator equal to zero. That procedure is executed for all the branches connected to the root. So, after the pruning procedure is finalized, the resulting tree has internal nodes with indicator equal to one and all the leaves have indicator equal to zero.

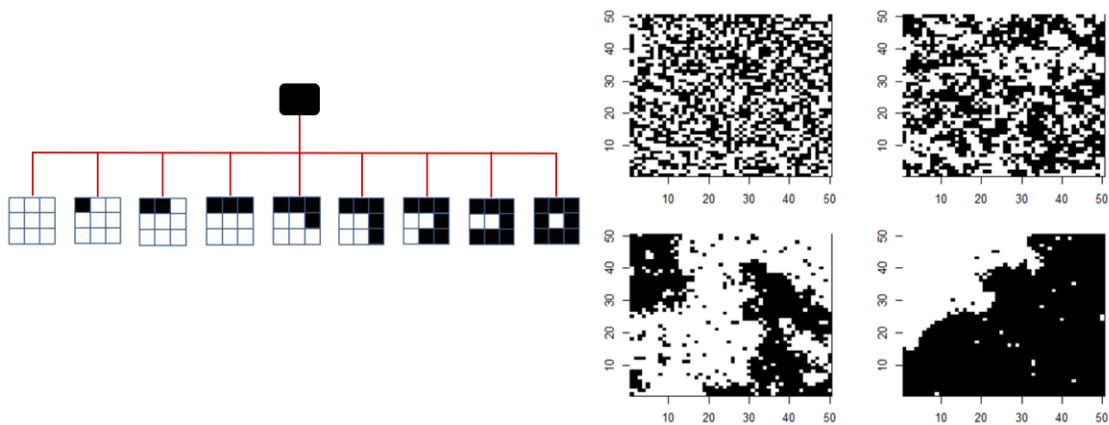
## CHAPTER 4

# Simulation Study

The purpose of this chapter is to validate the PCN algorithm explained in [Section 3.4](#). To do so, we conducted a simulation study for three different scenarios using the statistical software R (R Core Team, 2020). We seek to compare the estimated trees obtained through the PCN algorithm with the original trees which generated the sample.

Our simulations are based on a regular lattice with black and white sites. We borrow the notation used in [Section 3.2](#), considering  $A = \{-1, 1\}$  where  $a(i) = -1$ , if the observed value of site  $i$  is white, and  $a(i) = 1$  if it is black. Since  $|A| = 2$ , we have complementary events and determining the conditional probability of a site being black suffices to determine the conditional probability of it being white. In addition, we also consider frames to be equivalent if they have the same number of black sites within it, just as in the example provided in [Section 3.2](#).

### 4.1 Generating samples



**FIGURE 4.1** Left: Probabilistic context neighborhood tree structure. Right: Lattice simulations generated from the PCN structure on the left. For each black and white image shown on the right side, the tree structure was the same, the only variation was in the conditional probabilities assigned to the leaves.

In order to generate samples with a predefined spatial dependency, we first determined the PCN  $\mathcal{T}_0$ 's structure and the conditional probabilities associated with each leaf. The same PCN

tree structure can create different images when the conditional probabilities of each context neighborhood differ, as shown in [Figure 4.1](#).

Sampling is done using a Markov chain Monte Carlo (MCMC) method. Starting from a random configuration of black and white sites, we evaluate each site individually. A conditional probability of being black is attributed to a site based on its neighbors, as dictated by the PCN tree  $\mathcal{T}_0$ . An acceptance step, similar to the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970), is then used to determine whether the site under evaluation is black or white. Once this procedure is done for all sites, we have completed the first iteration. We perform iterations until the image “stabilizes”. From that point on, we consider that the process has converged to the target distribution.

In the simulations presented here the sites were inspected line by line, one column at a time. We experimented inspecting sites randomly within the lattice and did not notice any differences in computational time or time until convergence.

To evaluate the sites located on the boundaries, we mirrored the lattice first horizontally, then vertically. This step was later proved to be highly time-consuming and could be further improved. A better approach would be to generate a larger sample matrix and evaluate a smaller sample contained within it, eliminating the border correction problem entirely. This option was only considered after working with a real-world dataset in [Chapter 5](#). Since the simulation study presented in this chapter was performed first, the results shown here used the mirrored matrix approach.

## 4.2 Estimating a PCN $\mathcal{T}_0$

In this section, we present the estimated PCN trees obtained through the PCN algorithm proposed by Piroutek (2013). For each of the three scenarios, a black and white image was simulated from a given PCN  $\mathcal{T}_0$  as described in the previous section.

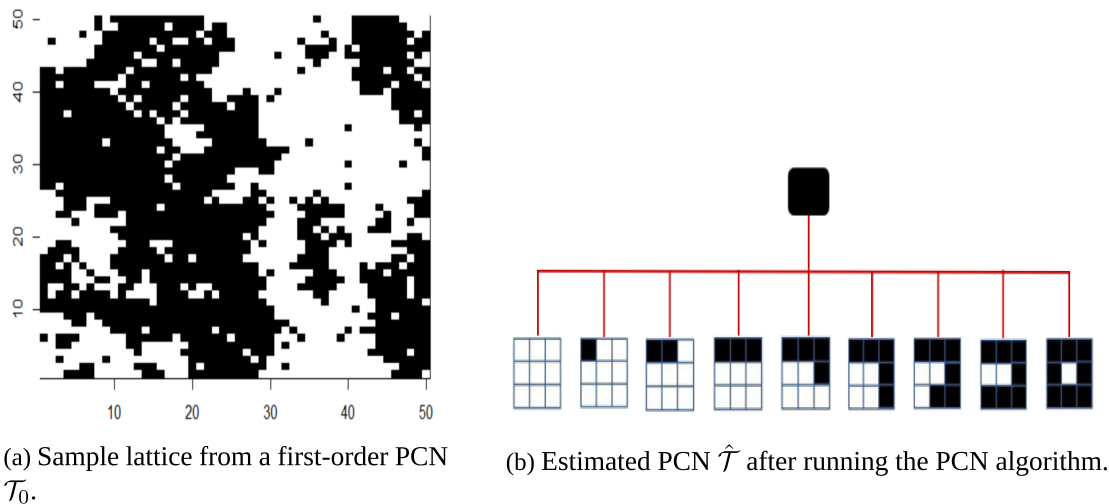
### 4.2.1 Simulation 1: First-order PCN $\mathcal{T}_0$

The first simulation was based on a simple first-order PCN  $\mathcal{T}_0$  as shown in the left side of [Figure 4.1](#). A  $50 \times 50$  lattice was obtained after 50 iterations of the sampling algorithm and can be seen in [Figure 4.2a](#).

After the PCN algorithm was run, the estimated tree recovered exactly the same tree structure used to generate the sample. The estimated PCN tree is given in [Figure 4.2b](#).

[Table 4.1](#) presents the comparison between the transition probabilities of the true PCN  $\mathcal{T}_0$  and the estimated PCN  $\hat{\mathcal{T}}$ .

To better quantify the uncertainty associated with the estimated conditional probabilities, we built an interval for these estimates. We are not aware of results for Markov random fields in two dimensions which guarantee specific properties to such intervals.



**FIGURE 4.2** Simulation results for first-order PCN tree.

The estimated intervals were obtained from the following steps. First, we generated a sample of 100 matrices from  $\mathcal{T}_0$ , created after 100 iterations of the sampling algorithm. Then, we ran the PCN algorithm to select only the matrices that recovered the true tree structure. In this case, all 100 matrices estimated a first-order PCN tree. Lastly, we used all of the simulated matrices to calculate the 2.5<sup>th</sup> percentile, median and 97.5<sup>th</sup> percentile of the sample's conditional probabilities.

In Hyndman and Fan (1996), different sample quantile formulations are analyzed based on six distinct properties. The authors conclude by recommending that the median-unbiased estimator is used. In our work, we follow their recommendation which is equivalent to selecting the argument `type = 8` in the `quantile` function in R.

Using the default argument in Simulation 1 did not appear to make a difference. However, we noticed a few discrepancies in other studies presented henceforth. The differences were observed for conditional probabilities of neighborhoods which appeared in few samples and the frequency counts were very low (between 1 and 10) within the samples in which they did appear.










The results containing the estimated intervals for Simulation 1 are presented in Table 4.2. The real conditional probabilities are contained in all 9 intervals provided. The median, as expected, provides a better point estimate for the majority of context neighborhoods compared to the single matrix estimate in Table 4.1.

#### 4.2.2 Simulation 2: Variable-neighborhood PCN $\mathcal{T}_0$ with $d(\mathcal{T}_0) = 2$










Our second simulation is based on a variable-neighborhood PCN  $\mathcal{T}_0$  with  $d(\mathcal{T}_0) = 2$ . The source PCN  $\mathcal{T}_0$  is shown in Figure 4.3. It has 6 first-order contexts and 51 second-order contexts neighborhoods. Each internal node of this tree has 17 children, representing all possible second-order frame configurations (that vary from 0 to 16 black sites within it). This PCN tree indicates that, if there are 3 black sites in the first frame (or 4 and 5), it is necessary to look at the second-



**TABLE 4.1** Comparison between the true probability of the site being black given the context neighborhood configuration and the point estimate for the conditional probability in Simulation 1.

Context	True	Estimate
	0.0392	0.0328
	0.0832	0.0693
	0.1680	0.1582
	0.3100	0.2697
	0.5000	0.5163
	0.6900	0.7548
	0.8320	0.8333
	0.9168	0.8964
	0.9608	0.9656

**TABLE 4.2** Comparison between the true probability of the site being black given the context neighborhood configuration and the estimated interval for each conditional probability in Simulation 1. The lower bound (LB) corresponds to the 2.5<sup>th</sup> percentile and the upper bound (UB) is the 97.5<sup>th</sup> percentile.

Context	True	Interval Estimate		
		LB	Median	UB
	0.0392	0.0000	0.0274	0.0722
	0.0832	0.0184	0.0918	0.1381
	0.1680	0.0852	0.1638	0.2457
	0.3100	0.1858	0.3009	0.3900
	0.5000	0.4053	0.4954	0.5985
	0.6900	0.6069	0.6834	0.7514
	0.8320	0.7922	0.8405	0.8868
	0.9168	0.8896	0.9127	0.9391
	0.9608	0.9527	0.9637	0.9746

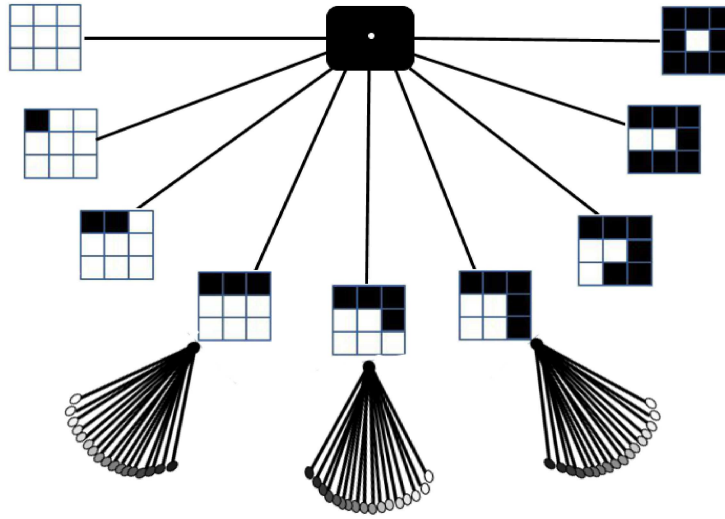


FIGURE 4.3 Variable-neighborhood PCN  $\mathcal{T}_0$  with depth  $d(\mathcal{T}_0) = 2$ .

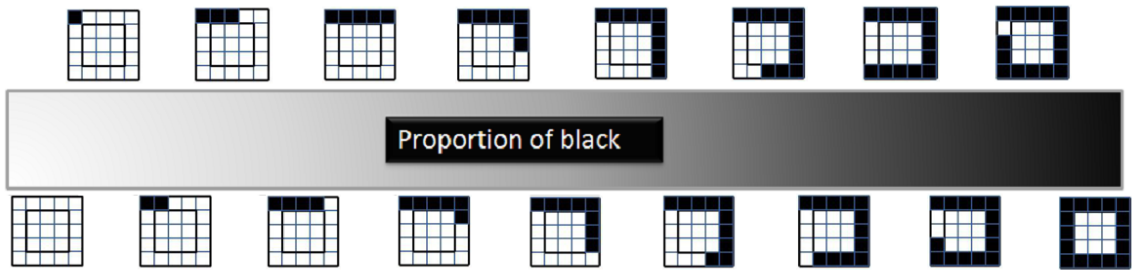


FIGURE 4.4 Grayscale representing the proportion of black sites in the second frame.

order frame configuration to determine the transition probability for the given site. Due to space limitations, we choose not to draw the second-order configurations and draw a grayscale instead. As indicated by Figure 4.4, the lighter the color, the less black sites exist in the second frame. On the other hand, the darker the color, the more black sites.























































A lattice with  $150 \times 150$  sites was created after 50 iterations of the MCMC algorithm. The resulting image is presented in Figure 4.5a. The estimated tree obtained from the pruning procedure in the PCN algorithm is given in Figure 4.5b.

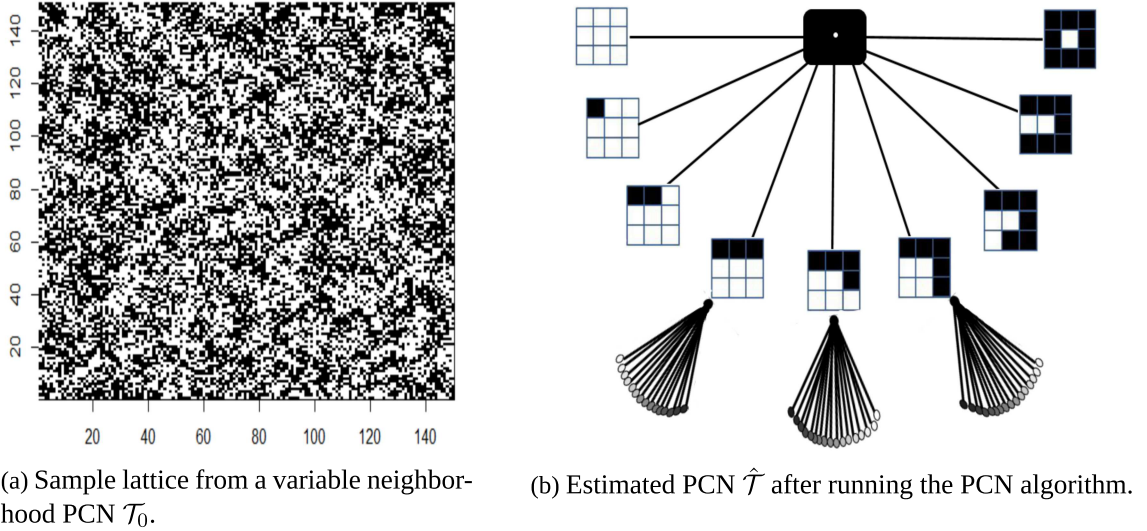
The tree structure recovered is almost identical to the original tree  $\mathcal{T}_0$  in Figure 4.3. The estimated PCN  $\hat{\mathcal{T}}$  has 6 first-order contexts, like the original tree, and 48 second-order contexts, compared to the 51 contexts in  $\mathcal{T}_0$ . The 3 missing context neighborhoods in the second order did not appear in the generated sample. This is believed to happen due to the relatively small sample size.

Table 4.3 shows the comparison between the conditional probabilities of the original tree and the estimated tree.

Using the same approach as in the previous simulation, we built intervals for the conditional

**TABLE 4.3** Comparison between the true probability of a site being black given the context neighborhood and the point estimate for the conditional probability in Simulation 2.

Context	True	Estimate	Context	True	Estimate
	0.3100	0.3844		0.5498	0.5342
	0.3543	0.3567		0.5987	0.5623
	0.4013	0.3736		0.6457	0.6034
	0.1680	0.3000		0.6900	0.6302
	0.1978	0.3103		0.7311	0.7216
	0.2315	0.2965		0.7685	0.5946
	0.2689	0.3205		0.8022	0.6000
	0.3100	0.3462		0.8320	1.000
	0.3543	0.3448		0.2315	0.1667
	0.4013	0.3889		0.2689	0.1176
	0.4502	0.4601		0.3100	0.2424
	0.5000	0.4944		0.3543	0.4043
	0.5498	0.5372		0.4013	0.4646
	0.5987	0.5633		0.4502	0.4987
	0.6457	0.5876		0.5000	0.5139
	0.6900	0.6316		0.5498	0.5822
	0.7311	0.6111		0.5987	0.5997
	0.7685	1.0000		0.6457	0.6460
	0.168	0.0000		0.6900	0.6774
	0.1978	0.3333		0.7311	0.6889
	0.2315	0.2973		0.7685	0.7153
	0.2689	0.2673		0.8022	0.6061
	0.3100	0.3515		0.8320	0.6154
	0.3543	0.3830		0.8581	1.0000
	0.4013	0.3970		0.5987	0.6118
	0.4502	0.4648		0.6457	0.6387
	0.5000	0.4908		0.6900	0.6952



**FIGURE 4.5** Simulation results for a variable-neighborhood PCN tree with  $d(\mathcal{T}_0) = 2$ .

probabilities of each context neighborhood. The 2.5<sup>th</sup> percentile, median and 97.5<sup>th</sup> percentile were computed based on a sample of 81 matrices. Out of 100 matrices generated from  $\mathcal{T}_0$  after 200 iterations of the MCMC algorithm, 81 of them recovered the original tree structure after the pruning procedure, and were used to build these intervals. This fluctuation is expected since there is an inherent variability within the tree structure as well as the conditional probabilities.

The results of the interval estimation for the conditional probabilities of Simulation 2 are presented in [Table 4.4](#).

There were only 4 instances, out of 57 estimated intervals, where the interval did not contain the true value. In these cases, the estimated interval was at most, 0.0101 away from it. The range of an interval varied depending on the number of times a context neighborhood was observed within the samples analyzed, and how many samples had that specific configuration. Due to very low frequencies for 8 context neighborhoods (appearing, less than 10 times within a matrix), the resulting interval covered the entire parametric space.

### 4.2.3 Simulation 3: Second-order PCN $\mathcal{T}_0$

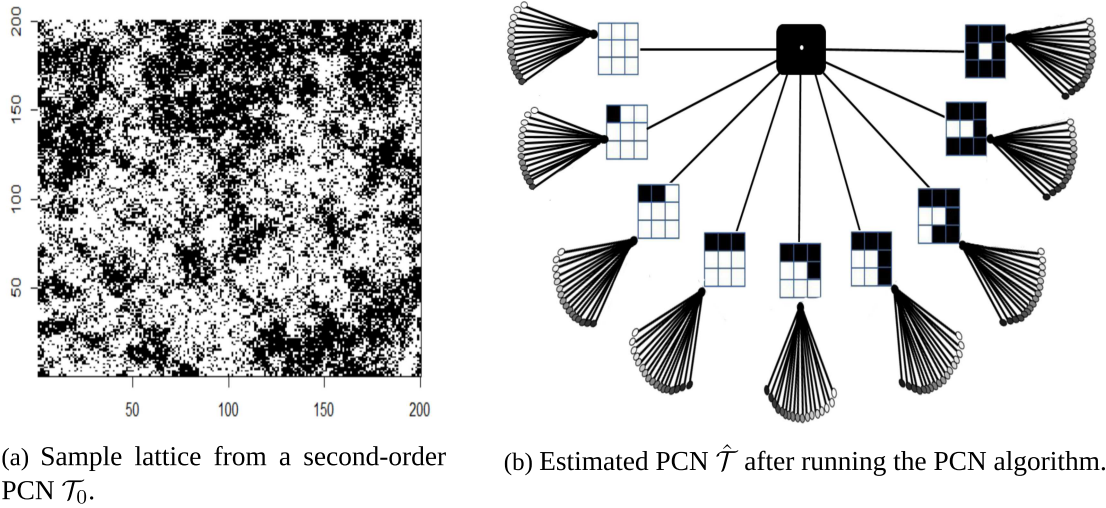
The third and final simulation was created to analyze the performance of the PCN algorithm applied to a sample of a complete second-order PCN tree  $\mathcal{T}_0$ . As given by Equation (3.4), the full second-order tree has 153 context neighborhoods. That is, each first-order node stemming from the root has 17 children nodes and all of them are considered context neighborhoods.

[Figure 4.6a](#) presents the  $200 \times 200$  matrix simulated after 100 iterations of the sampling algorithm. [Figure 4.6b](#) shows the estimated PCN tree obtained through the PCN algorithm.

As before, the structure of the estimated tree  $\hat{\mathcal{T}}$  is quite similar to the true tree  $\mathcal{T}_0$ . However, the estimated tree contains a total of 141 context neighborhoods of second order, rather than

**TABLE 4.4** Comparison between the true probability of a site being black given the context neighborhood and the estimated interval for each conditional probability in Simulation 2. The lower bound (LB) corresponds to the 2.5<sup>th</sup> percentile and the upper bound (UB) is the 97.5<sup>th</sup> percentile.

Context	True	Interval Estimate			Context	True	Interval Estimate		
		LB	Median	UB			LB	Median	UB
	0.3100	0.2524	0.3265	0.3968		0.5498	0.5095	0.5439	0.5730
	0.3543	0.3305	0.3575	0.3822		0.5987	0.5351	0.5764	0.6207
	0.4013	0.3634	0.3797	0.4075		0.6457	0.5678	0.6063	0.6468
	0.1419	0.0000	0.0000	1.0000		0.6900	0.5778	0.6407	0.7078
	0.1680	0.0000	0.3000	0.5682		0.7311	0.5588	0.6667	0.7943
	0.1978	0.1556	0.2826	0.4285		0.7685	0.5059	0.6757	0.8505
	0.2315	0.2416	0.3089	0.3982		0.8022	0.3486	0.6667	1.0000
	0.2689	0.2704	0.3297	0.3856		0.8320	0.0000	1.0000	1.0000
	0.3100	0.3030	0.3496	0.4015		0.1978	0.0000	0.0000	1.0000
	0.3543	0.3409	0.3808	0.4103		0.2315	0.0000	0.2679	1.0000
	0.4013	0.3736	0.4103	0.4512		0.2689	0.0935	0.3333	0.5837
	0.4502	0.4054	0.4446	0.4845		0.3100	0.2140	0.3542	0.5371
	0.5000	0.4327	0.4778	0.5202		0.3543	0.3260	0.4086	0.4796
	0.5498	0.4690	0.5185	0.5858		0.4013	0.3615	0.4417	0.5020
	0.5987	0.4727	0.5566	0.6253		0.4502	0.4200	0.4836	0.5399
	0.6457	0.4737	0.5966	0.6984		0.5000	0.4795	0.5188	0.5613
	0.6900	0.5076	0.6429	0.8028		0.5498	0.5153	0.5605	0.5908
	0.7311	0.4210	0.6667	0.9756		0.5987	0.5576	0.5906	0.6179
	0.7685	0.0000	0.7500	1.0000		0.6457	0.5872	0.6230	0.6631
	0.8022	0.0000	1.0000	1.0000		0.6900	0.6063	0.6481	0.6977
	0.168	0.0000	0.0000	1.0000		0.7311	0.6129	0.6759	0.7218
	0.1978	0.0000	0.2000	0.5656		0.7685	0.5985	0.7086	0.7663
	0.2315	0.1157	0.2857	0.4818		0.8022	0.5682	0.7234	0.8694
	0.2689	0.2104	0.3220	0.4435		0.8320	0.5000	0.7143	0.9771
	0.3100	0.2987	0.3557	0.4310		0.8581	0.0000	0.7083	1.0000
	0.3543	0.3220	0.3922	0.4363		0.5987	0.5973	0.6162	0.6414
	0.4013	0.3975	0.4231	0.4748		0.6457	0.6194	0.6468	0.6723
	0.4502	0.4206	0.4673	0.5071		0.6900	0.6030	0.6839	0.7427
	0.5000	0.4648	0.5032	0.5389					



**FIGURE 4.6** Simulation results for a complete second-order PCN tree.

153. Like in Simulation 2, the 12 missing contexts did not appear in the sample under study and therefore, did not show up in  $\hat{\mathcal{T}}$ . To capture all possible second-order frame configurations, a larger lattice would be necessary.

Due to the large number of leaves within this tree, we chose to omit the comparisons between the true conditional probabilities of each context neighborhood and their estimated values. The detailed results can be found in Appendix B under the “single matrix estimate” column.

We built an interval for the estimated conditional probabilities of this process, based on a sample of 50 matrices. The matrices were generated after 400 iterations of the MCMC algorithm, and selected after correctly recovering the PCN tree structure. We created 149 interval estimates for the 153 total conditional probabilities of PCN  $\mathcal{T}_0$ . Instead of intervals, we provided point estimates for 2 context neighborhoods since those configurations were each observed once inside one matrix. 2 neighborhoods did not appear in a single matrix, hence, no estimate was provided. All estimated intervals contained the true conditional probability. In 20 of them, however, the range covered the entire parametric space due to the extremely low counts for those particular context neighborhoods. For the same reason stated previously, these results are included in Appendix B.

The scenarios presented in this chapter were run using three distinct machines. We did not increase the size of the matrices in Simulations 2 and 3, due to the computational burden of this task and time constraints. Generating a single matrix in Simulation 2 and 3 took approximately 16 hours and 99 hours, respectively. Subsequently, the PCN algorithm was run in approximately 25 minutes for the matrix in Simulation 2 and 78 minutes for Simulation 3. These times were recorded for a computer with an Intel i5 processor running at 1.6 GHz and using 4GB of RAM. Creating a sample of matrices in Simulation 2 and applying the PCN algorithm to each matrix took approximately 33 hours and 31 hours, respectively. This task was performed with a more powerful machine available at UFMG’s Spatial Statistics Laboratory which has an Intel Xeon

processor running at 3.7GHz and using 128GB of RAM. Lastly, generating the sample of matrices in Simulation 3 took approximately 64 hours, while running the PCN algorithm took 53 hours. This was observed for a computer with an Intel i7 processor running at 1.3 GHz and using 12GB of RAM.



## CHAPTER 5

# Spatial Dependency of Fires in the Pantanal Biome

The previous chapter showed the adequacy of the PCN model and algorithm through simulation studies. Now we seek to demonstrate an application of this methodology to a real-world dataset.

Motivated by the record number of fire foci in the Pantanal Biome in the Center-West Region of Brazil through September of 2020 (INPE, 2021*b*), we conducted a study on the spatial dependency of fires in that region. Fires cause damage to local biodiversity, increase CO<sub>2</sub> emissions and can severely affect people's health. The PCN model can provide insight on the spatial dependency structure of this phenomena, as well as quantify the conditional probabilities of this unknown process. This type of information can be valuable to shape a more efficient fire prevention plan.

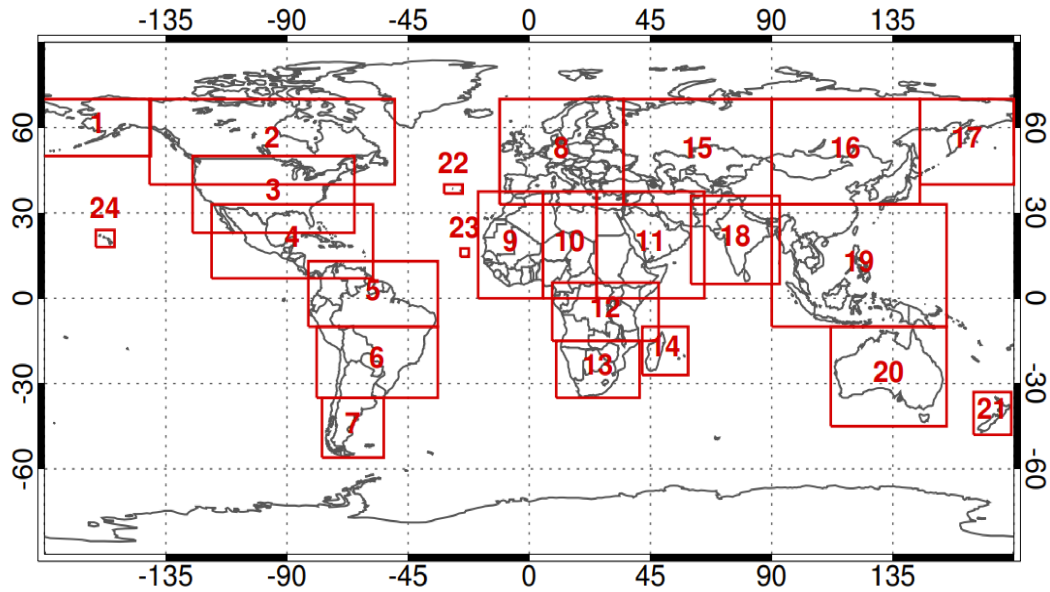
## 5.1 MODIS Data

We chose to work with NASA's Moderate Resolution Imaging Spectroradiometer (MODIS) Burned Area product due to its reliability and the fact that it is a well documented data source. The MCD64A1 Burned Area Product is a monthly and gridded 500-meter product containing burned areas per pixel. Therefore, we can evaluate the pixels in the grid as we evaluated the sites of a lattice in our simulation study in [Chapter 4](#).

All the results presented in this chapter were obtained through the MCD64A1 GeoTIFF files. These files are divided in 24 different windows as demonstrated by [Figure 5.1](#). We selected burned area product data for windows 5 and 6 regarding September of 2020. This was done by downloading the GeoTIFF files from the fuoco SFTP server as directed by the MODIS Burned Area Product User's Guide (Giglio et al., 2020).

We will disregard the temporal component of this study and focus only on its spatial aspect. The PCN model will be seen as a representation of a Markovian process for a given moment. We are interested in investigating the spatial dependency of fires in Pantanal in an unprecedented





**FIGURE 5.1** Coverage of GeoTIFF files, from Giglio et al. (2020).

time in history. September of 2020 saw 8106 fires detected by the reference satellite, compared to 2887 for the same month in the prior year. Before that, the maximum number of fire foci was 5993 recorded in August of 2005 (INPE, 2021b).

## 5.2 Data Treatment

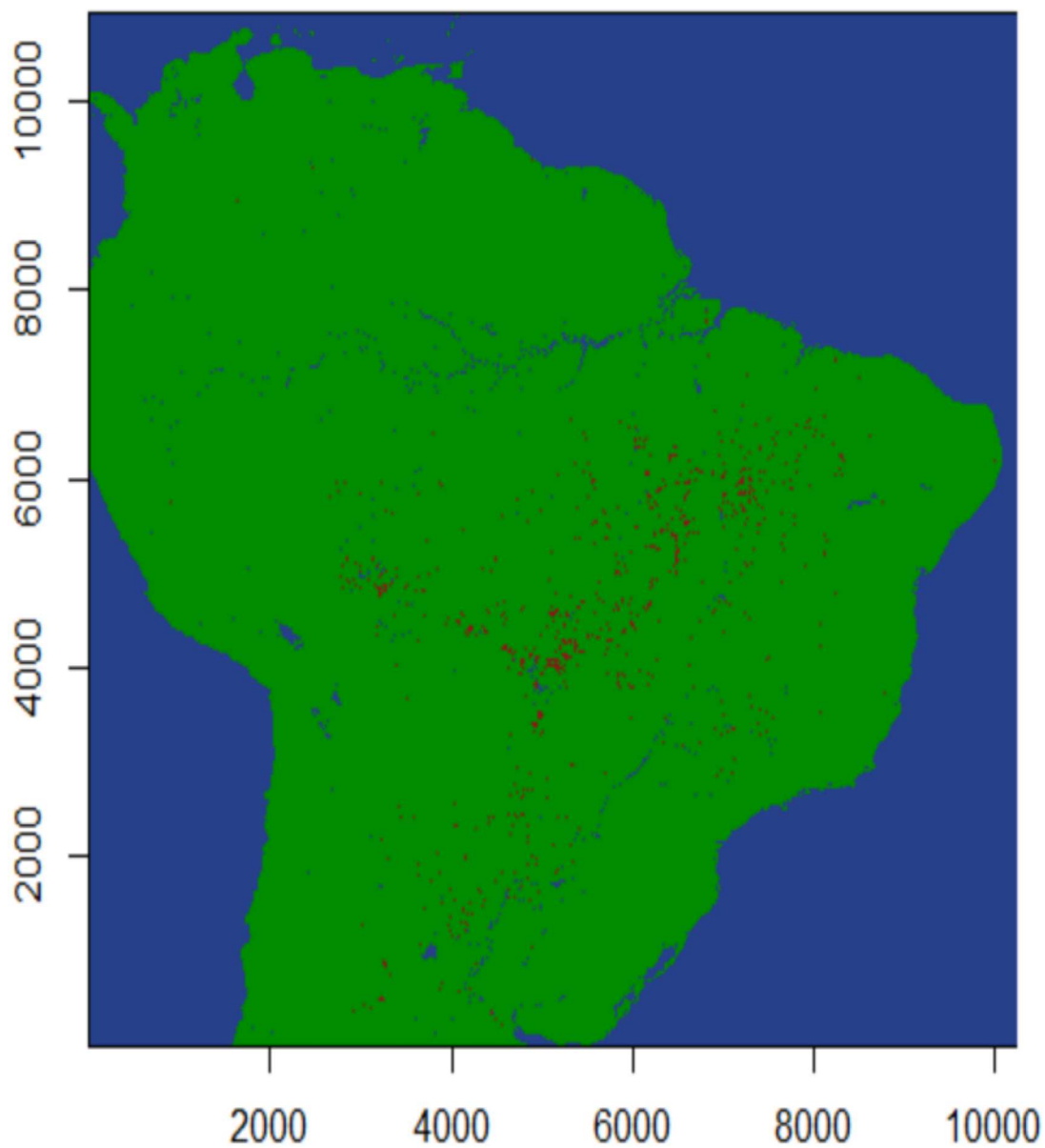
The MODIS data, as it was, could not be used in the analysis. This section will describe the steps needed before running the PCN algorithm.

First, we used the `stars` package (Pebesma, 2020) to read the GeoTIFF files into R. Then, this same package was used to transform the data into a simple matrix class object that is easier to manipulate. The values inside the matrix are classified in three distinct categories: fire, unburned land and water. These values suffered slight modifications, compared to the original version, that made them compatible with the PCN algorithm.

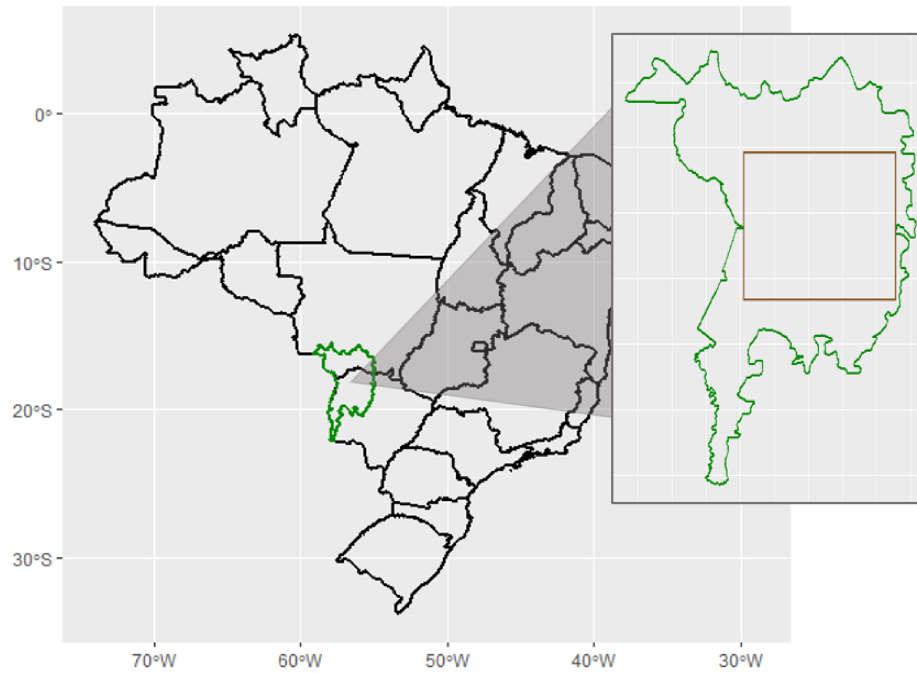
[Figure 5.2](#) displays the MCD64A1 Burned Area Product for September of 2020 corresponding to windows 5 and 6 after the above steps were performed. Each category is illustrated by a different color pixel. Fires are red, unburned land is green, and water is blue.

Next, using the `rgdal` R package (Bivand et al., 2020) and a shapefile obtained from INPE (2021a), we examined the boundaries of the Pantanal biome. Based on these geographic coordinates, we selected the largest square matrix within Pantanal to analyze. [Figure 5.3](#) shows the location of Pantanal (in green) inside the map of Brazil. The brown square inside the Pantanal boundary represents the sample under study.

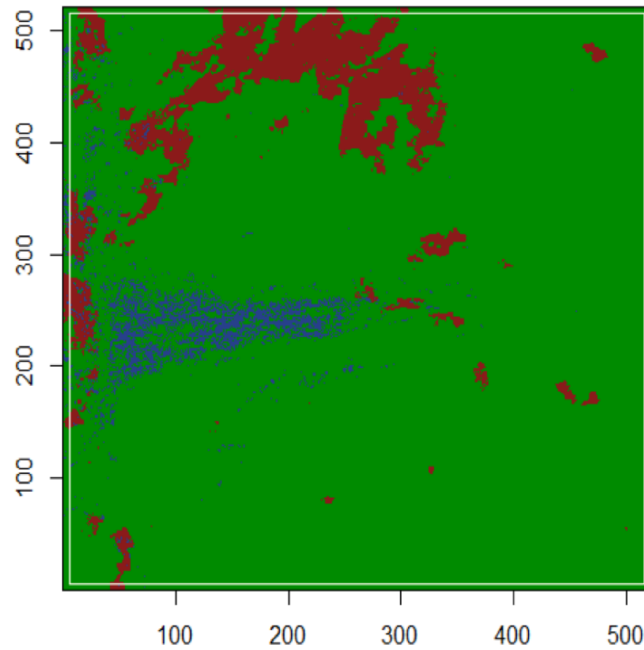
The final sample is a  $510 \times 510$  matrix as displayed in [Figure 5.4](#). There are a total of 260,100 pixels from which 230,114 are unburned land, 7,881 are water, and 22,105 are fire. Although



**FIGURE 5.2** Matrix object corresponding to the MCD64A1 Burned Area Product for windows 5 and 6 regarding September of 2020. A green pixel represents unburned land, a blue pixel corresponds to water, and a red pixel stands for fire.



**FIGURE 5.3** Map of Brazil divided by its states, created from a shapefile obtained from IBGE. The Pantanal biome boundary is represented in green. The brown box inside the Pantanal region corresponds to the sample selected for the analysis.



**FIGURE 5.4** Sample matrix of the Pantanal region, including the sites outside the border considered in the PCN algorithm. The color scheme is the same as before: unburned land is green, water is blue, and fire is red. The region inside the white box is the  $510 \times 510$  matrix evaluated by the PCN model.

there are three possible values for a site, when running the PCN algorithm, we consider a binary alphabet in our formulas. This is due to the fact that we are studying the dependency structure of fires. Water pixels will remain water pixels regardless of their neighborhood, therefore, it does not make sense to study the conditional probability of those sites becoming fire. So, for the purpose of the PCN model, there are only two possibilities for a site: fire and not fire. Since water sites are not dependent on the context neighborhood, they are not evaluated or counted in the PCN algorithm. They only influence this process when present in the neighborhood of a “valid” site. Then, water pixels are counted as “not fire” along with unburned land pixels.

### 5.3 Results

The PCN algorithm used in the simulation study had to be modified to produce results for the real-world data analysis. For the reason specified earlier, we had to make adjustments to skip the neighborhood evaluation of water pixels inside the sample. This way, water sites and their neighborhood configurations were not counted as part of this unknown process.

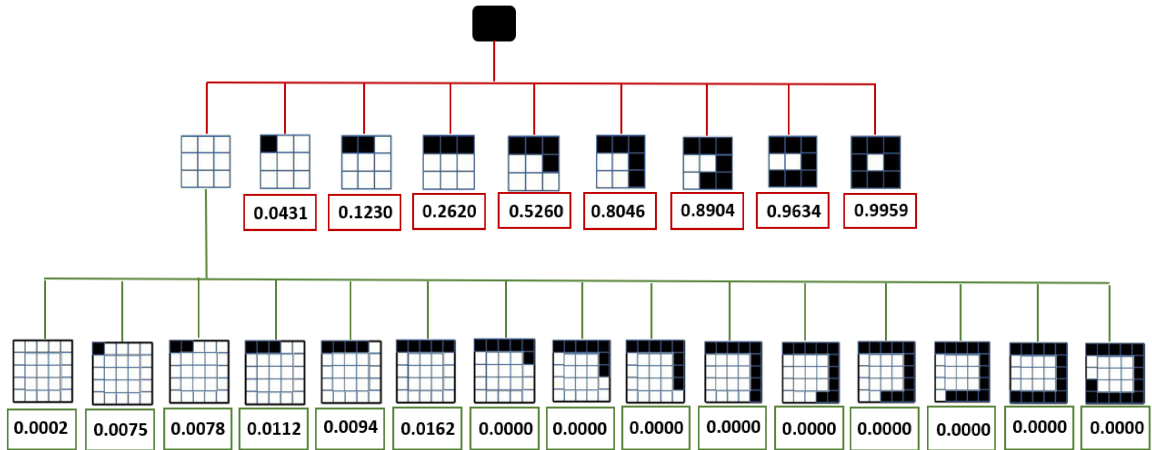
Another modification made had to do with the treatment of the border of the sample inside our algorithm. In [Chapter 4](#), the sample matrix was mirrored to allow the evaluation of sites near the border. Here, the boundaries were arbitrarily imposed by us and information regarding the region outside those boundaries was available (as shown in [Figure 5.2](#)). Therefore, we simply used the real values outside the selected sample. We discovered that removing the step responsible for mirroring the matrix made the PCN algorithm much more computationally efficient.

The most time-consuming stage of the algorithm, builds a tree from the sample under study containing all the site counts as well as their neighborhood counts. In the simulation study, building this tree for a  $200 \times 200$  matrix took approximately 32 minutes. In the real-world application study, the same step was performed in 4 minutes for a  $510 \times 510$  matrix, despite the depth of the tree growing with the sample size. It is worth noting that the other stages of the PCN algorithm, responsible for calculating  $\tilde{P}_{\mathcal{D}^j}(a(\Lambda_n))$ , the value  $V_{\mathcal{D}^j}^D(a(\Lambda_n))$  and the indicator  $\chi_{\mathcal{D}^j}^D(a(\Lambda_n))$ , as well as pruning the tree, only took a few seconds to run in both studies. The recorded times were observed on a computer with an Intel i7 processor running at 1.3 GHz and using 12GB of RAM.

#### 5.3.1 PCN $\hat{\mathcal{T}}$

The resulting PCN tree and the estimated conditional probabilities of this process are given by [Figure 5.5](#). Sites inside the sample are either fire (black) or “not fire” (white). As demonstrated by the root, this PCN tree represents the spatial dependency structure and probabilities of a site being fire conditioned on the context neighborhood.

[Figure 5.5](#) indicates that there are 23 total context neighborhoods. Every first-order neighborhood configuration is a context for this process, except for the neighborhood with 8 white



**FIGURE 5.5** PCN  $\hat{\mathcal{T}}$  recovered from the PCN algorithm applied to the Pantanal matrix. The point estimate for the conditional probability of each context (or leaf) is given underneath the neighborhood configuration. Red boxes refer to first-order contexts whereas green boxes refer to second-order contexts. This tree represents the probability of a site being fire given the neighborhood.

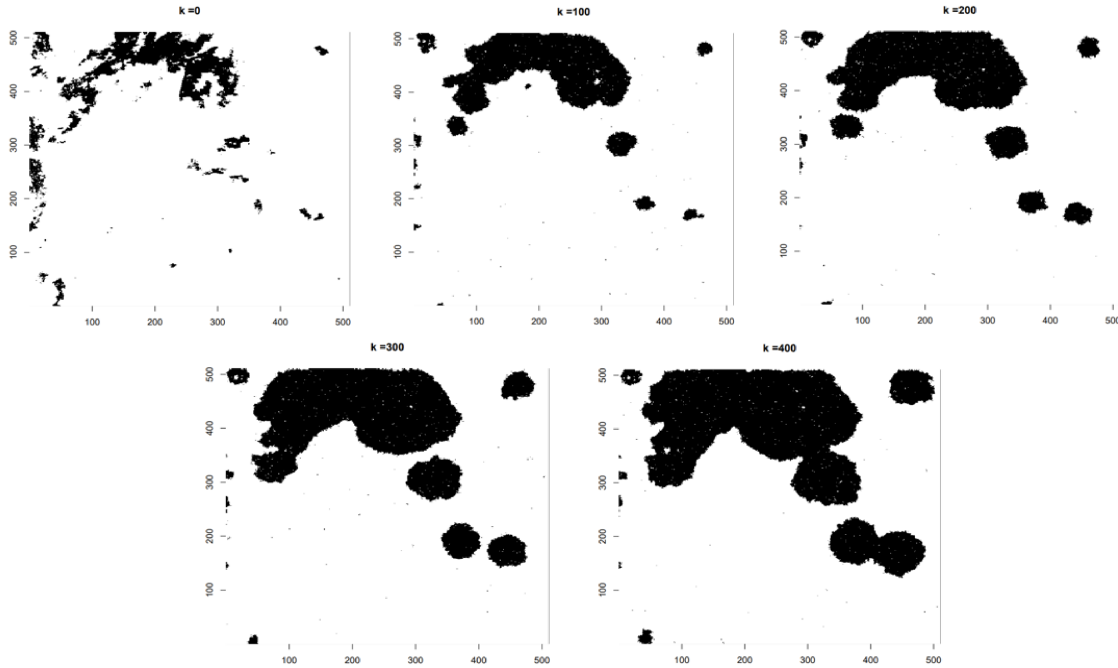
sites in the first frame. In other words, if no fires were observed in the first-order neighborhood, we need to inspect the second-order neighborhood to determine the conditional probability of the site under study. In addition, there are 15 second-order context neighborhoods out of 17 possible second-order configurations. First-order frames with 0 black sites combined with second-order frames with 15 and 16 black sites did not occur in the sample analyzed, therefore, did not appear in the estimated tree. Also, contexts with 8 to 14 black sites in the second frame appeared less than 30 times in the sample and resulted in an estimated conditional probability equal to zero.

In general, having sites of fire in the neighborhood, increases the probability of the center site being fire. Also, the conditional probability of the fires in Pantanal is mostly dependent on the immediate neighbors experiencing fires. In the cases where that does not happen, the conditional probabilities are determined based on a larger neighborhood scope, the second-order neighborhood.

### 5.3.2 Building Interval Estimates via Bootstrap

Like in the simulation study, we wanted to create interval estimates for the conditional probabilities of this unknown process. In [Chapter 4](#), however, we had the true PCN  $\mathcal{T}_0$  and the intervals were created based on it. In this chapter, we used the bootstrap method for this task, resampling from the estimated PCN  $\hat{\mathcal{T}}$  given in [Figure 5.5](#).

The resampling process was similar to the one described in [Section 4.1](#), with a few adjustments. The initial configuration was not random. Instead, we used the real matrix displayed in [Figure 5.4](#) as the starting point. Water sites did not suffer any changes throughout the iterations since they do not belong to the process we are trying to estimate. Also, the sampling algorithm could not run without a value for the conditional probabilities of the 2 “missing” second-order



**FIGURE 5.6** Iterations  $k = 0, 100, 200, 300, 400$  of the MCMC algorithm.

contexts. So, in the acceptance step, we used the empirical probability of a site being black conditioned on 8 white sites in the first frame. However, we believe any constant between 0 and 1 would suffice.

A total of 100 matrices with  $510 \times 510$  sites were created and stored after 400 iterations of this modified sampling algorithm. This task was performed in approximately 9 hours using a machine from UFMG’s Mathematics Department which has an Intel Xeon processor running at 3.8GHz and using 64GB of RAM.


























The progression of these matrices throughout the iterations is shown in [Figure 5.6](#). It seems that the limiting distribution of this process tends to have the whole matrix become fire (except for water pixels). The PCN model is simply a snapshot of the process in the short-term. Luckily, in the real-world, other factors come in place to interrupt this process.

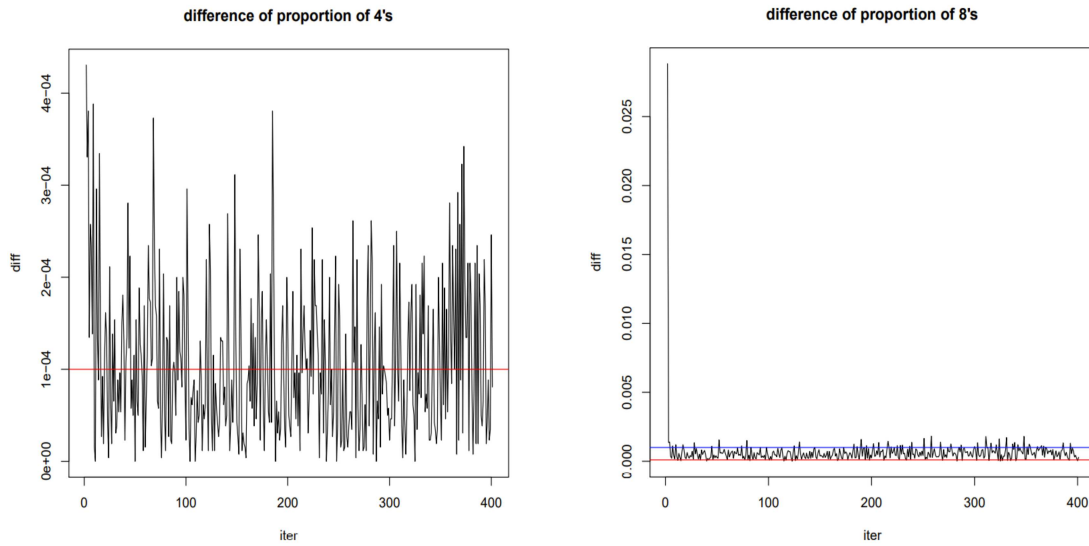
Our interest lies in recovering the PCN tree representing this phenomena, not recovering the image itself. We observed the difference in the frequencies of certain configurations from one iteration to the other to help decide when the matrices “stabilized”. [Figure 5.7](#) presents these results. The blue line represents a difference of  $10^{-3}$  while the red line is  $10^{-4}$ . We consider the matrices to have met the stabilization criterion when the difference between iterations falls underneath the blue line. Therefore, the matrices appear to settle within just a few iterations.

The estimated intervals were built based on the 2.5<sup>th</sup> percentile, median, and 97.5<sup>th</sup> percentile of the resample’s conditional probabilities. Once again, we followed Hyndman and Fan’s recommendation to use the median unbiased sample quantile estimator.

[Table 5.1](#) shows the results. In summary, the estimated intervals contain the conditional

**TABLE 5.1** Comparison between the probability of the site being fire given the context in the PCN  $\hat{\mathcal{T}}$  and the estimated interval obtained from the bootstrap method. The lower bound (LB) corresponds to the 2.5<sup>th</sup> percentile and the upper bound (UB) is the 97.5<sup>th</sup> percentile.

Context	PCN $\hat{\mathcal{T}}$	Interval Estimate		
		LB	Median	UB
	0.0431	0.0278	0.0386	0.0467
	0.1230	0.0907	0.1111	0.1307
	0.2620	0.2375	0.2586	0.2814
	0.5260	0.4793	0.5097	0.5322
	0.8046	0.7396	0.7627	0.7851
	0.8904	0.8821	0.8965	0.9109
	0.9634	0.9616	0.9665	0.9712
	0.9960	0.9952	0.9957	0.9962
	0.0002	0.0001	0.0002	0.0002
	0.0075	0.0024	0.0056	0.0103
	0.0078	0.0014	0.0075	0.0154
	0.0112	0.0019	0.0098	0.0224
	0.0094	0.0032	0.0098	0.0225
	0.0162	0.0000	0.0111	0.0356
	0.0000	0.0000	0.0108	0.0395
	0.0000	0.0000	0.0000	0.0645
	0.0000	0.0000	0.0000	0.0909
	0.0000	0.0000	0.0000	0.2129
	0.0000	0.0000	0.0000	0.1833
	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000
	-	-	0.0000	-



(a) Difference between the proportion of frames with 4 white sites in the first frame across each iteration.

(b) Difference between the proportion of frames with 8 white sites in the first frame across each iteration.

**FIGURE 5.7** Difference between the frequency of certain configurations within a matrix from one iteration to another, up to 400 iterations. The blue line represents a difference of  $10^{-3}$  and the red one is  $10^{-4}$ .

probability seen in  $\hat{\mathcal{T}}$  for all contexts, except for the one with 5 sites of fire in the first-order neighborhood. In that case, the upper bound falls short by 0.0194. All intervals have a relatively small range of values, increasing the range as the frequency of the configurations decreases within the resample (and within the matrices belonging to the resample). The intervals whose lower bound, median and upper bound all equaled zero appeared, at most, 3 times within the matrices that contained those neighborhoods. Additionally, the neighborhood containing 15 fires in the second order, appeared in one matrix a single time. This is the reason why there is no upper bound or lower bound associated with it. This specific configuration was not observed in the Pantanal original matrix.



## CHAPTER 6

# Conclusion

The probabilistic context neighborhood (PCN) model proposed by Piroutek (2013) offers a modeling alternative to studying the dependency structure of a Markov process in a two-dimensional lattice, similar to the probabilistic context tree (PCT) model in the one-dimensional case (Csiszár and Talata, 2006b). This tree representation provides an easy interpretation of the dependency of sites on their neighbors, which facilitates the understanding of data interaction behavior as demonstrated by Chapter 5.

The generalization to the multi-dimensional case was possible due to the replacement of the likelihood by the pseudo-likelihood, and the Bayesian information criterion (BIC) by the pseudo-Bayesian information criterion (PIC). In Csiszár and Talata (2006a), the consistency of the PIC estimator for the candidate neighborhood was proven, but an algorithm for the selection of the given estimator was not provided. The authors considered this task to be elusive. Since the PCN model sets a fixed frame neighborhood geometry, the cardinality of possible contexts can be calculated as given in Proposition 3.7. The main advantage of the PCN model is the proposal of an algorithm that selects the optimal PCN tree without the burden of having to calculate the PIC score for all possibilities. The consistency of the PIC estimator for the PCN tree is stated in Theorem 3.9.

Our simulation study in Chapter 4 showed the adequacy of this methodology and algorithm in practice. In all three scenarios, the algorithm correctly recovered the PCN  $\mathcal{T}_0$  that generated the sample. Although the results of Theorem 3.9 were proved for trees with depth given by  $D(n) = (\log |\Lambda_n|)^{\frac{1}{4}}$ , the simulation results demonstrated that this bound could be increased. Additionally, an empirical study suggests the suitability of the estimated transition probabilities.

Furthermore, there was a slight modification on the scale of  $n$  utilized. The outcome was not expected to be affected by this change, and this was confirmed by the results given. The BIC consistency result in the PCT case is valid when replacing  $\frac{(|E|-1)}{2}$  by any  $c > 0$ . Likewise, the consistency of the PIC estimator for candidate neighborhoods in Csiszár and Talata (2006a) is still valid when replacing  $|A|^{|T|}$  for any  $c > 0$ . In the PCN model, using  $\sqrt{n}$  is equivalent to considering  $\frac{(|A|-1)}{4}$  as the PIC penalty term in Definition 3.6, which is also greater than zero.

Possible areas worth exploring in further studies include: the generalization of the model results to lattices in  $\mathbb{Z}^d$ , an extension of this methodology to a more general graph structure

(outside of a lattice), studying the occurrence of “missing” branches in a PCN tree, and the proposal of a goodness-of-fit test for the estimated conditional probabilities of a VNRF using the pseudo-likelihood approach.

## Bibliography

Bejerano, G. and Yona, G. (2001), ‘Variations on probabilistic suffix trees: statistical modeling and prediction of protein families’, *Bioinformatics* **17**(1), 23–43.

**URL:** <https://doi.org/10.1093/bioinformatics/17.1.23>

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

**URL:** <http://www.jstor.org/stable/2987782>

Bivand, R., Keitt, T. and Rowlingson, B. (2020), *rgdal: Bindings for the ‘Geospatial’ Data Abstraction Library*. R package version 1.5-18.

Busch, J. R., Ferrari, P. A., Flesia, A. G., Fraiman, R., Grynberg, S. P. and Leonardi, F. (2009), ‘Testing statistical hypothesis on random trees and applications to the protein classification problem’, *The Annals of Applied Statistics* **3**(2), 542 – 563.

**URL:** <https://doi.org/10.1214/08-AOAS218>

Bühlmann, P. (2000), ‘Model selection for variable length markov chains and tuning the context algorithm’, *Annals of the Institute of Statistical Mathematics* **52**, 287–315.

**URL:** <https://doi.org/10.1023/A:1004165822461>

Bühlmann, P. and Wyner, A. J. (1999), ‘Variable length markov chains’, *The Annals of Statistics* **27**(2), 480–513.

**URL:** <http://www.jstor.org/stable/120101>

Csiszár, I. and Shields, P. C. (2000), ‘The consistency of the BIC Markov order estimator’, *The Annals of Statistics* **28**(6), 1601 – 1619.

**URL:** <https://doi.org/10.1214/aos/1015957472>

Csiszár, I. and Talata, Z. (2006a), ‘Consistent estimation of the basic neighborhood of markov random fields’, *The Annals of Statistics* **34**(1), 123–145.

**URL:** <http://www.jstor.org/stable/25463410>

- Csiszár, I. and Talata, Z. (2006b), ‘Context tree estimation for not necessarily finite memory processes, via bic and mdl’, *IEEE Transactions on Information Theory* **52**(3), 1007–1016.
- Duarte, D., Galves, A. and Garcia, N. (2006), ‘Markov approximation and consistent estimation of unbounded probabilistic suffix trees’, *Bulletin of the Brazilian Mathematical Society* **37**, 581–592.  
**URL:** <https://doi.org/10.1007/s00574-006-0029-7>
- Fahrmeir, L. and Lang, S. (2001), ‘Bayesian inference for generalized additive mixed models based on markov random field priors’, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **50**(2), 201–220.  
**URL:** <http://www.jstor.org/stable/2680887>
- Frank, O. and Strauss, D. (1986), ‘Markov graphs’, *Journal of the American Statistical Association* **81**(395), 832–842.  
**URL:** <http://www.jstor.org/stable/2289017>
- Galves, A., Galves, C., García, J. E., Garcia, N. L. and Leonardi, F. (2012), ‘Context tree selection and linguistic rhythm retrieval from written texts’, *The Annals of Applied Statistics* **6**(1).  
**URL:** <https://doi.org/10.1214/11-AOAS511>
- Garivier, A. and Leonardi, F. (2011), ‘Context tree selection: A unifying view’, *Stochastic Processes and their Applications* **121**(11), 2488–2506.  
**URL:** <https://www.sciencedirect.com/science/article/pii/S0304414911001591>
- Geman, S. and Geman, D. (1984), ‘Stochastic relaxation, gibbs distributions, and the bayesian restoration of images’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6**(6), 721–741.
- Giglio, L., Boschetti, L., Roy, D., Hoffmann, A. A., Humber, M. and Hall, J. V. (2020), *Collection 6 MODIS Burned Area Product User’s Guide Version 1.3*, NASA.  
**URL:** [https://lpdaac.usgs.gov/documents/875/MCD64\\_User\\_Guide\\_V6.pdf](https://lpdaac.usgs.gov/documents/875/MCD64_User_Guide_V6.pdf)
- Hastings, W. K. (1970), ‘Monte carlo sampling methods using markov chains and their applications’, *Biometrika* **57**(1), 97–109.  
**URL:** <http://www.jstor.org/stable/2334940>
- Hernández-Lemus, E. (2021), ‘Random fields in physics, biology and data science’, *Frontiers in Physics* **9**, 77.  
**URL:** <https://www.frontiersin.org/article/10.3389/fphy.2021.641859>
- Hyndman, R. and Fan, Y. (1996), ‘Sample quantiles in statistical packages’, *The American Statistician* **50**, 361–365.

- IBGE (n.d.), 'Brasil - unidades da federação 2018'. [Accessed: May 2021].  
**URL:** <https://portaldemapas.ibge.gov.br/portal.php#mapa222184>
- INPE (2021a), 'Limite do bioma pantanal - shapefile'. [Accessed: February 2021].  
**URL:** <http://terrabrasilis.dpi.inpe.br/downloads/>
- INPE (2021b), 'Monitoramento de focos ativos por bioma'. [Accessed: January 2021].  
**URL:** [https://queimadas.dgi.inpe.br/queimadas/portal-static/estatisticas\\_estados/](https://queimadas.dgi.inpe.br/queimadas/portal-static/estatisticas_estados/)
- Ji, C. and Seymour, L. (1996), 'A consistent model selection procedure for Markov random fields based on penalized pseudolikelihood', *The Annals of Applied Probability* **6**(2), 423 – 443.  
**URL:** <https://doi.org/10.1214/aoap/1034968138>
- Kim, I. Y. and Yang, H. S. (1995), 'An integrated approach for scene understanding based on markov random field model', *Pattern Recognition* **28**(12), 1887–1897.  
**URL:** <https://www.sciencedirect.com/science/article/pii/0031320395000615>
- Kindermann, R. and Snell, L. (1980), *Markov random fields and their applications*, Vol. 1, American Mathematical Society.
- Lin, Z., Sanders, S. J., Li, M., Sestan, N., State, M. W. and Zhao, H. (2015), 'A markov random field-based approach to characterizing human brain development using spatial–temporal transcriptome data', *The Annals of Applied Statistics* **9**(1), 429–451.  
**URL:** <http://www.jstor.org/stable/24522425>
- Löcherbach, E. and Orlandi, E. (2011), 'Neighborhood radius estimation for variable-neighborhood random fields', *Stochastic Processes and their Applications* **121**(9), 2151–2185.  
**URL:** <https://www.sciencedirect.com/science/article/pii/S0304414911001025>
- Martin, A., Seroussi, G. and Weinberger, M. J. (2004), 'Linear time universal coding and time reversal of tree sources via fsm closure', *IEEE Transactions on Information Theory* **50**(7), 1442–1468.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953), 'Equation of state calculations by fast computing machines', *The journal of chemical physics* **21**(6), 1087–1092.
- Onural, L., Pınar, M. Ç. and Fırtına, C. (2021), 'Modeling economic activities and random catastrophic failures of financial networks via gibbs random fields', *Computational Economics* **58**, 203–232.  
**URL:** <https://doi.org/10.1007/s10614-020-10023-3>

- Pebesma, E. (2020), *stars: Spatiotemporal Arrays, Raster and Vector Data Cubes*. R package version 0.4-3.
- Peng, F., Lu, J., Wang, Y., Xu, R. Y.-D., Ma, C. and Yang, J. (2016), ‘N-dimensional markov random field prior for cold-start recommendation’, *Neurocomputing* **191**, 187–199.  
**URL:** <https://www.sciencedirect.com/science/article/pii/S0925231216000825>
- Pensar, J., Nyman, H., Niiranen, J. and Corander, J. (2017), ‘Marginal Pseudo-Likelihood Learning of Discrete Markov Network Structures’, *Bayesian Analysis* **12**(4), 1195 – 1215.  
**URL:** <https://doi.org/10.1214/16-BA1032>
- Piroutek, A., Duarte, D. and Alves, C. (n.d.), Probabilistic context neighborhood model for lattices in  $\mathbb{Z}^2$ . Unpublished preprint.
- Piroutek, A. M. (2013), Novos modelos de vizinhança espacial e vigilância prospectiva espaço-tempo, PhD thesis, Universidade Federal de Minas Gerais.  
**URL:** [http://www.est.ufmg.br/portal/arquivos/doutorado/teses/tese\\_aline\\_piroutek.pdf](http://www.est.ufmg.br/portal/arquivos/doutorado/teses/tese_aline_piroutek.pdf)
- R Core Team (2020), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
- Rissanen, J. (1983), ‘A universal data compression system’, *IEEE Transactions on Information Theory* **29**(5), 656–664.
- Rue, H. and Held, L. (2005), *Gaussian Markov Random Fields: Theory And Applications (Monographs on Statistics and Applied Probability)*, Chapman & Hall/CRC.
- Schwarz, G. (1978), ‘Estimating the dimension of a model’, *The Annals of Statistics* **6**(2), 461–464.  
**URL:** <http://www.jstor.org/stable/2958889>
- Subudhi, B. N., Bovolo, F., Ghosh, A. and Bruzzone, L. (2014), ‘Spatio-contextual fuzzy clustering with markov random field model for change detection in remotely sensed images’, *Optics & Laser Technology* **57**, 284–292. Optical Image Processing.  
**URL:** <https://www.sciencedirect.com/science/article/pii/S003039921300368X>
- Talata, Z. (2005), ‘Model selection via information criteria’, *Periodica Mathematica Hungarica* **51**, 99–117.  
**URL:** <https://doi.org/10.1007/s10998-005-0023-7>
- Wei, Z. and Li, H. (2007), ‘A Markov random field model for network-based analysis of genomic data’, *Bioinformatics* **23**(12), 1537–1544.  
**URL:** <https://doi.org/10.1093/bioinformatics/btm129>

West, R., Paskov, H. S., Leskovec, J. and Potts, C. (2014), ‘Exploiting social network structure for person-to-person sentiment analysis’, *CoRR* **abs/1409.2450**.

**URL:** <http://arxiv.org/abs/1409.2450>

Willems, F. M., Shtarkov, Y. M. and Tjalkens, T. J. (1995), ‘The context-tree weighting method: basic properties’, *IEEE Transactions on Information Theory* **41**(3), 653–664.

Wu, Z., Lin, D. and Tang, X. (2016), ‘Deep markov random field for image modeling’, *CoRR* **abs/1609.02036**.

**URL:** <http://arxiv.org/abs/1609.02036>

Zhang, X., Xiao, P. and Feng, X. (2017), ‘Toward combining thematic information with hierarchical multiscale segmentations using tree markov random field model’, *ISPRS Journal of Photogrammetry and Remote Sensing* **131**, 134–146.

**URL:** <https://www.sciencedirect.com/science/article/pii/S0924271617302897>

# APPENDIX A

## Proof of Proposition 3.9

First, let us consider the case where frames are considered identical if the same elements in the alphabet appear the same number of times, regardless of their position. Since the  $k$ -th frame has  $8k$  sites and each site can assume  $|A|$  possible values. In this scenario, the number of possible configurations of a frame of order  $k$  is obtained by solving the number of possible ways  $8k$  sites can be distributed among  $|A|$  distinct groups. That is simply a problem of combination with repetition. So, the  $k$ -th frame has  $C(8k + |A| - 1, |A| - 1)$  possible configurations. Consequently, the number of leaves of a full PCN  $\mathcal{T}$  of depth  $d(\mathcal{T})$  is given by:

$$|\mathcal{T}| = \prod_{k=1}^{d(\mathcal{T})} \binom{8k + |A| - 1}{|A| - 1}.$$

The second scenario considers the position of sites within a frame to be important. The proof, in this case, follows from the definition of frames. [Definition 3.1](#) states that the  $k$ -th frame is obtained by taking a square of side  $2k + 1$  and removing a smaller square of side  $2k - 1$ , both centered on site  $i$ . Therefore, a neighborhood  $\mathcal{D}^k$ , which is the concatenation of frames of order 1 through  $k$ , is given by the number of sites within a square of side  $2k + 1$  minus the center site:

$$|\mathcal{D}^k| = (2k + 1)(2k + 1) - 1 = 4k^2 + 4k = \frac{8k(k + 1)}{2}.$$

Therefore, the number of leaves of a PCN  $\mathcal{T}$  of depth  $d(\mathcal{T}) = k$  is the number of possible arrangements of  $|\mathcal{D}^k|$  sites, where each site can assume  $|A|$  possible values.

$$|\mathcal{T}| = |A|^{|\mathcal{D}^k|} = |A|^{\frac{8k(k+1)}{2}}.$$

■



# APPENDIX B

## Simulation 3 Results

**TABLE B:** Comparison between the true probability of a site being black given the context neighborhood and the estimated conditional probabilities of Simulation 3. Both the initial single matrix estimation results, as well as the interval estimates obtained later from a sample of matrices, are included in the table. The lower bound (LB) corresponds to the 2.5<sup>th</sup> percentile and the upper bound (UB) is the 97.5<sup>th</sup> percentile.

Number of black sites in $\partial^1$	Number of black sites in $\partial^2$	True	Single matrix estimate	Interval Estimate		
				LB	Median	UB
0	0	0.0832	0.0752	0.0256	0.0605	0.1156
0	1	0.0998	0.0916	0.0651	0.0915	0.1290
0	2	0.1192	0.1107	0.0814	0.1114	0.1334
0	3	0.1419	0.1225	0.0923	0.1303	0.1839
0	4	0.1680	0.1784	0.1364	0.1639	0.2189
0	5	0.1978	0.2096	0.1550	0.1939	0.2616
0	6	0.2315	0.2135	0.1638	0.2264	0.3057
0	7	0.2689	0.3564	0.1571	0.2704	0.3948
0	8	0.3100	0.2973	0.1564	0.3027	0.4011
0	9	0.3543	0.3200	0.1956	0.3880	0.5794
0	10	0.4013	0.1000	0.0493	0.4365	0.9014
0	11	0.4502	0.5000	0.0000	0.5000	1.0000
0	12	0.5000	0.5000	0.0000	0.5000	1.0000
0	13	0.5498	1.0000	0.0000	0.5000	1.0000
0	14	0.5987	-	0.0000	1.0000	1.0000

Number of black sites in $\partial^1$	Number of black sites in $\partial^2$	True	Single matrix estimate	Interval Estimate		
				LB	Median	UB
0	15	0.6457	-	-	-	-
0	16	0.6900	-	-	-	-
1	0	0.0998	0.0797	0.0601	0.1056	0.1812
1	1	0.1192	0.1183	0.0913	0.1227	0.1579
1	2	0.1419	0.1442	0.1125	0.1449	0.1753
1	3	0.1680	0.1872	0.1384	0.1705	0.1957
1	4	0.1978	0.2143	0.1661	0.2015	0.2437
1	5	0.2315	0.2600	0.1919	0.2295	0.2668
1	6	0.2689	0.2802	0.2338	0.2656	0.3137
1	7	0.3100	0.3521	0.2662	0.3087	0.3661
1	8	0.3543	0.3681	0.2811	0.3569	0.4198
1	9	0.4013	0.3737	0.2662	0.3879	0.4990
1	10	0.4502	0.4576	0.2815	0.4565	0.5901
1	11	0.5000	0.4500	0.2774	0.4737	0.6675
1	12	0.5498	0.6000	0.2796	0.5833	0.8750
1	13	0.5987	0.7500	0.0000	0.6000	1.0000
1	14	0.6457	-	0.0000	0.6667	1.0000
1	15	0.6900	-	0.0000	1.0000	1.0000
1	16	0.7311	-	0.0000	0.0000	1.0000
2	0	0.1192	0.1758	0.0456	0.1231	0.2078
2	1	0.1419	0.1047	0.0924	0.1388	0.2012
2	2	0.1680	0.1701	0.1368	0.1739	0.2093
2	3	0.1978	0.1869	0.1699	0.1988	0.2314
2	4	0.2315	0.2200	0.1994	0.2308	0.2694
2	5	0.2689	0.2394	0.2348	0.2674	0.3027
2	6	0.3100	0.3156	0.2756	0.3168	0.3517
2	7	0.3543	0.3327	0.2992	0.3515	0.4057
2	8	0.4013	0.3853	0.3502	0.4034	0.4567
2	9	0.4502	0.4120	0.3771	0.4510	0.4978

Number of black sites in $\partial^1$	Number of black sites in $\partial^2$	True	Single matrix estimate	Interval Estimate		
				LB	Median	UB
2	10	0.5000	0.5115	0.4244	0.4981	0.6030
2	11	0.5498	0.4444	0.4518	0.5548	0.6638
2	12	0.5987	0.5610	0.4554	0.5871	0.7523
2	13	0.6457	0.5714	0.4070	0.6667	0.8414
2	14	0.6900	0.8000	0.3388	0.7071	1.0000
2	15	0.7311	0.5000	0.0000	0.7500	1.0000
2	16	0.7685	-	0.0000	1.0000	1.0000
3	0	0.1419	0.2500	0.0000	0.1501	0.3057
3	1	0.1680	0.1250	0.0700	0.1712	0.2481
3	2	0.1978	0.2017	0.1436	0.2037	0.2718
3	3	0.2315	0.2056	0.1992	0.2284	0.2813
3	4	0.2689	0.2437	0.2275	0.2595	0.3022
3	5	0.3100	0.3019	0.2702	0.3089	0.3470
3	6	0.3543	0.3617	0.3186	0.3561	0.3949
3	7	0.4013	0.3699	0.3577	0.4004	0.4334
3	8	0.4502	0.4428	0.3914	0.4528	0.4884
3	9	0.5000	0.5330	0.4419	0.5024	0.5404
3	10	0.5498	0.5020	0.4865	0.5476	0.6039
3	11	0.5987	0.6480	0.5192	0.5955	0.6510
3	12	0.6457	0.5636	0.5506	0.6420	0.7605
3	13	0.6900	0.6957	0.5772	0.6886	0.8272
3	14	0.7311	0.6970	0.5414	0.7276	0.8977
3	15	0.7685	0.4444	0.6176	0.8000	0.9652
3	16	0.8022	1.0000	0.2458	1.0000	1.0000
4	0	0.1680	0.0667	0.0000	0.1667	0.5292
4	1	0.1978	0.1395	0.0355	0.2162	0.3655
4	2	0.2315	0.2124	0.1597	0.2353	0.3272
4	3	0.2689	0.2889	0.1984	0.2739	0.3499
4	4	0.3100	0.3019	0.2558	0.3057	0.3698

Number of black sites in $\partial^1$	Number of black sites in $\partial^2$	True	Single matrix estimate	Interval Estimate		
				LB	Median	UB
4	5	0.3543	0.3237	0.3014	0.3570	0.4189
4	6	0.4013	0.3854	0.3559	0.3991	0.4657
4	7	0.4502	0.4481	0.4047	0.4470	0.5016
4	8	0.5000	0.5108	0.4539	0.5039	0.5364
4	9	0.5498	0.5709	0.5049	0.5511	0.5872
4	10	0.5987	0.5996	0.5559	0.5994	0.6364
4	11	0.6457	0.6732	0.5968	0.6411	0.7062
4	12	0.6900	0.6979	0.6324	0.6885	0.7354
4	13	0.7311	0.7110	0.6603	0.7353	0.7906
4	14	0.7685	0.7468	0.6884	0.7617	0.8511
4	15	0.8022	0.7097	0.5847	0.8157	0.9099
4	16	0.8320	0.8750	0.5329	0.8571	1.0000
5	0	0.1978	0.0000	0.0000	0.0000	0.8528
5	1	0.2315	0.1429	0.0000	0.2183	0.5408
5	2	0.2689	0.2000	0.1620	0.2941	0.4623
5	3	0.3100	0.2927	0.1928	0.3131	0.4464
5	4	0.3543	0.2992	0.2731	0.3730	0.4503
5	5	0.4013	0.4311	0.3346	0.4016	0.4758
5	6	0.4502	0.4649	0.3683	0.4482	0.5085
5	7	0.5000	0.5013	0.4350	0.5053	0.5531
5	8	0.5498	0.5375	0.4983	0.5506	0.6006
5	9	0.5987	0.6057	0.5599	0.5899	0.6455
5	10	0.6457	0.6277	0.6068	0.6446	0.6837
5	11	0.6900	0.6604	0.6381	0.6857	0.7427
5	12	0.7311	0.7285	0.6930	0.7292	0.7592
5	13	0.7685	0.7959	0.7347	0.7718	0.8047
5	14	0.8022	0.7890	0.7519	0.7912	0.8399
5	15	0.8320	0.8583	0.7499	0.8364	0.8794
5	16	0.8581	0.7778	0.6772	0.8404	0.9709

Number of black sites in $\partial^1$	Number of black sites in $\partial^2$	True	Single matrix estimate	Interval Estimate		
				LB	Median	UB
6	0	0.2315	-	0.0000	0.0000	1.0000
6	1	0.2689	0.0000	0.0000	0.1250	1.0000
6	2	0.3100	0.3750	0.0000	0.3333	0.9014
6	3	0.3543	0.5385	0.1659	0.4000	0.6803
6	4	0.4013	0.4151	0.2366	0.4000	0.5678
6	5	0.4502	0.4458	0.2939	0.4757	0.5463
6	6	0.5000	0.5935	0.4243	0.4964	0.5667
6	7	0.5498	0.5815	0.4725	0.5508	0.6424
6	8	0.5987	0.5980	0.5443	0.6083	0.6735
6	9	0.6457	0.6557	0.6078	0.6452	0.6767
6	10	0.6900	0.7263	0.6478	0.6886	0.7162
6	11	0.7311	0.7391	0.6929	0.7341	0.7600
6	12	0.7685	0.7651	0.7468	0.7729	0.7994
6	13	0.8022	0.7769	0.7719	0.8034	0.8288
6	14	0.8320	0.8009	0.7914	0.8362	0.8629
6	15	0.8581	0.8701	0.8079	0.8611	0.8989
6	16	0.8808	0.9367	0.8317	0.8795	0.9466
7	0	0.2689	-	0.0000	0.5000	1.0000
7	1	0.3100	-	0.0000	0.0000	1.0000
7	2	0.3543	0.5000	0.0000	0.1000	1.0000
7	3	0.4013	0.6000	0.0000	0.5000	1.0000
7	4	0.4502	0.4545	0.1569	0.4444	0.9408
7	5	0.5000	0.3600	0.3326	0.5314	0.7113
7	6	0.5498	0.6429	0.4632	0.5687	0.7000
7	7	0.5987	0.6143	0.4727	0.6165	0.6969
7	8	0.6457	0.6768	0.5846	0.6464	0.7315
7	9	0.6900	0.7378	0.6430	0.6909	0.7520
7	10	0.7311	0.7740	0.6749	0.7292	0.7781
7	11	0.7685	0.7726	0.7394	0.7657	0.8000

Number of black sites in $\partial^1$	Number of black sites in $\partial^2$	True	Single matrix estimate	Interval Estimate		
				LB	Median	UB
7	12	0.8022	0.7934	0.7806	0.8036	0.8276
7	13	0.8320	0.8106	0.8145	0.8339	0.8564
7	14	0.8581	0.8727	0.8366	0.8598	0.8811
7	15	0.8808	0.8762	0.8497	0.8817	0.9136
7	16	0.9002	0.9384	0.8334	0.8961	0.9378
8	0	0.3100	-	-	1.0000	-
8	1	0.3543	-	-	0.0000	-
8	2	0.4013	0.0000	0.0000	1.0000	1.0000
8	3	0.4502	0.0000	0.0000	0.5000	1.0000
8	4	0.5000	0.5000	0.0000	0.5000	1.0000
8	5	0.5498	0.7500	0.0000	0.6667	1.0000
8	6	0.5987	0.8462	0.2063	0.6307	0.9408
8	7	0.6457	0.5909	0.3912	0.6364	0.8431
8	8	0.6900	0.5833	0.5607	0.6865	0.8167
8	9	0.7311	0.7391	0.6418	0.7353	0.8395
8	10	0.7685	0.7961	0.6995	0.7648	0.8229
8	11	0.8022	0.8370	0.7569	0.7984	0.8584
8	12	0.8320	0.8416	0.8042	0.8345	0.8631
8	13	0.8581	0.8540	0.8248	0.8662	0.8851
8	14	0.8808	0.8911	0.8619	0.8849	0.9105
8	15	0.9002	0.9307	0.8561	0.9050	0.9254
8	16	0.9168	0.8977	0.8769	0.9283	0.9653