SPATIAL PRODUCT PARTITION MODEL THROUGH SPANNING TREES

LEONARDO VILELA TEIXEIRA

SPATIAL PRODUCT PARTITION MODEL THROUGH SPANNING TREES

Dissertation presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Renato Martins Assunção
Co-Advisor: Rosangela Helena Loschi

Belo Horizonte
June 2015

LEONARDO VILELA TEIXEIRA

MODELO PARTIÇÃO PRODUTO ESPACIAL USANDO ÁRVORES GERADORAS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Renato Martins Assunção
Coorientador: Rosangela Helena Loschi

Belo Horizonte
Junho de 2015

Teixeira, Leonardo Vilela.

T266s Spatial Product Partition Model through Spanning Trees / Leonardo Vilela Teixeira. — Belo Horizonte, 2015.

xxvi, 72 f. : il. ; 29cm

Dissertação (Mestrado) — Universidade Federal de Minas Gerais – Departamento de Ciência da Computação.

Orientador: Renato Martins Assunção:

1. Computação - Teses. 2. Classificação espacial - Teses. 3. Teoria bayesiana de decisão estatística – Teses. 4. Variáveis latentes – Teses. I Orientador II. Título.

519.6*63(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Spatial product partition model through spanning trees

LEONARDO VILELA TEIXEIRA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. Renato Martins Assunção - Orientador
Departamento de Ciência da Computação - UFMG

Prof. Fabio Gagliardi Cozman
Departamento de Engenharia Mecatrônica - USP

Prof. Pedro Olmo Stancioli Vaz de Melo
Departamento de Ciência da Computação - UFMG

RHLoschi
Profa. Rosangela Helena Loschi
Departamento de Estatística - UFMG

Belo Horizonte, 03 de junho de 2015.

Acknowledgments

I could not have completed this work without the help and support of many people throughout these years.

I want to thank my family for their continuous support and encouragement. My parents are a constant source of inspiration, always showing me how to be a better person. To my beloved sister, even though the distance has made our physical reunion less frequent, thank you for being always there for me. To my little niece, your arrival has been a constant source of joy in our lives.

To my advisor, Prof. Renato Assunção, thank you for being a real mentor and for your guidance in the past couple of years. You've done much for me and I constantly learn from you. I also thank my co-advisor, Prof. Rosangela Loschi, who has become a joyful presence on our weekly meeting, and whose contributions were essential to keep this work on the right tracks. To all the professors who contributed to my professional growth, thank you for your valuable lessons. To the secretaries who work at the DCC, thank you for being always so helpful.

I also thank all my friends, who constantly make my life happier. To my university colleagues, who shared the joy and challenges of these past years, thank you. To my gaming friends who transformed many afternoons and nights into delightful and entertaining moments of racing camels, eating bamboo and building settlements, I say this: you are the best! Lastly, to the group of friends, who are the brothers with whom I don't share blood, I thank each of you twenty six times!

Finally, I wish to thank you, who is reading this work. Thank you for devoting your time to check this, which has been an important part of my life. I hope you find it useful.

"Das Auge sieht weit, der Verstand noch weiter"
(German proverb)

Resumo

Ao analisar dados espaciais, muitas vezes há a necessidade de agregar áreas geográficas em regiões maiores, um processo chamado de regionalização ou agrupamento com restrições espaciais. Este tipo de agregação pode ser útil para tornar a análise de dados tratável, reduzir o efeito de diferentes populações levando a uma melhor manipulação estatística dos dados ou até mesmo para facilitar a visualização.

Neste trabalho, apresentamos um novo método de regionalização que incorpora o conceito de árvores geradoras a um modelo estatístico, formando um novo tipo de modelo partição produto espacial. Ao condicionar as partições em quebras de árvores geradoras, reduz-se o espaço de busca, possibilitando a construção de um algoritmo eficaz para amostragem da distribuição a posteriori das partições.

Nós mostramos que, ao usar um modelo estatístico Bayesiano, é possível acomodar melhor a variação natural dos dados e diminuir o efeito de valores extremos, produzindo assim melhores resultados quando comparado com as abordagens tradicionais. Nós também mostramos como nosso modelo é flexível o suficiente para acomodar dados com diferentes distribuições. Finalmente, nós avaliamos o nosso método através de experimentos com dados simulados, bem como através de dois estudos de caso.

Palavras-chave: classificação espacial; modelos de partição; modelos latentes; aprendizagem bayesiana de estrutura.

Abstract

When performing analysis of spatial data, there is often the need to aggregate geographical areas into larger regions, a process called regionalization or spatially constrained clustering. This type of aggregation can be useful to make data analysis tractable, reduce the effect of different populations for a better statistical handling of the data or even to facilitate the visualization.

In this work we present a new regionalization method which incorporates the concept of spanning trees into a statistical framework, forming a new type of spatial product partition model. By conditioning the partitions to splits of spanning trees we reduce the search space and enable the construction of an effective sampling algorithm.

We show how using a Bayesian statistical framework we are able to better accommodate the natural variation of the data and to diminish the effect of outliers, producing better results when compared with the traditional approaches. We also show how our model is flexible enough to accommodate distinct distributions of data. Finally, we evaluate our method through experiments with simulated data as well as with two distinct case studies.

Palavras-chave: spatial clustering; partition models; latent models; Bayesian structure learning;.

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

HDI	Human Development Index
IBGE	Instituto Brasileiro de Geografia e Estatística
IPEA	Instituto de Pesquisa Econômica Aplicada
MCMC	Markov Chain Monte Carlo
MRF	Markov Random Field
PPM	Product Partition Model
SPPM	Spatial Product Partition Model
UNDP	United Nations Development Programme

Contents

Chapter 1

Introduction

1.1 Motivation

We collect, represent, and store data in an effort to better understand our world and to help us make decisions. Once we have the data, the study and analysis of it becomes a central and important activity.

In many situations there are spatial characteristics naturally associated with the data. Human beings, for instance, are neither uniformly nor arbitrarily distributed in space. Frequently, geographical characteristic such as topology, weather, and others may affect the behavior and the spatial distribution of individuals. The recent advances in science and technology have made the acquisition and usage of spatial data not only something easier but also a more present task. Therefore, the spatial analysis becomes a particularly important exploratory and analytical tool.

When performing spatial analysis, it is often needed to aggregate geographical areas into larger regions. This can serve a range of purposes, such as reducing the noise introduced by outliers and inaccurate data, make data analysis tractable, providing a better statistical handling of the data (by reducing the effect of different populations), and even simply to facilitate the visualization and understanding of the information [Wise et al., 1997].

There are two possible ways to carry out this aggregation. One way is through an artificial aggregation, where the constructed regions are defined rather arbitrarily or using official or normative designations (such as states, districts and counties). This kind of aggregation, however, is usually the expression of political will and may not take into account the geographical characteristics or information specific to the domain being studied. Another way is to perform the aggregation based on the analysis of characteristics of data which are related to the phenomena being studied.

Through the use of official or nominative regions, in many cases, the result of statistical analysis can be affected by problems of the aggregation, such as the ecological fallacy [Robinson, 1950; Openshaw, 1984a], the modifiable areal unit problem [Openshaw, 1984b], among others. Therefore, the use of analytical regions is of particular relevance since this kind of analysis, if properly executed, may be able to produce more helpful results in the discovery of spatial patterns than the original data [Alvanides and Openshaw, 1999].

In spatial analysis, this type of aggregation receives the name of *regionalization* or *spatially constrained clustering*. It is the process of aggregating small contiguous geographical areas forming larger regions (called spatial clusters), with the purpose of partitioning the space into spatial clusters in a form such that areas which are similar according to some characteristics belong to the same spatial cluster. In Fig. 1.1 we show an example of an outcome of a regionalization method. The 3127 continental US counties have been aggregated into seven regions or clusters according to their percentage of votes for George Bush in the US Presidential Election of 2004. Neighboring counties with similar voting percentages are clustered together.

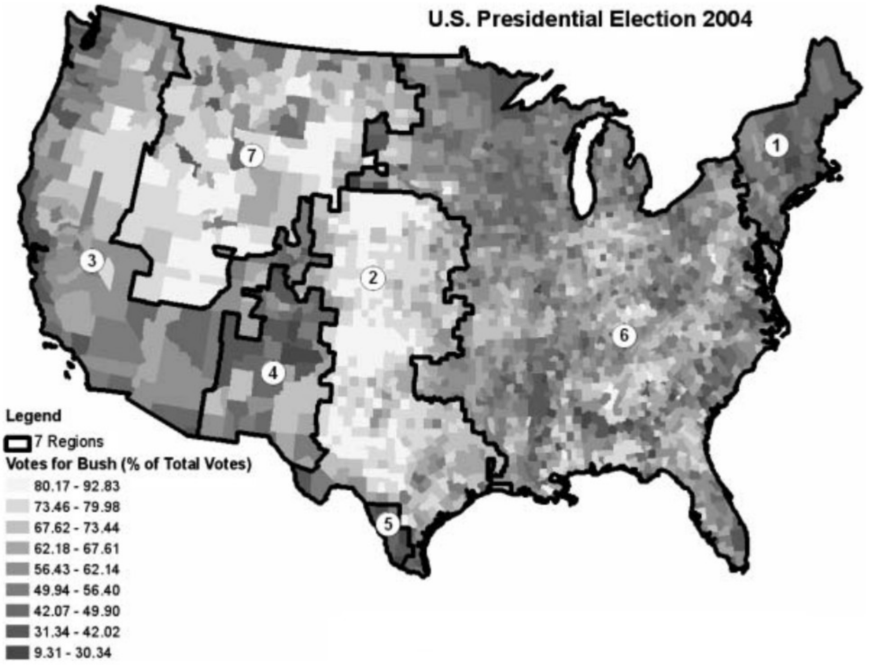


Figure 1.1: Example of regionalization as seen in Guo [2008]

Many applications benefit from the usage of regionalization. For instance, a specific regionalization of resources for healthcare delivery has been implemented by the Department of Veterans Affairs (VA, formerly the Veterans Administration) in the United States. Rather than using established administrative boundaries, the

VA regional system determined service areas boundaries taking into account the expected patterns of health service use [Ricketts, 1997]. Other examples of applications are: sampling procedures [Martin, 1998], establishment of communication protocols in geo-sensor networks [Reis et al., 2007], classification of areas with high incidence of diseases [Martins-Bedé et al., 2009], and environmental planning [Bernetti et al., 2011].

Most of the regionalization techniques consider the data as fixed, static values. Frequently, this assumption is inappropriate, as it doesn't allow for measurement error or uncertainty on the areas' measures and it doesn't allow also for the evaluation of the uncertainty of the obtained clusters. It is valid to think of neighbouring areas as having similar values due to an underlying process which has a regional effect. Or to consider the collected data as only a random manifestation of some characteristic which may vary in time.

Consider, as an example, the measured infant mortality rate of a small town in a given year. This value should not be considered as the most representative value for the true mortality rate. This value can be severely impacted by small differences from one year to another, particularly if the area has a small population. The measured value can show a natural variability, expressed in this widely different infant mortality rate variation in two successive years in small population areas, which is not considered in the traditional regionalization techniques. Thus, using a explicitly stochastic framework to perform the regionalization can provide better and more helpful results. In Fig. 1.2 a regionalization produced by a non-stochastic method is shown. This is a map of municipalities in the South of Brazil partitioned according to their value of the bladder cancer mortality rate. The regionalization technique used is named Automated Zoning Procedure.

In this work we present a regionalization method which explores both a statistical model for the data as well as a representation of the spatial relationship of the data through a graph. We adopt a hierarchical Bayesian model to represent the data. Any type of distribution can be adopted for the observed data, continuous or discrete. The clustering structure is induced by a product partition model, a statistical framework for clustered data, together with spanning trees of the spatial graph built from the data to provide an effective algorithm to perform spatial inference and regionalization.

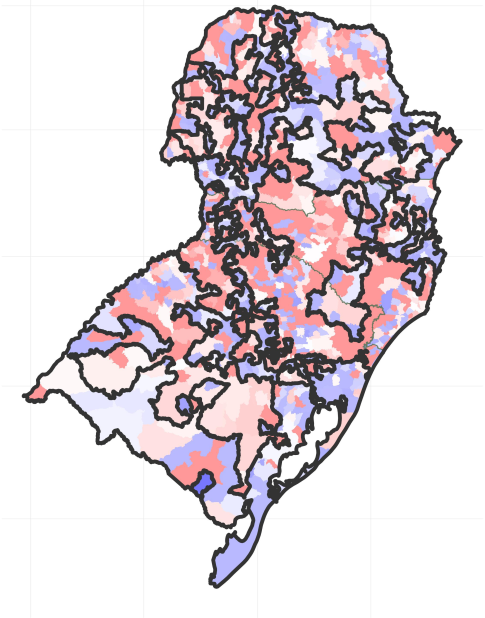


Figure 1.2: Example of a bad regionalization

1.2 Contributions

In this work we incorporate the concept of spanning trees into a hierarchical Bayesian model, introducing a new spatial product partition model. Our model allows for any type of distribution for the observed data.

By conditioning the partitions to splits of spanning trees, we substantially reduce the space of partitions or clusters we have to explore. The clustering structure assumes a product form, which conveniently allows for some sort of conjugacy where the posterior distribution can be derived from the prior distribution of each cluster. Based on this framework, we propose an efficient Gibbs sampler algorithm to sample from the posterior distributions, specially that of the partition.

Through our new model, we provide a new framework to cluster data which is flexible and can be used to model data with different structures, and yet effective and viable for inference to be made through a Gibbs sampler. Examples of application of the proposed model and the sampling algorithm are provided for data modelled through Normal and Poisson distributions.

1.3 Outline

The remainder of this work is organized as follows. First, Chapter 2 provides a summary of the technical background used in our work and of the existing work related to regionalization. Next, in Chapter 3 we describe how we incorporate the spanning trees as a tool to perform sampling of partitions. In Chapter 4 we describe our proposed model, integrating the sampling through spanning trees with the product partition model, as well as the application of our model to Normal and Poisson data. Chapter 5 presents our attempt to construct a more complex model, based on a Markov Random Field. This provided us with useful insights and better understanding of the problem. However, this promising model did not perform as well as our final model. In Chapter 6, we present our experimental evaluation of the proposed model. We discuss the datasets considered in the studies and the results obtained with our algorithm. We also provide a comparison with alternative techniques. Finally, Chapter 7 concludes this thesis with a review of our main contributions and the discussion of the results obtained through the experimental evaluation.

Chapter 2

Background and Theoretical Framework

The goal of this chapter is to introduce the elements needed to understand both the problem we have at hand as well as the proposed solution. We also take a moment to review previous literature that is related to the problem and the main current solutions to it.

Throughout this text, we attempt to maintain a consistent notation. We use bold font to denote vectors of random variables and Greek letters to denote parameters in a model. We denote the data vector by $\boldsymbol{Y} = (Y_1, \cdots, Y_n)$. Hyperparameters and indices are denoted by lower case letters. A partition of a set of indices is denoted by the Greek letter $\boldsymbol{\pi}$ and $\mathcal{T}$ represents a spanning tree. $\mathcal{G}_k$ denotes the k-th cluster (or the subgraph induced by the nodes in the group $\mathcal{G}_k$ when talking about graphs). We use $f(\cdot)$ and $f(\cdot \mid \cdot)$ to denote marginal and conditional probability density function as well as probability mass function when in the continuous and discrete random vector case, respectively.

We start with Section 2.1, in which we give a textual description of the problem and the task we seek to solve in this work. This description builds upon what was already stated in the introduction and forms the base for the model we describe next. In Section 2.2, a formulation of the problem in terms of a graph is given. Such a modeling is helpful to understand the nature of the data and is an opportunity to define and explain in more detail the elements involved in both the problem as well as the proposed solutions. Next, in Section 2.3, a brief overview of some Markov Chain Monte Carlo (MCMC) sampling techniques is presented, to help those unfamiliar with such tools to follow the next chapters. Finally, in Section 2.5 we provide a review of the literature and the work that are related with our problem.

2.1 A description of the problem

2.1.1 Clustering

Cluster analysis (or simply *clustering*) is one of the main tasks in unsupervised machine learning. The goal of clustering is to partition a set of objects into groups (called *clusters*) in such a way that the objects of a given cluster are similar to each other and differ from the objects of other clusters.

Tasks or problems involving clustering are not recent, with examples that go back decades in disciplines as psychology, geology, marketing, and many other fields [Zubin, 1938; Tryon, 1939; Cattell, 1943]. The problem is also generic enough, with so many different assumptions and constraints, that several different approaches have emerged over the years.

Generally, the process of clustering consists of partitioning a set of n objects, each object with k features (or attributes). To define the clusters, a *similarity measure* is usually necessary, to indicate how similar are two objects. Frequently, this measure is some kind of *distance measure* between the objects, such as the *Euclidean distance* between numeric vector objects.

There are many variations on the clustering problem, due to the varied number of assumptions and constraints that take place. As a result, many algorithms with different approaches have been developed over the years. These different approaches can usually be divided into two categories: hierarchical and partitional methods. The hierarchical methods generate a nested series of partitions, while the partitional methods generate only one. This taxonomy can be further extended with descriptions such as: connectivity based (builds on the idea that objects are more similar to nearby objects than to farther objects), centroid based (represent the clusters through centroids - a mean vector), density based (define cluster by the density of the data nearby each object), graph based (work upon connectivity definitions in graphs), fuzzy clustering (each object belongs to each cluster to a certain degree), distribution models (model the clusters with statistical distributions), among others.

Another important type of clustering is the *constrained clustering*, which is directly connected to the problem which is the main focus of this work. In this kind of clustering constraints are imposed on how the objects can or cannot be grouped together. Usually these constraints impose that two objects must (or cannot) be in the same group. An example is the constrained k-means clustering [Wagstaff et al., 2001].

A survey on data clustering can be found in Jain et al. [1999], where not only the different types of clustering techniques are discussed but also other related topics such as pattern representation, feature selection, similarity measures and applications.

2.1.2 Spatially Constrained Clustering

A particular type of clustering is the *spatially constrained clustering*, also known as *regionalization*. In this problem, not only the similarity between the objects is taken into consideration, but also their spatial organization plays an important role on how the clustering process is done.

A typical scenario for this problem is when the data being clustered has its spatial characteristic derived from geographic properties. An example is a map divided into small areas. In this case, the goal is to group these small areas into larger regions, according to their similarity.

The main difference with respect to the usual clustering problem is that there is an additional constraint: while the objects are still clustered based on the similarity between themselves, the clusters formed must be spatially contiguous. The contiguity, in our example, is defined by the geographic neighbourhood.

It is important to highlight that with spatially constrained clustering we have two distinct spaces. There is a feature space, defined by the values and characteristics associated with the data, which are used to determine the similarity between objects. There is also the constraint space, defined by the spatial relationship present in the data, which restricts how the objects can be grouped together but is not directly used to compute similarity between them. This distinction is important because two very similar objects according to the feature space would be assigned to the same cluster in traditional clustering methods, but may end up in different clusters when the spatial constrain is taken into consideration.

A review of this problem can be found in the survey from Duque et al. [2007]. In this survey, the authors discuss many contributions to the area in the past decades and also provide a taxonomy of methods for solving regionalization problems. There are two main categories of methods: those that treat the spatial constraints implicitly and those which explicitly incorporate the constraints.

The first category includes traditional clustering methods as an initial step and only afterwards enforces the spatial constraints. This approach was initially proposed by Openshaw [1973]. However, in such techniques, the shape of the resulting clusters is highly dependent of the technique chosen to perform the initial clustering. Alternatives have been proposed with the objective of creating more compact regions, still without taking into account the spatial constraints. Weaver and Hess [1963] published one of the pioneer works in this subject, which proposes a procedure to generate political districts. Hess et al. [1965] formally presented this method.

In the second category, the methods explicitly consider the spatial constraints and usually model this problem as optimization problems, such as integer programming. Search for an exact answer to regionalization through an optimization problem is not computationally feasible [Altman, 1998]. Many attempts were made with different formulations of the problem as an optimization problem [Macmillan and Pierce, 1994; Garfinkel and Nemhauser, 1970; Mehrotra et al., 1998; Zoltners, 1979].

Since obtaining the exact solution is unfeasible, different heuristics were proposed such as starting regions from a (seed) area and grow them by adding neighbouring areas [Vickrey, 1961; Taylor, 1973; Openshaw, 1977a,b; Rossiter and Johnston, 1981] or start from an initially feasible solution and search for improvements [Nagel, 1965; Openshaw, 1977a, 1978; Browdy, 1990].

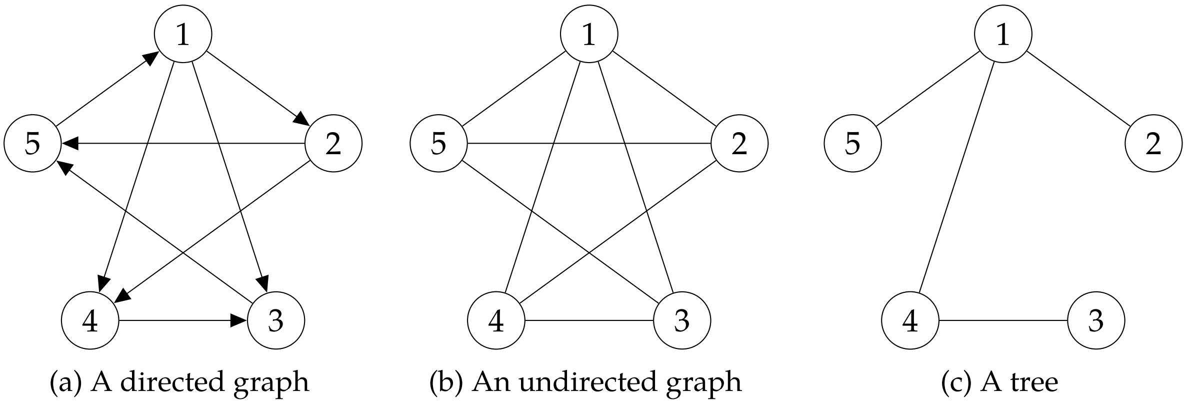
2.2 A graph representation

The modeling in terms of a graph becomes quite natural to handle this problem. There is a collection of objects which we want to cluster. There is one or more attributes associated with each of these objects. They also relate to each other in terms of a neighbourhood structure, which gives a constraint on how these objects can be grouped with each other. Our goal is to cluster the objects into groups such that the objects in a group are similar to each other whereas objects from different groups are dissimilar to each other.

The spatial constraint comes into play as we might have two objects which are very similar to each other, but are located on widely separated regions on the map and as such must not be grouped together in the same cluster.

2.2.1 Basic definitions

In this document, we define a **graph** as an ordered pair $G = (V, E)$ comprising a set V of **vertices** or **nodes** together with a set $E \subseteq V \times V$ of **edges** (i.e. the edges are pairs of vertices). An edge $e = (v_i, v_j)$ between the vertices v_i and v_j means that those vertices are *adjacent* to each other, or *neighbours*. An *undirected graph* is a graph in which the edges are an *unordered* pair of vertices. That is, the relation between two vertices doesn't depend on the direction or order of the vertices - if u is adjacent



(a) A directed graph (b) An undirected graph (c) A tree

Figure 2.1: Illustration of different graph types

to v, then v is also adjacent to u. Since the graphs we use in this documents are all undirected, we will refer to them simply as *graphs*, without using the word *undirected*.

A *path* in a graph is an alternating sequence of vertices and edges, beginning and ending with a vertex, such that all edges and vertices are distinct (with the possible exception of the first and last vertices) and each edge is incident with both the vertex immediately preceding it as well as the vertex immediately following it. A *cycle* is a closed path, i.e. a path whose first vertex is the same as the last. We say that two vertices are *connected* if there is a path between them. A graph is connected if every pair of vertices of this graph is connected.

A **tree** is a undirected connected graph without cycles. In a tree, any two vertices can be connected by a unique path. In Fig. 2.1 these different types of graph are illustrated.

A **connected component** (or just **component**) of an undirected graph G is a connected subgraph of G which is not contained in any connected subgraph of G with more vertices or edges than it has. In other words, if we grow the subgraph by adding vertices or edges of G that are not currently in it, the subgraph becomes disconnected. There is no vertex outside of the subgraph that can be connected to a vertex of the subgraph;

It is also possible for each edge of a graph to have a **weight** w_{ij} associated with it, in which case we call the graph a *weighted graph*. The *number of nodes* is given by $|V| = n$ and we denote by $u \sim v$ the adjacency relation between vertices u and v.

A **spanning tree** of a graph G is a tree that *spans* G. That is, it is a tree that has *all* the nodes of G and some of the edges of G. In other words, a spanning tree of $G = (V, E)$ is a subgraph $T = (V, E')$, with $E' \subseteq E$ in such a way that T is a tree.

The tree in Fig. 2.1c is a spanning tree of the graph in Fig. 2.1b. If the original graph G was not connected, the analogous of the spanning tree would be a *spanning* *forest* - a set of trees, each of them a spanning tree for one of the components of the graph.

2.2.2 The modeling of the data using graphs

The data is modeled in the following way: the objects (which represent random variables, random vectors or other data associated with each location) are represented by vertices, so for each object we have a vertex. The neighbourhood structure is modeled through the edges. That is, each edge (u, v) represents that nodes u and v are adjacent. This model is quite natural and is basically a translation of the data into a graph. The geographical neighbourhood structure in the data is mapped to the neighborhood in the graph.

As an example, in Fig. 2.2 we show the map of Belo Horizonte city, divided into its administrative regions. The resulting graph built from this map is superimposed in the figure.

Formally, we describe the model such as:

- The graph $G = (V, E)$ is built with V being the set of nodes v_i where each v_i represents an object (data point).

- The edge set $E = \{(u, v) \in V \times V \mid adj(u, v)\}$ where each edge represents the adjacency between two objects.

- Each vertex has a set of (one or more) attributes, which are the data associated with each object in the dataset. Thus, for each vertex v_i, we represent its attributes as $\boldsymbol{Y}_i$. These attributes can be a vector of attributes, a single attribute or can even be a set of features and a response variable.

2.2.3 The task in terms of the modeling

Considering the model we described, the task at our hands of clustering the data into groups where the spatial constraint is respected (i.e. groups where each object is adjacent to another object of the group) can be translated into the task of partitioning the graph.

In a more formal way, what we want is to partition the set of vertices V into c distinct groups, $\mathcal{G}_k, k = 1, \cdots, c$, such that $V = \bigcup_{k=1}^{c} \mathcal{G}_k$ and $\mathcal{G}_i \cap \mathcal{G}_j = \varnothing, \forall\, i \neq j$.

To ensure the spatial constraint, we also assume that

$$|\mathcal{G}_k| > 1 \implies \forall \quad v_i, v_j \in \mathcal{G}_k \quad | \quad v_i \neq v_j \quad \exists \quad \text{path}(u, v).$$

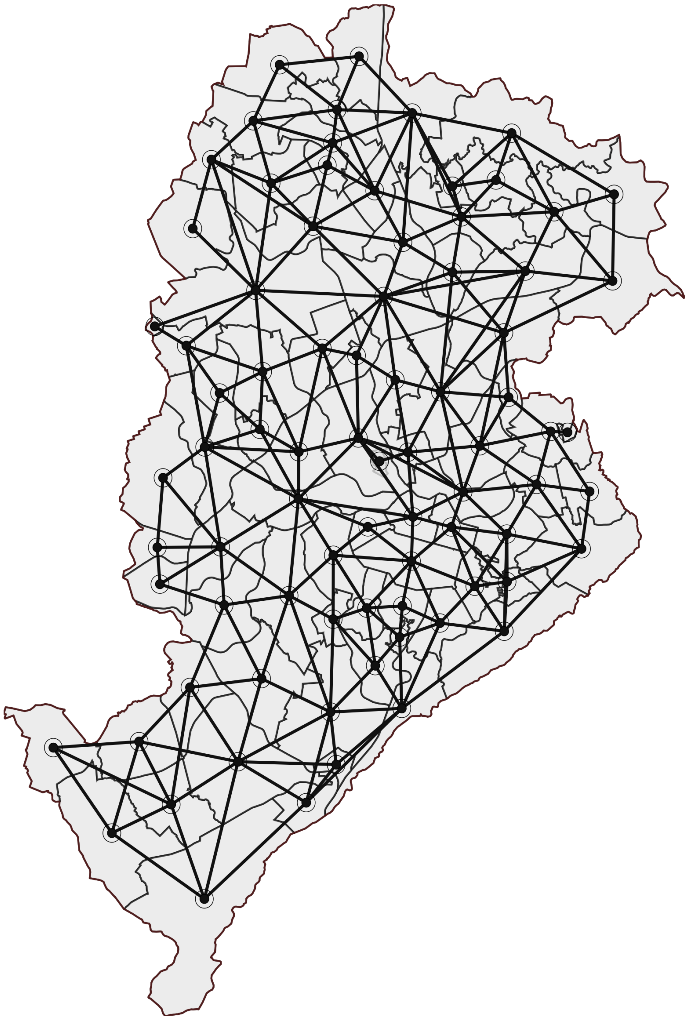


Figure 2.2: Example of the model as a graph for the administrative regions of Belo Horizonte

Another way to describe this constraint is to say that each group $\mathcal{G}_k$ is a connected subgraph. If we remove from the original graph all the edges connecting nodes from different groups, the resulting graph will be a disconnected graph and for each of its connected components, the set of its nodes is exactly one of the groups defined by the partitioning.

This partitioning of the graph must not only respect the spatial constraint as it must be done in some way the vertices in a given group have similar attributes whereas vertices from different groups have dissimilar attributes.

The definition of what is meant by similar and dissimilar is not given here and it is usually one of the points where the algorithms to solve this problem differ. When we discuss the modeling of our proposed solutions (and the related works) this concept of similarity will become more clear.

2.3 Markov Chain Monte Carlo Methods

One important class of tools used in our solutions is the Markov Chain Monte Carlo (MCMC) methods. These algorithms for sampling from a probability distribution are based on the construction of an appropriate Markov chain.

The general idea behind MCMC methods is that we want to sample from a given probability distribution $f(x)$ which is complex and from which we don't know how to sample directly. The way these methods work is by constructing a Markov chain in a specific way such that the *stationary distribution* of the chain is the desired probability $f(x)$ from which we want to sample. The samples are taken, therefore, by carrying out a random walk on the constructed chain.

In this section we briefly describe the two main MCMC methods, namely the Metropolis-Hastings algorithm and the Gibbs sampler, which we used in our work. More information about these methods can be found in various sources, such as in the introductory paper by Andrieu et al. [2003] or in the book by Murphy [2012].

2.3.1 Metropolis-Hastings algorithm

This algorithm, initially proposed by Metropolis et al. [1953] and later extended to a more general case by Hastings [1970] presents a way of sampling from a target distribution $\mathbb{P}(x)$ (possibly multivariate) from which we can't directly sample. This distribution might be too complex, or it may be even not completely specified (i.e. it can be specified up to a normalizing constant).

The way the algorithm works is by using a different distribution $\mathbb{Q}(x)$, called a *proposal distribution* from which sampling is easy. The algorithm generates a sequence of samples where each new value is dependent only on the current sampled value (thus making it a Markov chain). At each iteration a *candidate* value is sampled from the proposal distribution given the current value. This new candidate is then accepted or rejected, according to an acceptance rule. If it is accepted, the candidate becomes the new sample value. If it is rejected, the candidate is discarded and the current value is reused in the next iteration.

As it was said, the algorithm is based on the construction of a Markov chain in such a way that the stationary distribution is exactly the target distribution $\mathbb{P}(x)$. To achieve such a distribution, some conditions must be met. The chain must be *ergodic* (i.e. *irreducible*, *aperiodic* and *positive recurrent*) to guarantee the uniqueness of the stationary distribution. The *detailed balance* is used as sufficient condition to make sure such stationary distribution will exist. The acceptance rule mentioned in the previous paragraph is defined to ensure these conditions are attained. The acceptance ratio, which is used to determine if the new candidate x' should be accepted or rejected in the Metropolis-Hasting algorithm, is given by

$$A(x^{(t-1)} \to x') = \min\left(1, \frac{\mathbb{P}(x')}{\mathbb{P}(x^{(t-1)})} \frac{\mathbb{Q}(x' \to x^{(t-1)})}{\mathbb{Q}(x^{(t-1)} \to x')}\right), \tag{2.3.1}$$

where $Q(x \to y)$ denotes the probability of the proposal distribution, that is, the probability of proposing the state y given the state x.

At each step of the process, a new candidate is sampled. Then, this acceptance ratio is computed. Finally, a uniform value between 0 and 1 is sampled and compared with the acceptance ratio to decide if the new candidate should be accepted or rejected. In Algorithm 1, the pseudo code for the Metropolis-Hastings algorithm can be seen.

Algorithm 1 The Metropolis-Hastings algorithm

```
procedure Metropolis-Hastings
    $x^{(0)} \leftarrow$ random initial value
    for $t \leftarrow 1, N$ do
        $x' \leftarrow$ sample from $\mathbb{Q}(x|x^{(t-1)})$
        $A(x^{(t-1)} \to x') = \min\left(1, \frac{\mathbb{P}(x')}{\mathbb{P}(x^{(t-1)})} \frac{\mathbb{Q}(x' \to x^{(t-1)})}{\mathbb{Q}(x^{(t-1)} \to x')}\right)$
        $u \leftarrow U(0,1)$
        if $u \leq A(x^{(t-1)} \to x')$ then
            $x^{(t)} \leftarrow x'$
        else
            $x^{(t)} \leftarrow x^{(t-1)}$
        end if
    end for
end procedure
```

The selected proposal distribution should be, ideally, as similar as possible to the target distribution in order to avoid a frequent rejection of candidates, which would slow down the convergence of the chain. It is also important to highlight that, in the acceptance ratio, both directions in the proposal distribution are needed (i.e. we need both $\mathbb{Q}(x' \to x^{(t-1)})$ and $\mathbb{Q}(x^{(t-1)} \to x')$). It is also worth to note that in the case of these two values being the same (i.e. the distribution is symmetric), the algorithm becomes the simpler Metropolis algorithm, where the proposal distribution density is not needed to compute the acceptance ratio.

2.3.2 Gibbs sampler

Gibbs sampler (or Gibbs sampling) was introduced by Geman and Geman [1984] and named after the physicist Josiah Willard Gibbs, in reference to an analogy between the sampling algorithm and statistical physics. It is a special case, in its basic incarnation, of the Metropolis-Hastings algorithm.

The Gibbs sampler is particularly useful when the probability from which we want to sample is multivariate and it is unknown or difficult to sample from it, whereas the full conditional distribution of each variable is known and easy to sample from.

The Gibbs sampler provides a way for sampling from the distribution $\mathbb{P}(\boldsymbol{x})$ by sampling each of its components x_i at a time, from the full conditional distribution.

Suppose we want to obtain a sample from a joint distribution $f(x_1, \cdots, x_n)$. The i-th sample is denoted by $\boldsymbol{X}^{(i)} = (x_1^{(i)}, \cdots, x_n^{(i)})$. The algorithm proceeds as follows:

- Start with some initial state $\boldsymbol{X}^{(0)}$.
- For the i-th sample, sample each dimension $j = i, \cdots, n$ from the full conditional distribution:

$$\mathbb{P}(x_j \mid x_1^{(i)}, \cdots, x_{j-1}^{(i)}, x_{j+1}^{(i-1)}, \cdots, x_n^{(i-1)}). \tag{2.3.2}$$

It is important to note that we always use the most recently sampled value for each variable and update the value of a variable as soon as a new value is sampled.

The Gibbs sampler is a special case of the Metropolis-Hastings algorithm. Take the proposal distribution as that in Eq. (2.3.2). Then, the Metropolis-Hastings ratio in Eq. (2.3.1) is equal to 1 and the acceptance rate is also 1. That is, every sample is accepted.

It is also noteworthy that a single step of the Metropolis-Hastings algorithm can be used for variables from which the full conditional distributions are not easy to sample. There are other extensions for the Gibbs sampler. One of them is the *blocked*

Gibbs sampler, in which two or more variables are grouped together and sampled from their joint distribution conditioned on all other variables. Another one is the *collapsed Gibbs sampler* which integrates out (marginalizes over) one or more variables when sampling for some other variable.

2.4 Product Partition Model

In this section we review the product partition model (PPM), introduced by Barry and Hartigan [1992], which is a convenient framework to model data that follow different regimes.

In the PPM, it is assumed that observations in different components of a random data partition are independent and those inside a component are independent and identically distributed (iid). Moreover, the probability distribution for the random partition assumes a product form.

Let $\boldsymbol{Y} = (Y_1, \cdots, Y_n)$ be a set of observed data. Consider a random partition $\boldsymbol{\pi} = \{(Y_{1,1}, \cdots, Y_{1,n_1}), \cdots, (Y_{c,1}, \cdots, Y_{c,n_c})\}$ of this data into c groups of contiguous objects where the k-th group has n_k observations. Denote by $\boldsymbol{Y}_{\mathcal{G}_k} = (Y_{k,1}, \cdots, Y_{k,n_k})$ the observations in the k-th group.

We say that the random quantity $(\boldsymbol{Y}, \boldsymbol{\pi})$ follows a PPM, denoted by $(\boldsymbol{Y}; \boldsymbol{\pi}) \sim PPM$ if the following two conditions are met:

(i) The prior distribution of partition $\boldsymbol{\pi}$ into c blocks is given by the following product distribution:

$$\mathbb{P}(\boldsymbol{\pi}) = L \prod_{k=1}^{c} c(\mathcal{G}_k), \tag{2.4.1}$$

where $c(\mathcal{G}_k)$ is a nonnegative number which expresses the similarity among the observations belonging to group k, and L is a constant chosen so that the sum of $\mathbb{P}(\boldsymbol{\pi})$ over all possible partitions is unity. The component $c(\mathcal{G}_k)$ is named prior cohesion of group k and is a subjective choice.

(ii) Conditionally on the partition $\boldsymbol{\pi}$, the sequence $\boldsymbol{Y} = (Y_1, \cdots, Y_n)$ has the joint density given by:

$$f(\boldsymbol{Y} \mid \boldsymbol{\pi}) = \prod_{k=1}^{c} f_{\mathcal{G}_k}(\boldsymbol{Y}_{\mathcal{G}_k}), \tag{2.4.2}$$

where $f_{\mathcal{G}_k}(\boldsymbol{Y}_{\mathcal{G}_k})$ is the joint density of the random vector $\boldsymbol{Y}_{\mathcal{G}_k} = (Y_{k,1}, \cdots, Y_{k,n_k})$, called data factor. That is the block of n_k observations belonging to the k-th group.

A more interesting type of product partition model, which is considered in our work, is a model in which a set of parameters $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_n$ is partitioned into c groups, inducing a partition in the dataset. Given a partition, there are common parameters $\boldsymbol{\theta}_{\mathcal{G}_1}, \cdots, \boldsymbol{\theta}_{\mathcal{G}_k}$ of each group that are assumed to be independent. The joint distribution of the parameters, the data and the partition is a product partition model. The joint distribution of the parameters and the data, given a partition can be expressed as:

$$f(\boldsymbol{Y}, \boldsymbol{\theta} \mid \boldsymbol{\pi}) = \prod_{k=1}^{c} f(\boldsymbol{Y}_{\mathcal{G}_k} | \boldsymbol{\theta}_{\mathcal{G}_k}) f(\boldsymbol{\theta}_{\mathcal{G}_k}),$$

where

$$f(\boldsymbol{Y}_{\mathcal{G}_k} \mid \boldsymbol{\theta}_{\mathcal{G}_k}) = \prod_{i \in \mathcal{G}_k} f(\boldsymbol{Y}_i \mid \boldsymbol{\theta}_{\mathcal{G}_k}).$$

Thus, the joint distribution of the random variables in cluster $\mathcal{G}_k$, that is, the data factor, becomes:

$$f(\boldsymbol{Y}_{\mathcal{G}_k}) = \int_{\Theta} \left[\prod_{i \in \mathcal{G}_k} f(\boldsymbol{Y}_i | \boldsymbol{\theta}_{\mathcal{G}_k}) \right] f(\boldsymbol{\theta}_{\mathcal{G}_k}) \, \mathrm{d}\boldsymbol{\theta}_{\mathcal{G}_k},$$

the posterior by cluster of $\boldsymbol{\theta}_{\mathcal{G}_k}$, given $\boldsymbol{Y}_{\mathcal{G}_k}$, is

$$f(\boldsymbol{\theta}_{\mathcal{G}_k} \mid \boldsymbol{Y}_{\mathcal{G}_k}) = \frac{\left[\prod_{i \in \mathcal{G}_k} f(\boldsymbol{Y}_i \mid \boldsymbol{\theta}_{\mathcal{G}_k})\right] f(\boldsymbol{\theta}_{\mathcal{G}_k})}{f(\boldsymbol{Y}_{\mathcal{G}_k})}$$

and consequently, the marginal posterior of each individual parameter $\boldsymbol{\theta}_i$ becomes

$$\mathbb{P}(\boldsymbol{\theta}_i \mid \boldsymbol{Y}) = \sum \mathbb{P}(\boldsymbol{\theta}_{\mathcal{G}_k} \mid \boldsymbol{Y}_{\mathcal{G}_k}) \mathbb{P}(\mathcal{G}_k \in \boldsymbol{\pi} \mid \boldsymbol{Y}),$$

where the sum is over all possible partitions and $\mathcal{G}_k$ is the group that contain i. The posterior distribution for the partition assumes the form

$$\mathbb{P}(\boldsymbol{\pi} \mid \boldsymbol{Y}) \propto \prod_{k=1}^{c} c(\mathcal{G}_k) \cdot f(\boldsymbol{Y}_{\mathcal{G}_k})$$

and, finally, the posterior distribution for the number of clusters is given by

$$\mathbb{P}(C = c \mid \boldsymbol{Y}) = \sum I[\boldsymbol{\pi}, c] \mathbb{P}(\boldsymbol{\pi} \mid \boldsymbol{Y})$$

where the sum is over all possible partitions and $I[\boldsymbol{\pi}, c]$ is an indicator function assuming 1 if the partition $\boldsymbol{\pi}$ is composed of c clusters and zero otherwise.

The PPM thus offers a convenient framework to do inference on clustered parameters, since it implies some sort of conjugacy on the model. We explore this in order to perform the inference on both the parameters $\boldsymbol{\theta}_i$ as well as in the partitions of the observations.

2.5 Related Work

Many different methods and improvements have been proposed to deal with the problem of regionalization. Openshaw and Wymer [1995] proposed a simulated annealing variant of the k-means with a posterior step to enforce the spatial constraint. Variants of the *Automatic Zoning Procedure* (AZP) were proposed using different search procedures such as *simulated annealing*, and *tabu search* [Openshaw and Rao, 1995]. These methods work as an optimization problem and the number of regions to be constructed must be informed.

A modification of the AZP with tabu search was proposed by Duque and Church [2004] and is named *Automatic Regionalization with Initial Seed Location* (ARiSeL). Under this method, the construction of an initial feasible solution is repeated several times before running the tabu search. The author argue that constructing initial solutions is less expensive than performing a local search.

Kohonen [1990] proposed an algorithm called *Self Organizing Maps* (SOM), an unsupervised neural network which adjust its weights to represent a data set distribution on a regular lattice. This method has been used to perform regionalization. However, it doesn't guarantee the spatial contiguity. A variant of this method is proposed in Bação et al. [2004] with a different procedure to explore the neighbourhood. The same authors proposed a method which uses genetic algorithm to define the center of the regions [Bação et al., 2005].

Another heuristic is devised by Aldstadt and Getis [2006], called AMOEBA (*A Multidirectional Optimum Ecotope-Based Algorithm*). It starts with an initial area and grows it by adding neighbouring areas until a local spatial autocorrelation statistic fails to increase. This process is repeated to all areas and a final step resolves overlaps.

Another technique, called *Max-p-regions* works by constructing an initial feasible solution and performing local improvement. It clusters the areas into the maximum number of homogeneous regions such that the value of a spatially extensive regional attribute is above a predefined threshold value [Duque et al., 2012].

Assunção et al. [2006] proposed *Spatial 'K'luster Analysis by Tree Edge Removal* (SKATER), a graph based method that uses a minimal spanning tree to reduce the search space. The regions are then defined by the removal of edges from the spanning tree. The removed edges are chosen to minimize a dissimilarity measure.

Inspired by SKATER, Guo [2008] proposed REDCAP (*Regionalization with dynamically constrained agglomerative clustering and partitioning*) with six methods exploring different connection strategies using the underlying graph structure.

There are several statistical methods related to regionalization. In disease mapping, Knorr-Held and Raßer [2000] presented a Bayesian approach to the spatial partition of small areas into contiguous regions. They assume that each area has a random disease count y_i with Poisson distribution with unknown relative risks θ_i. They also assume these relative risks can be grouped into g spatial clusters where the θ_i's have the same value within a cluster. The number g of clusters is unknown and hence the unknown parameter vector is composed of g plus the distinct elements in the vector θ and some additional hyperparameters. Since the parameter vector has a variable dimension, they use reversible jump Markov chain Monte Carlo (MCMC) to obtain a sample from the posterior distribution [Green, 1995; Richardson and Green, 1997].

In the same year, Gangnon and Clayton [2000] proposed a different Bayesian approach for this problem. Let $r = (r_1, r_2, \ldots, r_n)$ be a specific cluster labelling function with c clusters. That is, r_i labels the cluster to which area i belongs so $r_i = j$ where $j = 1, \ldots, c$. They adopt a prior distribution $\mathbb{P}(c) \propto \exp(-\sum_j S_j)$ where S_j is a known function of the j-th cluster's geometry, $S_j = \alpha + f(\text{size}) + g(\text{shape})$, which penalizes clusters with large size or convoluted shapes. To make inference on the space of spatial clusters, they propose an algorithm to calculate an approximation to the posterior. This algorithm has two components. The first is a window of plausibility, an adaptation of the Occam's window approach to model selection [Madigan and Raftery, 1994]. In the second, given a window of plausibility, they use a randomized search algorithm similar to backwards elimination methods used for variable selection in regression problems.

Denison and Holmes [2001] also considered this problem. They use a Voronoi tesselation T with m centers (tiles or regions) to define a partition into spatial clusters. The possible centers of the tesselation are the n areas' centroids and hence the search space of the centers is drastically reduced. As in Knorr-Held and Raßer [2000], they also assume that all areas within a tesselation tile have the same distribution depending on an unknown θ. They assume *a priori* that the probability of a tessellation with M centers depends only on the number of centers.

Lu and Carlin [2005] and Banerjee and Gelfand [2006] have worked on a dual problem: rather than aggregating similar areas into homogeneous regions, they aim to identify sharp boundaries between pairs of areas. Homogeneous regions can be obtained as a byproduct of their procedures. They call their methods boundary (or wombling) analysis.

Hegarty and Barry [2008] proposed a model based on a PPM with cohesions built in such a way as to take the spatial information into account. While it is an interesting Bayesian method, the sampling of the partitions is not simple and a parameter that indicates whether there will be few or many clusters in the partition must be explicitly and carefully determined since giving it a prior distribution makes the calculations needed for the sampling of the partitions difficult.

Another interesting Bayesian method is proposed by Wakefield and Kim [2013]. In the proposed method, the maximum number of clusters has to be specified, the clusters found tend to have a circular shape and the model focus on defining only the clusters which have a "higher" or "lower" value. What means to have a higher or lower value has to be carefully specified through the prior distributions attributed to these two types of clusters.

Recently, Anderson et al. [2014] proposed a Bayesian regionalization method which takes two steps. In the first step, a hierarchical clustering method is used to define a set of possible partitions. In the second step the best partition is chosen by fitting a Bayesian model for each of the partition and evaluating them according to DIC, a criterion of model selection.

Chapter 3

Sampling partitions using Spanning Trees

One of the contributions and main point of this work is the use of a spanning tree as a tool to explore the space of partitions on which we have interest. We start by showing how we can use a spanning tree to simplify the search space of possible partitions. Then, we show how we sample from this space (both trees and partitions). Finally, we describe how this trick is used together with a PPM to make it an effective tool for analysing spatial data.

3.1 Using a spanning tree to explore the partitions

A major problem faced when dealing with partitioning of a dataset or a graph is the huge number of possible partitions that compose the search space. In this section we show how we can use a spanning tree as a tool to tackle this problem and make the exploration of this space of possible partitions something feasible. Back in Chapter 2 we showed a graph built for the administrative regions of Belo Horizonte (Fig. 2.2). For that graph, a possible spanning tree is portrayed in Fig. 3.1.

The basic idea behind using a spanning tree to generate partitions is that it provides a simple way to get a partition of the dataset conforming to the spatial constraint of having spatially coherent groups.

In a spanning tree, we have a set of $n-1$ edges which connect the nodes of the graph. To generate a partition of c spatially connected groups, the only thing we have to do is to remove $c-1$ edges from this spanning tree. Each edge we remove disconnects a part of the graph, creating a new component (a new group). In Fig. 3.2

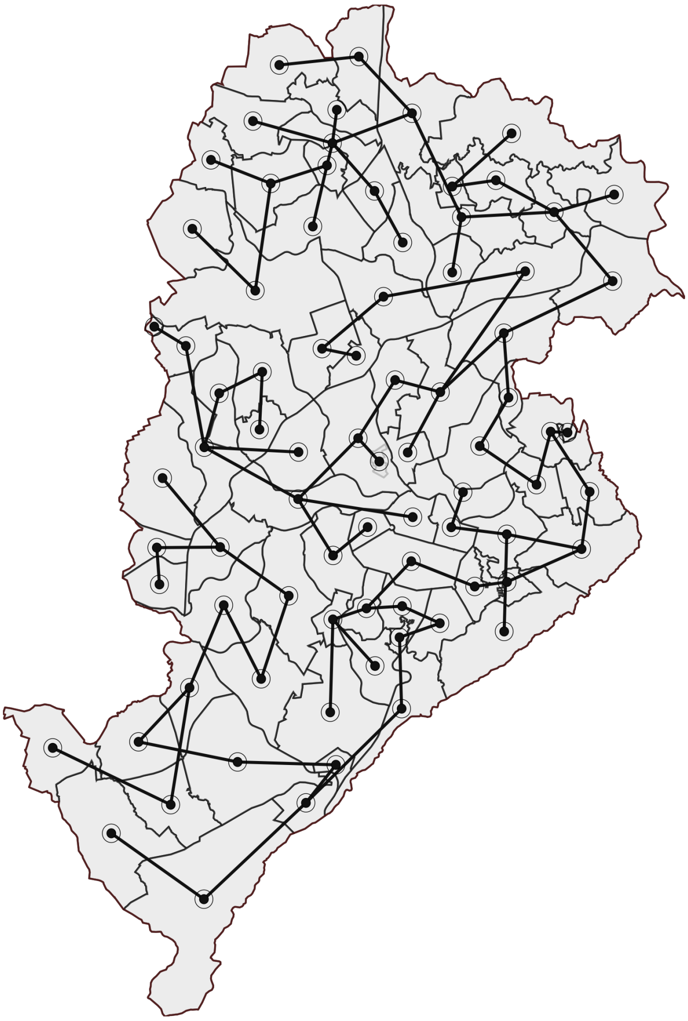


Figure 3.1: Spanning tree built for the Belo Horizonte data

we show a spanning tree with 5 groups defined by the removal of 4 edges (displayed in bold).

The great advantage we get from using the spanning tree and the edge removal is the reduction of the search space. Originally, we would have to navigate through a huge space of all the possible partitions of the dataset (having also to discard those that do not respect the spatial constraint). By using the spanning tree, we have now a simpler way of exploring the space. Moreover, all the partitions generated in this way will, by definition, respect the spatial constraint. After all, the spanning tree is built from the neighbourhood structure and, therefore, the groups formed will be spatially connected.

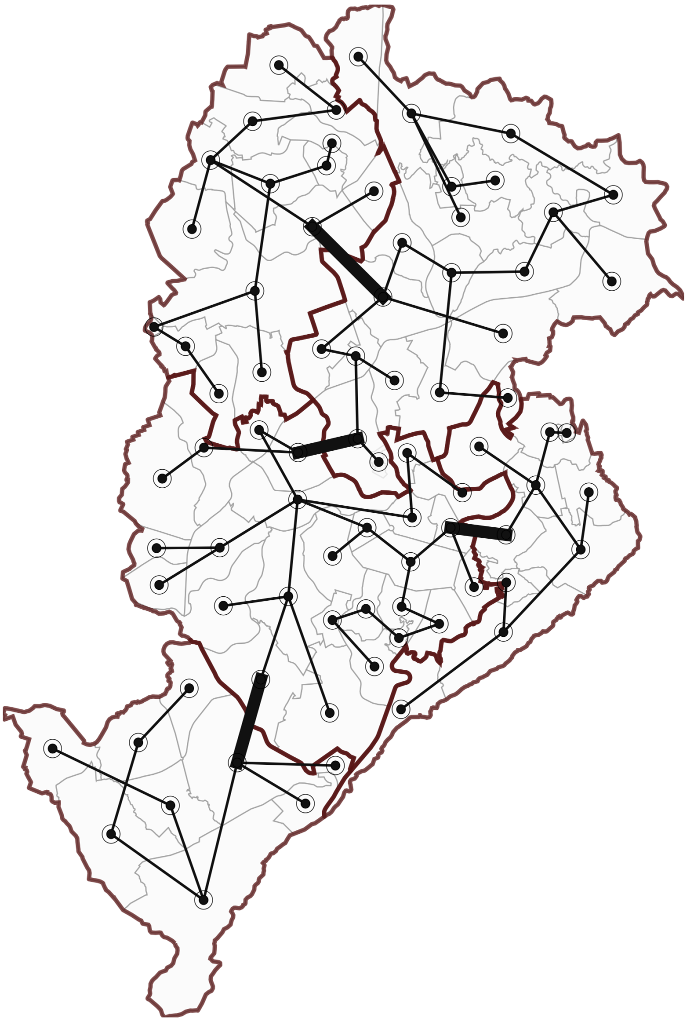


Figure 3.2: The groups generated by a spanning tree edge removal and the removed edges

3.2 Sampling partitions and tree

In this section we will discuss how to sample partitions of a given graph using spanning trees. We talk here about sampling on a generic context. We assume we have a connected graph G and we want to sample partitions for this graph.

3.2.1 Generic model for sampling partitions

A simple model for such a situation would be to consider all possible trees and partitions equally probable, that is, the distribution of spanning trees is uniform and the distribution of the partitions, given a specific tree $\mathcal{T}$, is also uniform. So we have the following hierarchical model:

$$\begin{aligned} \mathbb{P}(\mathcal{T}) &\propto 1 \\ \mathbb{P}(\pi \mid \mathcal{T}) &\propto \begin{cases} 1 & \text{if } \pi \prec \mathcal{T} \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{3.2.1}$$

where $\pi \prec \mathcal{T}$ means that π is compatible with $\mathcal{T}$, that is, the partition π can be achieved by removing edges from the tree $\mathcal{T}$.

An improvement to this model can be made by changing the distribution of the partitions given a tree. Instead of considering all partitions equally probable, we can model them giving larger probability to those partitions that have a certain amount of groups.

One way of doing that is adding a parameter ρ, which models the probability of removing an edge from the tree. In this model, it is as if, for each edge, a coin is tossed, individually, to decide if the edge should be removed or not. The probability of success of this coin is ρ. It can be defined as a fixed value or it can be given a prior probability distributions, such as a Beta distribution. This new model becomes:

$$\begin{aligned} \mathbb{P}(\mathcal{T}) &\propto 1 \\ \mathbb{P}(\pi \mid \mathcal{T}) &\propto \begin{cases} \rho^{(c-1)}(1-\rho)^{(n-c)} & \text{if } \pi \prec \mathcal{T} \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{3.2.2}$$

An immediate consequence is that the prior number of clusters is given by $n\rho$. The first model can be seen as a particular case of this new model, where the probability $\rho = 0.5$.

Even though sampling a partition given a tree, from this simple model, is a trivial task, it is usually a harder task when the model is more complex and involves more parameters and data. For this reason, we describe next a Gibbs sampler approach for the problem, which will be the basis of the sampling we do when dealing with a more complex model.

3.2.2 Gibbs sampler for the partition

The Gibbs sampler we describe here is inspired by the transformation suggested by Barry and Hartigan [1993]. A partition $\boldsymbol{\pi}$ of the graph can be transformed into a vector $\boldsymbol{U}$ of $n-1$ binary variables, given a compatible spanning tree, where each variable U_i represents one of the edges of the spanning tree and its value is 0 if the edge must be removed from the tree to form $\boldsymbol{\pi}$ and 1 otherwise.

With this transformation we can map a partition (given a compatible tree) to a vector $\boldsymbol{U}$ and back in a unique way. All possible partitions represent all possible configurations of this vector. And now, we have a multidimensional random variable which can be sampled with Gibbs sampler. For that purpose, we must be able to specify the full conditional probability $\mathbb{P}(U_i \mid \boldsymbol{U}_{-i}, T)$, where we denote by $\boldsymbol{U}_{-i}$ the set of all U_j where $j \neq i$, that is, $\boldsymbol{U}_{-i} = \{U_1, \cdots, U_{i-1}, U_{i+1}, \cdots, U_{n-1}\}$. What is needed, then, is to decide, for each edge in the tree, given all the other edges, if it should be removed or not. In a general case, it may be hard to specify the exact value of this probability, but since the variable can assume only one of two values, it is sufficient to know the ratio between the $\mathbb{P}(U_i = 1 \mid \boldsymbol{U}_{-i}, T)$ and $\mathbb{P}(U_i = 0 \mid \boldsymbol{U}_{-i}, T)$. We can further develop this

$$\begin{aligned}\mathbb{P}(U_i = u \mid \boldsymbol{U}_{-i}, T) &= \frac{\mathbb{P}(\boldsymbol{U}, T)}{\mathbb{P}(\boldsymbol{U}_{-i}, T)}\\ &= \frac{\mathbb{P}(U_i = u, \boldsymbol{U}_{-i} \mid T)\mathbb{P}(T)}{\mathbb{P}(\boldsymbol{U}_{-i}, T)}.\end{aligned}$$

so the ratio becomes:

$$\begin{aligned}R_i &= \frac{\mathbb{P}(U_i = 1 \mid \boldsymbol{U}_{-i}, T)}{\mathbb{P}(U_i = 0 \mid \boldsymbol{U}_{-i}, T)}\\ &= \frac{\mathbb{P}(U_i = 1, \boldsymbol{U}_{-i} \mid T)\cancel{\mathbb{P}(T)}\,\cancel{\mathbb{P}(\boldsymbol{U}_{-i}, T)}}{\mathbb{P}(U_i = 0, \boldsymbol{U}_{-i} \mid T)\cancel{\mathbb{P}(T)}\,\cancel{\mathbb{P}(\boldsymbol{U}_{-i}, T)}}\\ R_i &= \frac{\mathbb{P}(\boldsymbol{\pi}^{(1)} \mid T)}{\mathbb{P}(\boldsymbol{\pi}^{(0)} \mid T)},\end{aligned}$$

where $\boldsymbol{\pi}^{(1)}$ is the partition with the edge U_i and $\boldsymbol{\pi}^{(0)}$ is the partition without that edge.

Thus, we can sample from the distribution of U_i simply by sampling a uniform value $u \sim \text{Uniform}(0, 1)$ and using the following accept/reject criterion:

$$U_i = \begin{cases} 1 & \text{if } R_i \geq \frac{u}{1-u} \\ 0 & \text{otherwise} \end{cases} \tag{3.2.3}$$

In the case of the simple models described above in Eq. (3.2.1) and Eq. (3.2.2), this process is simpler and amounts to comparing the uniform value u to the probability ρ and remove the edge if $u \leq \rho$ or leave it in the tree otherwise.

In a more complex model, involving not only the tree and the partition, but also data and parameters, the sampling process is similar. The difference will be seen in the ratio R_i, which must be derived from the model. In this case, the sampling is done following Eq. (3.2.3).

3.2.3 Sampling the tree

Since we are using a Gibbs sampler, we must also be able to sample a new tree given a partition. The first step for that is to compute the full conditional distribution for the tree, which is given in Eq. (3.2.4).

$$\mathbb{P}(\mathcal{T} \mid \boldsymbol{\pi} = (U_1, \cdots, U_{n-1})) \propto \begin{cases} 1 & \text{if } \boldsymbol{\pi} \prec \mathcal{T} \\ 0 & \text{otherwise.} \end{cases} \tag{3.2.4}$$

From that equation, we can notice that the only possible trees to be sampled are those compatible with the current partition - that is, trees from which we can remove edges and get the current partition. These valid trees are uniformly distributed over the subset of trees compatible with the partition.

One way of sampling these trees is by following a hierarchical procedure. First we consider the subgraphs of each group. For each group, we take the subgraph induced by its vertices and sample a uniform spanning tree over this subgraph.

Now that we have a subtree for each group (uniformly generated), the next step is to connect the subtrees to form the final tree. This can be done by constructing a new graph where we add a vertex for each group and all the edges between two groups. That is, we have c vertices (one representing each group) and, for each original edge (u, v) where $u \in \mathcal{G}_i$ and $v \in C_j \neq C_i$, we add an edge in the new graph, connecting the vertices i and j. If there are multiple edges connecting two distinct groups in the original graph, they will be present as multiple distinct edges between the two vertices on the new graph, so that all possibilities are kept in the process.

Once the new (multi)graph is constructed, we sample a uniform spanning tree for this new graph. The edges selected in this spanning tree are included, together with the spanning trees constructed for each group, in the final spanning tree, which will then be the uniformly selected spanning tree, compatible with the current partition.

This procedure can be seen in Algorithm 2. The generation of uniformly distributed spanning trees is a subject of study since 1989 [Broder, 1989]. The main algorithm in the literature (Wilson's algorithm [Wilson, 1996]) is based on a random walk on the graph.

Algorithm 2 Sampling a uniform tree given a partition

```
procedure Sample-Tree( $\pi, G$ )
    $G' \leftarrow \varnothing$
    for $k \leftarrow 1, c$ do
        $T_i \leftarrow \text{UniformSpanningTree}(\mathcal{G}_k)$
        $G' \leftarrow \text{Vertex}(\mathcal{G}_k)$
    end for
    for $(u, v) \in E$ do
        if $\pi(u) \neq \pi(v)$ then
            $G' \leftarrow \text{Edge}(u, v)$
        end if
    end for
    $T_g \leftarrow \text{UniformSpanningTree}(G')$
    return $\mathcal{T} = T_g \cup T_1 \cup \cdots \cup T_c$
end procedure
```

We use, however, a different approach. Instead of using this procedure with Wilson's algorithm, we employ another procedure to sample the tree which uses a minimum spanning tree (MST) algorithm. This procedure gives good approximated result and is much simpler to understand and implement.

To sample a new tree, we first assign weights to the edges in the graph. The edges that connect vertices belonging to the same group receive a low weight, obtained from a uniform distribution which generates low values (e.g. between 0 and 1). The edges that connect vertices belonging to different groups receive a high weight, obtained from a uniform distribution which generates higher values (e.g. between 10 and 20). These values are arbitrary. What is necessary is that the weights for the edges connecting vertices from different groups must be higher than the weights of the other edges. Once the weights are assigned, the minimum spanning tree is obtained and it is the new sampled tree, compatible with the current partition.

The reason for using these two sets of values is that the algorithm computes the spanning tree with minimal sum of weights. When we use weights in this way, we ensure that the tree will be compatible with the partition, since the edges with higher weights are added to the tree only when all possible connections through edges with a lower weight are already explored. This way, it only adds a connection between clusters when all the possible connections inside a cluster have been visited.

This procedure is illustrated in Algorithm 3.

Algorithm 3 Sampling the tree using MST algorithm

```
procedure SAMPLE-TREE-MST($\pi, G$)
    for $(u, v) \in E$ do
        if $\pi(u) = \pi(v)$ then
            $w(u, v) \leftarrow \text{Uniform}(0, 1)$
        else
            $w(u, v) \leftarrow \text{Uniform}(10, 20)$
        end if
    end for
    $\mathcal{T} \leftarrow \text{MinimumSpanningTree}(G)$
    return $\mathcal{T}$
end procedure
```

This algorithm works because the MST algorithm (either by the Prim's algorithm or by the Kruskal's algorithm) selects the edges according to their weights. Since all edges separating groups have higher weights, they are only selected after the lower edges, which connect vertices inside a group. The random attribution of weights ensures that the tree will respect the partition and, since we randomly assign weights each time we sample a tree, we get a random tree, even though the algorithm is deterministic.

Chapter 4

Spatial PPM induced by spanning trees

In this chapter we introduce the Spatial Product Partition Model, a variation of the PPM adapted for spatial data. A spatial PPM was previously introduced by Hegarty and Barry [2008] where the prior cohesions were built in order to include the spatial associations among the neighbour areas. Our approach differs from this previous one because we join two main components: the traditional PPM (section 2.4) and the use of spanning trees (chapter 3). The use of the spanning tree imposes a kind of ordering in the spatial partition space which makes the sampling feasible for this spatial PPM.

We start in section 4.1 describing the model and how we adapted the traditional PPM to the spatial case, by using the spanning tree. Then, we describe the Gibbs sampler for this model, in section 4.2.

4.1 Proposed model

The Spatial Product Partition Model (SPPM) is built upon the simplified model presented in chapter 3, adding to it a model for the data based on its partitioning. As before, we use a spanning tree as a modeling tool which allows us to explore the space of possible partitions in a feasible way.

Let $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ be the observation set and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$ be the vector of parameters such that, given $\boldsymbol{\theta}$, $Y_1, \ldots, Y_n$ are independent and

$$Y_i \mid \boldsymbol{\theta}_i \sim f(Y_i \mid \boldsymbol{\theta}_i), \;\; i = 1, \ldots, n.$$

Assume that $(Y_1, \boldsymbol{\theta}_1), \ldots, (Y_n, \boldsymbol{\theta}_n)$ are nodes of a graph whose edges are defined by their spatial neighbour structure on the map. Let $I = \{1, \ldots, n\}$ be the set of indices of the nodes.

To introduce the cluster structure, let us assume that $\mathcal{T}$ is a spanning tree associated with that graph. Denote by $\boldsymbol{\pi}$ a partition of I compatible with $\mathcal{T}$ satisfying the conditions given in Section 2.2.3. Assume also that, given $\mathcal{T}$ and a partition $\boldsymbol{\pi} = \{\mathcal{G}_1, \ldots, \mathcal{G}_c\}$, there are common parameters $\boldsymbol{\theta}_{\mathcal{G}_k}$, $k = 1, \ldots, c$, such that, for all nodes whose indices belong to $\mathcal{G}_k$, we have that

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{\mathcal{G}_k}, \quad i \in \mathcal{G}_k.$$

As before, let us denote by $\boldsymbol{Y}_{\mathcal{G}_k}$ the set of observations associated with the nodes in $\mathcal{G}_k$.

We define a spatial PPM induced by the spanning trees as the joint distribution of $(\boldsymbol{Y}, \boldsymbol{\theta}, \boldsymbol{\pi})$, given $\mathcal{T}$ and $\boldsymbol{\pi} \prec \mathcal{T}$, denoted by $(\boldsymbol{Y}, \boldsymbol{\theta}, \boldsymbol{\pi} \mid \mathcal{T}) \sim$ SPPM, that satisfy the following conditions.

(i) Given $\mathcal{T}$ and $\boldsymbol{\pi} = \{\mathcal{G}_1, \ldots, \mathcal{G}_c\}, \boldsymbol{\pi} \prec \mathcal{T}$, the common parameters $\boldsymbol{\theta}_{\mathcal{G}_1}, \ldots, \boldsymbol{\theta}_{\mathcal{G}_c}$ are independent with joint distribution given by

$$\boldsymbol{\theta}_{\mathcal{G}_1}, \ldots, \boldsymbol{\theta}_{\mathcal{G}_c} \mid \boldsymbol{\pi}, \mathcal{T} \sim \prod_{k=1}^{c} f(\boldsymbol{\theta}_{\mathcal{G}_k});$$

(ii) Given $\mathcal{T}$, $\boldsymbol{\pi} = \{\mathcal{G}_1, \ldots, \mathcal{G}_c\}$ with $\boldsymbol{\pi} \prec \mathcal{T}$, and $\boldsymbol{\theta}_{\mathcal{G}_1}, \ldots, \boldsymbol{\theta}_{\mathcal{G}_c}$, the sets of observations $\boldsymbol{Y}_{\mathcal{G}_1}, \ldots, \boldsymbol{Y}_{\mathcal{G}_c}$ are independent and such that

$$Y_i \mid \boldsymbol{\theta}_{\mathcal{G}_k} \overset{ind}{\sim} f(Y_i \mid \boldsymbol{\theta}_{\mathcal{G}_k}), \; \forall \; i \in \mathcal{G}_k$$

(iii) Given $\mathcal{T}$, the prior distribution of $\boldsymbol{\pi}, \boldsymbol{\pi} \prec \mathcal{T}$ is a product distribution given by

$$\mathbb{P}(\boldsymbol{\pi} = \{\mathcal{G}_1, \ldots, \mathcal{G}_c\} \mid \mathcal{T}) = \frac{\prod_{k=1}^{c} c(\mathcal{G}_k)}{\sum_{\mathcal{C}} \prod_{k=1}^{c} c(\mathcal{G}_k)}$$

where $c(\mathcal{G}_k) \geq 0$ denotes the prior cohesion associated to the subgraph $\mathcal{G}_k$ and represents the similarity among the vertices in $\mathcal{G}_k$.

To complete the model specification we must elicit a prior distribution for $\mathcal{T}$. We assume that $\mathcal{T}$ has a uniform distribution on the space of spanning trees of the original graph G, that is,

$$\mathbb{P}(\mathcal{T}) \propto 1.$$

By considering this structure of spanning tree, we simplify the original graph structure in a way which makes it easy to form partitions by simply removing edges of this tree. Thus, the prior cohesion can be built as a function of weights of the edges, which can represent the probability of removing the edge. For instance, assume that all edges have equal weight ρ, that is, ρ is the probability of removing each edge.

$$c(\mathcal{G}_k) = \begin{cases} (1-\rho)^{n_{\mathcal{G}_k}-1}\rho & \text{if } k < c \\ (1-\rho)^{n_{\mathcal{G}_k}-1} & \text{if } k = c, \end{cases}$$

where $n_{\mathcal{G}_k}$ is the number of edges in $\mathcal{G}_k$ not removed. Consequently, the prior distribution of $\boldsymbol{\pi} = \{\mathcal{G}_1, \ldots, \mathcal{G}_c\}$, given $\mathcal{T}$ and ρ is

$$\mathbb{P}(\boldsymbol{\pi} \mid \mathcal{T},\, \rho) = \begin{cases} \rho^{(c-1)}(1-\rho)^{(n-c)} & \text{if } \boldsymbol{\pi} \prec \mathcal{T} \\ 0 & \text{otherwise.} \end{cases}$$

If we also assume that, *a priori*,

$$\rho \sim \text{Beta}(r, s),$$

the distribution for the partition $\boldsymbol{\pi}$ given the tree $\mathcal{T}$ takes a form that resembles the binomial distribution with a parameter ρ. Since the parameter ρ can be interpreted as the probability of removing each edge from the spanning tree, a bigger value for this parameter would indicate that we expect a high number of groups and a lower value has the opposite meaning.

Under the proposed model, we have that the posterior distribution of the partition, given $\mathcal{T}$ and ρ is

$$\mathbb{P}(\boldsymbol{\pi} = \{\mathcal{G}_1, \ldots, \mathcal{G}_c\} \mid \boldsymbol{Y}, \mathcal{T}, \rho) \propto \left[\prod_{k=1}^{c} f(\boldsymbol{Y}_{\mathcal{G}_k})\right] \rho^{c-1}(1-\rho)^{n-c}.$$

The joint posterior distribution of the common parameters $\boldsymbol{\theta}_{\mathcal{G}_1}, \ldots, \boldsymbol{\theta}_{\mathcal{G}_c}$, given $\boldsymbol{\pi}$ and

$\mathcal{T}$, is given by

$$\mathbb{P}(\boldsymbol{\theta} \mid \boldsymbol{\pi} = \{\mathcal{G}_1, \ldots, \mathcal{G}_c\}, \mathcal{T}, \boldsymbol{Y}, \rho) \sim \prod_{k=1}^{c} f(\boldsymbol{\theta}_{\mathcal{G}_k} \mid \boldsymbol{Y}_{\mathcal{G}_k})$$

which establishes a posterior conditional independence among them.

Under the structure of our Spatial PPM, the posterior distribution of $\boldsymbol{\theta}_i$, given $\mathcal{T}$, is

$$\mathbb{P}(\boldsymbol{\theta}_i \mid \mathcal{T}, \boldsymbol{Y}) \sim \sum \mathbb{P}(\boldsymbol{\theta}_i \mid \boldsymbol{Y}_{\mathcal{G}^*})\mathbb{P}(\boldsymbol{\pi} = \{\mathcal{G}_1, \ldots, \mathcal{G}_c\} \mid \boldsymbol{Y})$$

where $\mathcal{G}^*$ denotes the group containing i and the summation is over all partitions.

The posterior distribution for the number of groups conditionally in $\mathcal{T}$ is given by

$$\mathbb{P}(C = c \mid \mathcal{T}, \boldsymbol{Y}) = \sum_{\boldsymbol{\pi}} I[\boldsymbol{\pi}, c]\mathbb{P}(\boldsymbol{\pi} \mid \boldsymbol{Y}, \mathcal{T}, \rho).$$

where $I[\boldsymbol{\pi}, c]$ is an indicator function assuming 1 if the partition $\boldsymbol{\pi}$ has c clusters and 0 otherwise.

Next we describe how to sample from the posterior distribution in this model and how the model of the data and the parameters change the Gibbs sampler we presented in chapter 3.

4.2 Gibbs sampler for the SPPM

The sampling for the SPPM is quite similar to what we have already described in the chapter 3. What is different is that now the model is expanded to include not only the tree and the partition, but the data and the parameters as well.

Our goal is to sample $(\boldsymbol{\pi}, \mathcal{T}, \rho, \theta)$ given $\boldsymbol{Y}$. For that we will construct a sampling scheme that takes advantage of the proposed SPPM and uses a Gibbs sampler scheme very similar to the one we described earlier.

We begin by describing how to sample $(\boldsymbol{\pi}, \mathcal{T} \mid \boldsymbol{Y})$. To do this, we use a Gibbs sampler. We need then, to be able to sample from the full conditional probabilities, that is, we need to be able to sample $(\mathcal{T} \mid \boldsymbol{\pi}, \boldsymbol{Y})$ and $(\boldsymbol{\pi} \mid \mathcal{T}, \boldsymbol{Y})$. From our model, we have that:

$$\mathbb{P}(\mathcal{T} \mid \boldsymbol{\pi}, \boldsymbol{Y}) \propto \begin{cases} 1 & \text{if } \boldsymbol{\pi} \prec \mathcal{T} \\ 0 & \text{otherwise.} \end{cases}$$

This result is the same we found on chapter 3, and as such, we can sample from it as we have already described. Next, we have the full conditional distribution of the partition. Since our model is a PPM, we have that

$$\mathbb{P}(\boldsymbol{\pi} \mid \boldsymbol{Y}, \mathcal{T}, \rho) \propto \left[\prod_{k=1}^{c} f(\boldsymbol{Y}_{\mathcal{G}_k}) \right] \rho^{c-1}(1-\rho)^{n-c}. \tag{4.2.1}$$

From Eq. (4.2.1), if we integrate out the ρ parameter, we obtain that

$$\mathbb{P}(\boldsymbol{\pi} \mid \boldsymbol{Y}, \mathcal{T}) = \int_P \mathbb{P}(\boldsymbol{\pi} \mid \boldsymbol{Y}, \mathcal{T}, \rho)\, \mathbb{P}(\rho) \;\; \mathrm{d}\rho \tag{4.2.2}$$

$$= f(\boldsymbol{Y}) \frac{\Gamma(r+s)\,\Gamma(s+n-c)\,\Gamma(r+c-1)}{\Gamma(r+s+n-1)\,\Gamma(r)\,\Gamma(s)}$$

where, for each partition, the factor $f(\boldsymbol{Y})$ can be obtained by marginalizing the distribution of the data over the parameters as follows:

$$f(\boldsymbol{Y}) = \prod_{k=1}^{c} \int_{\Theta} f(\boldsymbol{Y}_{\mathcal{G}_k} \mid \theta_{\mathcal{G}_k}) f(\theta_{\mathcal{G}_k}) \, \mathrm{d}\theta_{\mathcal{G}_k}.$$

This factor is an important part of the Gibbs sampler we constructed and the marginalization of the parameter $\boldsymbol{\theta}_{\mathcal{G}_k}$ is one of the reasons we could achieve a Gibbs sampler which converges fast, is simple and needs no Metropolis step. Having a conjugate distribution for $\boldsymbol{\theta}_{\mathcal{G}_k}$ can be very helpful since the computation of this integral may become easier. We will provide two distinct models for $\boldsymbol{Y}$ and $\boldsymbol{\theta}$ which further exemplify this.

Once we have the formula for $\mathbb{P}(\pi \mid \mathcal{T}, \boldsymbol{Y})$, the sampling of the partition follows the same scheme we described before on chapter 3. Following this scheme, we need to compute, for each edge of the tree, the ratio between the probabilities of keeping or removing the edge. For the SPPM, this ratio then becomes:

$$R_i = \frac{f^{(1)}\left(\boldsymbol{Y}_{\mathcal{G}_k}\right)}{f^{(0)}\left(\boldsymbol{Y}_{\mathcal{G}_k}^{(L)}\right) f^{(0)}\left(\boldsymbol{Y}_{\mathcal{G}_k}^{(R)}\right)} \frac{(r+c-2)}{(s+n-c)}$$

where R_i is the ratio we compute for the i-th edge of the tree, $f^{(1)}\left(Y_{\mathcal{G}_k}\right)$ the prior predictive of the group formed when the edge i is present, whereas $f^{(0)}\left(\boldsymbol{Y}_{\mathcal{G}_k}^{(L)}\right)$ and $f^{(0)}\left(\boldsymbol{Y}_{\mathcal{G}_k}^{(R)}\right)$ are predictive of the two groups formed when the edge i is removed from the tree.

Thus, we have a Gibbs sampler we can use to sample from the distribution of $(\boldsymbol{\pi}, \mathcal{T} \mid \boldsymbol{Y})$. Once we have that, we can sample from $(\boldsymbol{\theta} \mid \boldsymbol{\pi}, \mathcal{T}, \boldsymbol{Y})$ and $(\rho \mid \boldsymbol{\theta}, \boldsymbol{\pi}, \mathcal{T}, \boldsymbol{Y})$ obtaining samples from the full set of parameters $(\boldsymbol{\theta}, \rho, \mathcal{T}, \boldsymbol{\pi})$ *a posteriori*.

When sampling ρ, we have only to sample from the conjugate beta distribution:

$$\rho \mid \boldsymbol{\theta}, \boldsymbol{\pi}, \mathcal{T}, \boldsymbol{Y} \sim \text{Beta}(r+c-1, s+n-c)$$

The sampling of $\boldsymbol{\theta}$ will depend on how the data is modeled. But a good characteristic of the PPM is that the groups are independent from each other and, usually, the sampling of these parameters will be simple and will depend only on the data of each group.

4.3 SPPM for Normal data

In this section we show an implementation of the SPPM for data modeled with a normal distribution. Here we instantiate the general framework we presented before to show how it can be applied to a specific case.

The bulk of the model remains the same as we showed before. What changes is the specification of a particular distribution for $\boldsymbol{\theta}$ and $\boldsymbol{Y}$. For this model, we assume that the data are a random sample of a normal distribution with mean μ and precision τ. We also assume that, *a priori*, the mean μ and the precision τ have their behavior jointly modeled by a Normal-gamma distribution. The parameters are clustered, which means, in this case, that we have distinct mean and precision for the normal distributions in the distinct groups.

Consequently, we have that

$$\begin{aligned} Y_i \mid \mu_{\mathcal{G}_k}, \tau_{\mathcal{G}_k} &\overset{iid}{\sim} \text{Normal}\left(\mu_{\mathcal{G}_k}, \frac{1}{\tau_{\mathcal{G}_k}}\right), \;\; \forall\, i \in \mathcal{G}_k \\ \theta_{\mathcal{G}_k} = (\mu_{\mathcal{G}_k}, \tau_{\mathcal{G}_k}) &\sim \text{Normal-Gamma}(m, v, a, b). \end{aligned}$$

To use the Gibbs sampler we presented before, we need the prior predictive distribution by clusters, which is given by

$$\begin{aligned} f(\boldsymbol{Y}_{\mathcal{G}_k}) &= \iint_{\Theta_{\mathcal{G}_k}} f(\boldsymbol{Y}_{\mathcal{G}_k} \mid \mu_{\mathcal{G}_k}, \tau_{\mathcal{G}_k}) f(\mu_{\mathcal{G}_k}, \tau_{\mathcal{G}_k}) \mathrm{d}\mu_{\mathcal{G}_k} \mathrm{d}\tau_{\mathcal{G}_k} \\ &= \frac{\Gamma\left(a + \frac{n}{2}\right) b^a}{\Gamma\left(a\right) (b + \frac{v}{2} + \frac{1}{2}\sum_{i \in \mathcal{G}_k} Y_i^2)^{a+\frac{n}{2}}} \sqrt{\frac{v(2\pi)^{n+2}}{(n+v)}} \exp\left\{ -\frac{v+n}{2} \left(\frac{\sum_{i \in \mathcal{G}_k} Y_i + vm}{n+v} \right)^2 \right\}. \end{aligned}$$

The distribution of $\theta_{\mathcal{G}_k} = (\mu_{\mathcal{G}_k}, \tau_{\mathcal{G}_k})$ given the observations in the cluster $\mathcal{G}_k$, is the updated Normal-gamma distribution

$$\theta_{\mathcal{G}_k} = (\mu_{\mathcal{G}_k}, \tau_{\mathcal{G}_k}) \mid \boldsymbol{Y}_{\mathcal{G}_k} \sim \text{Normal-Gamma}\left(m^*, v^*, a^*, b^*\right),$$

that is hierarchically given by:

$$\begin{aligned} \mu_{\mathcal{G}_k} \mid \tau_{\mathcal{G}_k}, \boldsymbol{Y}_{\mathcal{G}_k} &\sim \text{Normal}\left(m^*, \frac{1}{v^* \tau_{\mathcal{G}_k}}\right) \\ \tau_{\mathcal{G}_k} \mid \boldsymbol{Y}_{\mathcal{G}_k} &\sim \text{Gamma}\left(a^*, b^*\right), \end{aligned}$$

where $m^* = \frac{vm + n_k \overline{Y}_{\mathcal{G}_k}}{v+n_k}$, $v^* = v + n_k$, $a^* = a + \frac{n_k}{2}$, $b^* = b + \frac{1}{2}\sum_{i \in Y_{\mathcal{G}_k}} (Y_i - \overline{Y}_{\mathcal{G}_k})^2 + \frac{vn_k}{v+n_k} \frac{(\overline{Y}_{\mathcal{G}_k} - m)^2}{2}$, n_k is the number of observations in the group $\mathcal{G}_k$ and $\overline{Y}_{\mathcal{G}_k}$ denotes the average of the observations belonging to cluster $\mathcal{G}_k$.

4.4 SPPM for Poisson data

Another very popular and intensively used model in the literature is a Poisson distribution. This model assumes that the data follow a Poisson distribution and that the common parameter within a cluster has a Gamma distribution. That is, each group has a distinct parameter $\phi_{\mathcal{G}_k}$, and we assume that the data belonging to group $\mathcal{G}_k$ are independent with a Poisson distribution, with parameter $\lambda_i = E_i \phi_{\mathcal{G}_k}$, where E_i is a value known *a priori* for each i.

Under these assumptions, we have an expected value for each point (the E_i parameter) calculated under the assumption that there is no spatial variation in risk in the map. We assume that a given group will have an influence on how the actual data deviates from the expected value under no spatial risk variation. The group as a whole have values higher (or lower) than the expected. Therefore, this group deviation is what explicits the partitioning of the data. Formally, we assume that,

$$\begin{aligned} Y_i \mid \phi_{\mathcal{G}_k} &\overset{ind}{\sim} \text{Poisson}\left(E_i \cdot \phi_{\mathcal{G}_k}\right) \\ \phi_{\mathcal{G}_k} &\sim \text{Gamma}(a, b). \end{aligned}$$

For this model to be used with the Gibbs sampler we presented before, we need the prior predictive distribution for the observations $\boldsymbol{Y}_{\mathcal{G}_k}$, which is given by

$$\begin{aligned} f(\boldsymbol{Y}_{\mathcal{G}_k}) &= \int_0^\infty \left[\prod_{i \in \mathcal{G}_k} f(Y_i \mid \phi_{\mathcal{G}_k})\right] f(\phi_{\mathcal{G}_k}) \mathrm{d}\phi_{\mathcal{G}_k} \\ &= \left[\prod_{i \in \mathcal{G}_k} \frac{E_i^{Y_i}}{Y_i!}\right] \frac{\Gamma\left(a + \sum y_i b^a\right)}{\Gamma\left(a\right)\left(b + \sum Y_i\right)^{(a + \sum Y_i)}}. \end{aligned}$$

The distribution of $\phi_{\mathcal{G}_k}$ given the data information $\boldsymbol{Y}_{\mathcal{G}_k}$ in the cluster $\mathcal{G}_k$ is the updated Gamma distribution given by:

$$\phi_{\mathcal{G}_k} \mid \boldsymbol{Y}_{\mathcal{G}_k} \sim \text{Gamma}\left(a^*, b^*\right),$$

where $a^* = a + \sum_{i \in \mathcal{G}_k} Y_i$ and $b^* = b + \sum_{i \in \mathcal{G}_k} E_i$.

In this model, as well as in the one we presented earlier, we have a single set of parameters for each group. More complex models can be built where, for instance, there is a parameter for each observation, instead of a unique set for the whole group. But in such models, computing the integral in 4.2.2 becomes more difficult and if it cannot be easily calculated, we have to sample the partition through a Metropolis step. As we show next, in such scenario different challenges take place.

Chapter 5

MRF-SPPM

In this chapter we describe a more complex variation on our model which uses a Markov Random Field in order to more explicitly incorporate the spatial influence on the data and allows for individual parameters instead of the single common parameter for the cluster.

A Markov Random Field (MRF) is a graphical model which consists of an undirected graph in which the nodes represent random variables and the edges encode the conditional independence among the variables. In a MRF, a random variable is independent from all the other variables, given its neighbours in the graph. Given its structure, a MRF is a useful model for spatial statistics, thanks to its ability to encode the cyclic dependence between the variables, according to their spatial structure. For more information on this type of model and its application in spatial statistics, we refer to Rue and Held [2005].

This model still uses the same SPPM framework we presented earlier, with the main difference that the parameters associated to the clusters are, in fact, Markov Random Fields within each clusters, constructed from the same graph we use to define the spanning trees, which carries the spatial information of the data.

In the following sections we describe this model and the modifications we introduced to overcome the difficulty encountered to sample from it. We also present some arguments on the shortcomings of this more complex model and why they are arise in comparison to the simpler models we presented earlier.

5.1 The model

This model is an attempt to join the idea of having spanning trees as a tool to traverse the space of partitions with a model for the data in the form of Markov Random Fields, which bring the benefits of a Bayesian modelling but without the strong assumption that the observations within a given cluster are identically distributed or have their distribution dependent on set of parameters which is shared for all the observations in the cluster.

The motivation behind using Markov Random Fields to model the data is that these fields incorporate the spatial influence on the areas, since they are structured according to the spatial relationship present in the data.

In this model, we describe the data Y_i as coming from independent normal distributions, each of them with a distinct mean μ_i but with a shared precision τ_y:

$$Y_i \mid \mu_i, \tau_y \sim \text{Normal}\left(\mu_i, \frac{1}{\tau_y}\right).$$

The precision parameter is assumed to have a Gamma distribution:

$$\tau_y \sim \text{Gamma}(a_1, b_1).$$

The collection of all the μ_i forms a MRF. This field is further defined by the partition $\boldsymbol{\pi}$, the tree $\mathcal{T}$, and another precision parameter τ_μ.

$$\mathbb{P}(\boldsymbol{\mu} \mid \boldsymbol{\pi}, \mathcal{T}, \tau_\mu) \propto \tau_\mu^{\frac{(n-c)}{2}} \exp\left(-\frac{\tau_\mu}{2}\sum_{i\sim j}\delta_{ij}(\mu_i-\mu_j)^2\right).$$

The idea behind the prior for $\boldsymbol{\mu}$ is based on the conditional autoregressive model introduced by Besag et al. [1991]. For a thorough review of this kind of probability distribution, see Rue and Held [2005]. The less familiar $n-c$ in the exponent of τ_μ is due to the improper character of this prior distribution, and the definition of this prior is can be an arbitrary choice. For more discussion about it, we refer to Knorr-Held [2003], Hodges et al. [2003] and Lavine and Hodges [2012]. In this priori, c is the number of Markov connected component fields [Møller and Waagepetersen, 1998].

According to the partition, each cluster present on the data will be separated from the others. Thus, the interactions on the field are only between those adjacent areas which belong to the same cluster. The term δ_{ij} is an indicator which assumes the value of 1 when the areas i and j are spatially related (i.e. are adjacent in the tree) and belong to the same cluster, and 0 otherwise.

The precision parameter τ_μ controls how similar should be the means inside a cluster and is also modelled from a Gamma distribution:

$$\tau_\mu \sim \text{Gamma}(a_2, b_2).$$

For the partition $\boldsymbol{\pi}$, any partition compatible with the tree (i.e. any partition that can be obtained by removing edges from the tree) are given an equal probability, thus, having the following distribution:

$$\mathbb{P}(\boldsymbol{\pi} \mid \mathcal{T}) \propto \begin{cases} 1 & \text{if } \boldsymbol{\pi} \prec \mathcal{T} \\ 0 & \text{otherwise} \end{cases}$$

The trees are, also, uniformly distributed in the space of all spanning trees of the underlying graph.

$$\mathbb{P}(\mathcal{T}) \propto 1$$

Combined, these make the vector of parameters, $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\pi}, \mathcal{T}, \tau_y, \tau_\mu)$. The sampling of these parameters is made through the Metropolis-Hastings MCMC sampling technique. The posteriori distribution is given by:

$$\mathbb{P}(\boldsymbol{\theta} \mid \boldsymbol{Y}) \propto \left[\prod_{i=1}^{n} \mathbb{P}(Y_i \mid \mu_i, \tau_y)\right] \mathbb{P}(\boldsymbol{\mu} \mid \tau_\mu, T, \pi)\mathbb{P}(\pi \mid T)\mathbb{P}(\mathcal{T})\mathbb{P}(\tau_\mu)\mathbb{P}(\tau_y) \qquad (5.1.1)$$

from which we derive the full conditional distributions of the parameters:

$$
\begin{aligned}
(\mu_i \mid \boldsymbol{\mu}_{-i}, \tau_\mu, \tau_y, \mathcal{T}, \boldsymbol{\pi}, \boldsymbol{Y}) &\sim \text{Normal}\left(\frac{\tau_\mu \sum_j \delta_{ij}\mu_j + \tau_y Y_i}{\tau_y + \tau_\mu \sum_j \delta_{ij}}, \frac{1}{\tau_y + \tau_\mu \sum_j \delta_{ij}}\right) \\
(\tau_y \mid \boldsymbol{\mu}, \tau_\mu, \mathcal{T}, \boldsymbol{\pi}, \boldsymbol{Y}) &\sim \text{Gamma}\left(a_1 + \frac{n}{2}, b_1 + \frac{1}{2}\sum_i (y_i - \mu_i)^2\right) \\
(\tau_\mu \mid \boldsymbol{\mu}, \tau_y, \mathcal{T}, \boldsymbol{\pi}, \boldsymbol{Y}) &\sim \text{Gamma}\left(a_2 + \frac{n-c}{2}, b_2 + \frac{1}{2}\sum_{i\sim j} \delta_{ij}(\mu_i - \mu_j)^2\right) \\
\mathbb{P}(\boldsymbol{\pi} \mid \boldsymbol{\mu}, \tau_\mu, \tau_y, \mathcal{T}, \boldsymbol{Y}) &\propto \tau_\mu^{\frac{n-c}{2}} \exp\left(-\frac{\tau_\mu}{2}\sum_{i\sim j} \delta_{ij}(\mu_i - \mu_j)^2\right) \\
\mathbb{P}(\mathcal{T} \mid \boldsymbol{\mu}, \tau_\mu, \tau_y, \boldsymbol{\pi}, \boldsymbol{Y}) &\propto \exp\left(-\frac{\tau_\mu}{2}\sum_{i\sim j} \delta_{ij}(\mu_i - \mu_j)^2\right)
\end{aligned}
$$

The parameters τ_y, τ_μ and $\boldsymbol{\mu}$ can be directly sampled. On the other hand, the partition $\boldsymbol{\pi}$ and the tree $\mathcal{T}$ don't have a known distribution and a Metropolis-Hasting sampling is needed.

For the tree $\mathcal{T}$, we sample a tree in the same manner we described in chapter 3. This new tree is used as our proposal, which is then accepted or rejected according to the acceptance rate given by:

$$
A(\mathcal{T} \to \mathcal{T}' \mid \cdots) = \exp\left\{-\frac{\tau_\mu}{2}\sum_{i\sim j}(\delta'_{ij} - \delta_{ij})(\mu_i - \mu_j)^2\right\}
$$

For the partition $\boldsymbol{\pi}$, the proposed distribution consist in sampling a new partition by taking a step further from the current partition. First, it is decided if we are sampling a new partition with one more cluster or one less cluster. So, with probability p_s, we split one of the existing clusters, creating a new partition with one more cluster. Conversely, with probability $1 - p_s$ we, instead, merge two existing clusters, creating a new partition with one less cluster. The resulting partition is then accepted or rejected according with the following acceptance rate:

$$
A = \begin{cases}
\tau_\mu^{\frac{-1}{2}} \exp\left\{-\frac{\tau_\mu}{2}\sum_{i\sim j}(\delta'_{ij} - \delta_{ij})(\mu_i - \mu_j)^2\right\} \frac{(1-p_s)(n-c)}{c \cdot p_s} & \text{if split} \\
\tau_\mu^{\frac{1}{2}} \exp\left\{-\frac{\tau_\mu}{2}\sum_{i\sim j}(\delta'_{ij} - \delta_{ij})(\mu_i - \mu_j)^2\right\} \frac{p_s(c-1)}{(1-p_s)(n-c-1)} & \text{if merge}
\end{cases}
$$

Here we see an important difference from the simpler model. While with a shared cluster parameter we have a direct sampling method for the partition, here we must use a Metropolis-Hasting sampling. That is because it is very hard to derive a Gibbs sampling for this model as we did on the other model. Before, we had for each group, a single set of variables to describe the whole group, with this model, there is a variable for each node in the group. The integral we need to compute to marginalize the parameters becomes intractable, both because the number of variables is much higher, but also because the number of areas in the groups change from iteration to iteration.

5.2 Proposed modifications

In this section we describe the collection of modifications we did on this model while trying to overcome some of its limitations and get better results. Proposing these modifications and seeing their effect played an important role on better understanding the problem and the shortcomings of this model.

The first modification is to change the prior distribution for the partition. An uniform distribution for it is a generic assumption. But in this sort of problem, the number of clusters we expect to find is small if compared to the number of areas, so we choose a distribution which can better incorporate this idea. We introduced the following model for the partition probability:

$$\mathbb{P}(\pi \mid \mathcal{T}) = \begin{cases} \rho^{(c-1)}(1-\rho)^{(n-c)} & \text{if } \pi \prec \mathcal{T} \\ 0 & \text{otherwise} \end{cases}$$

with the introduction of a new parameter, ρ, modelled as a Beta variable:

$$\rho \sim \text{Beta}(r, s).$$

The reasoning for this distribution is the same we described in section 3.2. While the uniform distribution would have an expected number of cluster of $\frac{n}{2}$, with this new distribution we can change ρ to push the expected number of clusters towards a more reasonable value.

The second modification we proposed is in how we sampled the partition. In the initial model, to generate a candidate partition, we chose to split or merge cluster with a fixed probability, regardless of the current partition or how many clusters we expect to find.

To change this, we made the probability of choosing to split a cluster a value derived from the current partitions and the expected number of clusters. Considering that, with the new model for the partition, we expect to find $E_c = \rho n$ clusters, we used a logistic function to compute the probability of splitting a cluster:

$$p_s(c) = 1 - \frac{1}{1 + e^{-k(c - E_c)}}$$

This function transitions from 0 to 1, with the transition point centered on E_c, the expected number of clusters. The parameter k controls the steepness of the curve. Thus, this function gives a high probability of splitting when we have fewer clusters than expected and a low probability when we have more clusters than expected.

The last modification is also in the way we sampled the partition. In the initial model, once defined if we are splitting or merging clusters, we proceed by choosing an edge to either remove or reintroduce to the graph, respectively. This edge was being chosen uniformly amongst all the viable edges.

In this modification, we changed that to choose the edge to be removed or reintroduced by means of weighted sampling. We give each edge e_i a weight w_i computed by some measurement of similarity between the two areas it connect. Then, we sample the edges with each edge having the probability $\frac{w_i}{\sum_j w_j}$ of being chosen.

With this new method, we take into consideration the value ($\boldsymbol{Y}$) of the nodes as an indicative of similarity between the areas. The result is that the candidate partitions we generate tend to split on points where there is a dissimilarity in the tree and merge where there are similar areas.

5.3 Shortcomings of the model

Despite the modifications we discussed above, the result obtained with this model, even on clearly separated simulated data, was far from what we desired. In the results, the boundary between groups was respected (meaning that the sampled partitions separate nodes from distinct groups). But the model was overestimating the number of clusters.

What we noticed was that, despite all the changes we made, the real groups were subdivided in many smaller groups by the method. And what is worse, these groups would frequently be similar to each other, indicating that they should actually be joined, instead of separated. When we tried to understand the reasons for this poor performance, we noticed a few issues that can be used to explain the behavior we found.

The first problem we noticed is in the prior probability for $\boldsymbol{\mu}$. In this improper probability, the partition plays a strong role on how the actual values of μ are taken into consideration. The highest possible probability occurs when the partition separates all areas (creating n clusters). In this case, the values of μ aren't even taken into consideration (any value is equally probable). This has a huge impact on the posterior probability of the partitions. As can be seen in the acceptance rate for the sampling of the partition and in the posterior probability of the partition, whenever we propose a new partition, derived by splitting a group, we have a higher probability and bigger acceptance rate than when we have the converse (i.e. a new partition by merging two groups). For this reason, it is easier to accept a partition when split than when we merge. Specially because in a split, we remove a pair of vertices from the sum, while in a merge we take into account one more pair of vertices (possibly quite different).

Another problem we noticed is how the current partition $\boldsymbol{\pi}$ and means $\boldsymbol{\mu}$ shape the sampling of each other. When we are sampling the $\boldsymbol{\mu}$, since τ_μ is generally higher than τ_y (as we want homogeneous groups), the value of the means of the neighbours of an area plays a bigger role than the value of the data Y_i we have for that area. And because the neighbours are defined by the partition, there is a tendency to fit the means to the current partition, even if such a partition is a bad one.

This kind of problem is avoided in our previous models, because we marginalize out the parameters, removing their influence from the sampling of the tree. In this model, however, it is unfeasible to do such a thing and the sampling of the partition depends directly only on the value of $\boldsymbol{\mu}$ and not the data $\boldsymbol{Y}$. The influence of the data on the partition is indirect, through the means $\boldsymbol{\mu}$. But as we saw, we have a bigger influence of the means of the neighbours of the node on the sampling of μ_i then of the data Y_i associated with it.

Chapter 6

Experimental evaluation

In this chapter we analyze some simulated data sets considering our model. We start comparing the proposed model to some competing regionalization techniques well discussed in the literature. We also present two case studies in order to illustrate the practical use of the model developed in Chapter 4.

6.1 Simulations

For the simulations, different datasets were created. To define the spatial structure of the data (coordinates, shapes and spatial adjacency) we used the geographical neighbourhood of municipalities of Brazil. Assuming this spatial structure, the observations were generated with different scenarios. To generate the clusters and the observations we were inspired by the applications we present in Section 6.2.

There are six datasets which can be divided into three categories, according to the distribution used to generate the observations. The first category generates the observations according to a normal distribution. For the other two categories, the distribution used is a Poisson distribution. For each of the categories, two distinct methods were used to generate the data, one using distinct parameters for each observation and other using a common cluster parameter for all observations within a cluster.

Inspired by the applications we present in Section 6.2, the two Poisson categories adopt a model frequently used in epidemiology studies. What distinguishes the two categories is that while one uses a Poisson distribution with a parameter with a larger value, the other uses a Poisson distribution with a small value parameter, as it is the case when studying the incidence cases of a more frequent or a more rare disease.

Normal data: The datasets with normal data are composed of 853 areas, which we divided into three clusters: two big clusters covering almost all data and a third small (10 areas) cluster islanded in the middle of one of the two other clusters, with its value being strongly different from the surrounding areas. The observations were sampled from Normal distributions:

$$Y_i \sim \text{Normal}(\mu_i,\ \sigma_i^2).$$

In the first dataset, all observations within a cluster are *iid* from a single normal distribution, thus, the parameters μ_i and σ_i^2 are the same for all observations in the same cluster:

$$(\mu_j, \sigma_j^2) = (\mu_k, \sigma_k^2)\ \ \forall j \in \mathcal{G}_k.$$

The observations range from 0.5 to 0.8, with standard deviation of about 0.03 within each group.

For the second dataset, each area is independently distributed. The spatial structure is the same, with the same three clusters. In this dataset, however, the mean and precision used in each cluster was distinct, but yet very similar within the observations of each cluster. The observations range again from 0.5 to 0.8, with the standard deviations around 0.045 and the means ranging around 0.78, 0.7 and 0.63, with a smooth variation between the areas in the same group.

Poisson data with high rate: For this category, the data comprises a total 1188 areas with their neighbourhood structure derived from the spatial structure of the municipalities in the south region of Brazil. The observations are generated from a Poisson distribution with the parameters inspired by the models used in epidemiology studies and the model we used in the applications in Section 6.2. Each observation mimics a disease count (such as deaths counts by a specific cancer type) and they come from a Poisson distribution:

$$Y_i \sim \text{Poisson}(E_i \cdot \phi_i).$$

The factor E_i is the expected count (distinct for each area), while the factor ϕ_i represents a relative risk associated with the cluster, which offsets the values of the cluster from their expected value. In the first dataset in this category, all the observations within a cluster $\mathcal{G}_k$ share the same relative risk (i.e. $\phi_j = \phi_k,\ \forall j \in \mathcal{G}_k$). The dataset was divided into 10 clusters, with the relative risks ranging from 1.45 to 1.40.

The second dataset has, once again, a distinct parameter for each area. Instead of all the observations within a cluster having the same relative risk, they have each a distinct value for the parameter, although they are quite similar for the observations within a cluster. That is, instead of a common shared parameter, each observation has its own parameter, and they all vary, albeit smoothly, for the observations of a given cluster. The spatial structure is the same and the relative risks range from 0.75 to 1.45.

Poisson data with low rate: The datasets built in this category are generated in the same way as the previous one, except that the value used for the expected value of each area E_i is considerably lower than the ones used before. The same spatial neighbourhood is used, but this time the data is divided into 13 clusters. The procedures used to generate the two datasets in this category is the same as we described for the previous category. The dataset with the common parameter within the cluster has the values of the relative risk varying from 0.45 to 1.4, while the second dataset, with a distinct parameter for each area, has the relative risks varying from 0.45 to 1.75.

The two Poisson categories are very similar. The only difference between them (besides the partition of the data) is the expected value used for each area, which yields observations with different values. While the relative risks fall in the same range, the actual observations in the high rate Poisson are, in average, 8 times larger than those for the low rate Poisson.

6.1.1 Evaluation metrics

The evaluation of the cluster analysis results is not a simple task. Some usual metrics evaluate internal characteristics of the clusters such as the dissimilarity between the observations belonging to it. However, such approaches may not properly reflect the quality of the results. The validity of the metric used depends on both the assumptions of the structure of the dataset and the metric. For some datasets, the fact that the data within a cluster are dissimilar, to a certain degree, does not necessarily indicates a poor quality of the clustering.

The benefit of using simulated data is that we have the ground truth which we can use to evaluate the results, since we know the actual parameters and the true cluster structure used to generate the data. The metrics we use to evaluate the results are based on this information.

The first group of metrics we use are based on the estimation error of the parameters used to generate the data. We call θ_i the true parameter used to generate the data (the mean μ_i in the case of normal data and the relative risk ϕ_i on the Poisson data. We call $\hat{\theta}_i$ the estimated parameter. Under the usual regionalization methods, in the case of normal data, $\hat{\theta}_i$ is obtained by the averaging the observations Y_i in a cluster:

$$\hat{\theta}_i = \frac{1}{n_k} \sum_{j \in \mathcal{G}_k} Y_j \;\; \forall i \in \mathcal{G}_k.$$

For the Poisson data, $\hat{\theta}_i$ is the ratio between the sum of the observations Y_i and the sum of the expected number of cases E_i for the areas within a cluster:

$$\hat{\theta}_i = \frac{\sum_{j \in \mathcal{G}_k} Y_j}{\sum_{j \in \mathcal{G}_k} E_j} \;\; \forall i \in \mathcal{G}_k.$$

Under the SPPM we obtain a sample of the posterior distributions of these parameters. We used the average of these samples as the estimation $\hat{\theta}_i$ for the parameter θ_i.

The metrics based on the estimation of the parameter are as follows.

Mean Absolute Error (MAE): This metric measures the absolute difference between the real parameter used to generate the data and the estimation of this parameter according to the resulting partitioning. It is given by:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \mid \theta_i - \hat{\theta}_i \mid .$$

Mean Relative Error (MRE): This metric measures the relative difference between the real parameter used to generate the data and the estimation of this parameter according to the resulting partitioning. It is given by:

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^{n} \frac{\mid \theta_i - \hat{\theta}_i \mid}{\theta_i}.$$

Mean Squared Error (MSE): This metric measures the squared difference between the real parameter used to generate the data and the estimation of this parameter according to the resulting partitioning. It is given by:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\theta_i - \hat{\theta}_i)^2.$$

The other set of metrics are based on the difference between the real clusters used to generate the data and the clusters obtained by the different methods. We used an adaption of some traditional evaluation metrics used in classification tasks. In these metrics we consider every pair of data points. Each pair is classified as positive if they are in the same cluster and as negative if they are in different clusters. We define as true positives (TP) the number of such pairs which are in the same cluster both in the true partition as well as in the estimated one. False positives (FP) are the number of such pairs which were assigned to distinct clusters in the result but were actually in the same cluster in the true partition. True negatives (TN) are the number of pairs which were correctly assigned to distinct clusters and false negatives (FN) is the number of pairs which are incorrectly assigned to distinct clusters.

With this definitions, we obtained the following traditional metrics:

Rand measure: This measure can be viewed as the percentage of corrected assignments made by the algorithm and is given by

$$RI = \frac{TP + TN}{TP + FP + FN + TN}.$$

F_1 score: This score tries to balance the contribution of false negatives. We define *precision* as

$$P = \frac{TP}{TP + FP},$$

and *recall* as

$$R = \frac{TP}{TP + FN}.$$

The F_1 score is, then, the harmonic mean of these values, given by

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R}.$$

Jaccard index: This metric quantifies the similarity between the assignments. It is given by

$$JI = \frac{TP}{TP + FP + FN}.$$

Fowlkes-Mallows index: This index is the geometric mean of the precision and recall (as defined in the F_1 score), that is

$$FM = \sqrt{P \cdot R}.$$

6.1.2 Methods used in the comparison

In this simulation we evaluate the performance of our method and compare it to 8 different methods. The algorithms we use are: *SKATER*, through its implementation on the `spdep` R package[1], the original Automatic Zoning Procedure (*AZP*), and its variations with Simulated Annealing (*AZP_SA*), Tabu (*AZP_TABU*), Reactive tabu (*AZP_RTABU*), the Automatic Regionalization with Initial Seed Location (*ARISEL*), the Max-p-regions Tabu model (*MAXP*) and the "A Multidirectional Optimum Ecotope-Base Algorithm" (*AMOEBA*), all implemented in the Python ClusterPy[2] library [Duque et al., 2011]. A brief description of these methods can be found in Section 2.5. Most of these methods require as an input the predefinition of the number of regions to be generated. In all cases, we use three different values: the true number c of clusters, 3 less clusters and 3 more clusters than the true number.

Since our method generate a sample of the posterior distribution of the random partition instead of a single partition of the dataset, to evaluate the metrics that compare the cluster structure, we used a form of a summarization of this sample of partitions. This summarization is constructed as follows: we take the underlying graph of the spatial structure of the data and, for each edge (thus for each pair of neighbouring areas), we compute how often they were in the same cluster in the sampled partitions. This percentage is assigned to each edge. Then, we trim the edges by removing all those which are below a certain threshold. Once the infrequent edges are removed, the remaining components of the graph define the clusters. The reasoning is that the removed edges are exactly those which are frequently crossing the borders between clusters in the sampled partitions and, by removing them, the bulk of the clusters frequently present in the sampled partitions remain connected by the graph.

6.1.3 Results

Tables 6.1 to 6.6 show the model fit measures for the proposed model and the eight competitor methods previously mentioned for normal data (Tables 6.1 and 6.2)

[1] http://cran.r-project.org/package=spdep
[2] http://www.rise-group.org/

and Poisson data (Tables 6.3 to 6.6).

In Tables 6.1, 6.3 and 6.5 data are generated from the same distribution and in Tables 6.2, 6.4 and 6.6 we assume different parameters in each area. In Tables 6.5 and 6.6 we evaluate the methods in a scenario which simulates rare diseases situations.

For the SPPM, we ran the MCMC for 5000 iterations, skipping the first 500 samples as a *burn-in* period and thinning the result by taking only every 5th value. For the normal datasets, we used as the prior distribution a Normal-Gamma distribution with parameters $m = 0.65$, $v = 1$, $a = 400$ and $b = 1$. With this distribution, the standard deviation of the clusters concentrates around 0.05 and the means range from 0.55 to 0.75. With this, we can capture the range of the observations and the expected variation of the observations in the clusters. For the Poisson datasets, we used as the prior distribution the Gamma with parameters $a = 2$ and $b = 2$, which concentrates its mass around 1, as we would expect the relative risk to be. Furthermore, the $\text{Gamma}(2, 2)$ distribution has 90% of its probability mass concentrated between 0.18 and 2.37, which is a huge range for the relative risk of common human diseases.

In all the simulated datasets our model outperformed all the other methods. The only scenarios where our method had inferior results were in the normal dataset with common parameter, where the *MAE* for the SPPM was the second best and the Poisson dataset with low rate and common parameter where *SKATER* had the lowest MRE. However, in both cases, the SPPM had better performance according to all other metrics we consider to evaluate the models.

In all datasets, particularly in the Poisson datasets, the error metrics (*MAE*, *MSE*, *MRE*) for our method was from 1.5 to 5 times smaller than the other methods. In the other metrics our method achieved drastically better results as well.

Normal data with common parameter								
Method	**Parameter**	**MAE**	**MSE**	**MREL**	**RAND**	**F1**	**JI**	**FM**
SPPM	70%	0.00466	**0.00010**	**0.00701**	**96.50**	**96.51**	**93.26**	**96.51**
	80%				96.30	96.26	92.79	96.27
	90%				94.74	94.58	89.72	94.65
SKATER	$c-3$	0.00745	0.00024	0.01132	92.08	92.22	85.56	92.22
	c	0.00503	0.00016	0.00771	94.10	94.08	88.82	94.09
	$c+3$	0.00909	0.00027	0.01389	89.24	88.56	79.47	88.79
AZP	$c-3$	0.03526	0.00125	0.05304	50.34	66.92	50.28	70.87
	c	0.00488	0.00017	0.00751	92.52	92.34	85.77	92.39
	$c+3$	0.01158	0.00035	0.01799	84.98	83.11	71.10	83.85
AZP_SA	$c-3$	0.00488	0.00015	0.00751	96.03	96.07	92.44	96.07
	c	0.01902	0.00071	0.02865	67.15	62.10	45.04	62.93
	$c+3$	0.00849	0.00028	0.01304	81.07	77.92	63.82	79.14
AZP_TABU	$c-3$	0.02972	0.00104	0.04536	60.17	67.43	50.86	68.50
	c	0.00702	0.00021	0.01093	87.63	86.55	76.29	86.95
	$c+3$	0.01171	0.00037	0.01805	86.35	84.84	73.67	85.45
AZP_RTABU	$c-3$	0.02680	0.00108	0.03943	60.49	63.33	46.34	63.47
	c	0.00683	0.00023	0.01056	93.50	93.59	87.96	93.60
	$c+3$	0.00530	0.00022	0.00807	92.39	92.17	85.49	92.23
ARISEL	$c-3$	**0.00456**	0.00015	0.00701	96.02	96.07	92.43	96.07
	c	0.00945	0.00024	0.01481	88.77	87.92	78.45	88.23
	$c+3$	0.00542	0.00025	0.00824	86.91	85.72	75.01	86.14
AMOEBA	—	0.02497	0.00074	0.03770	70.89	64.90	48.04	66.45
MAXP	10	0.01243	0.00036	0.01879	51.45	7.89	4.10	19.15
	100	0.01551	0.00058	0.02317	65.74	55.54	38.44	58.36

Table 6.1: Model Fit, Normal data with common parameters

Normal data with distinct parameter								
Method	**Parameter**	**MAE**	**MSE**	**MREL**	**RAND**	**F1**	**JI**	**FM**
SPPM	70%	**0.00705**	**0.00015**	**0.01065**	50.37	66.58	49.90	70.32
	80%				**90.97**	**90.84**	**83.22**	**90.86**
	90%				90.33	89.63	81.21	89.92
SKATER	$c-3$	0.01086	0.00037	0.01624	87.45	87.55	77.85	87.55
	c	0.01131	0.00039	0.01702	86.06	85.93	75.33	85.94
	$c+3$	0.01263	0.00041	0.01927	82.92	81.75	69.14	81.99
AZP	$c-3$	0.01213	0.00046	0.01822	85.08	85.40	74.51	85.40
	c	0.03308	0.00130	0.04894	52.96	59.88	42.74	60.49
	$c+3$	0.01162	0.00043	0.01741	86.41	86.51	76.23	86.51
AZP_SA	$c-3$	0.01102	0.00037	0.01651	87.88	88.07	78.69	88.08
	c	0.01256	0.00049	0.01891	73.41	69.46	53.21	70.33
	$c+3$	0.01281	0.00053	0.01931	76.11	74.31	59.12	74.57
AZP_TABU	$c-3$	0.02277	0.00113	0.03506	65.63	69.11	52.80	69.42
	c	0.01512	0.00066	0.02280	78.56	78.98	65.27	78.99
	$c+3$	0.01566	0.00055	0.02374	73.43	68.39	51.96	69.77
AZP_RTABU	$c-3$	0.01386	0.00054	0.02068	83.51	83.73	72.02	83.73
	c	0.01317	0.00050	0.01978	84.15	84.15	72.63	84.15
	$c+3$	0.01394	0.00051	0.02076	79.09	77.26	62.95	77.61
ARISEL	$c-3$	0.01206	0.00044	0.01808	85.87	86.12	75.62	86.12
	c	0.01149	0.00040	0.01722	87.01	87.17	77.26	87.17
	$c+3$	0.01477	0.00047	0.02199	77.31	73.55	58.16	74.69
AMOEBA	*None*	0.03450	0.00144	0.05156	61.44	52.00	35.14	53.76
MAXP	10	0.01180	0.00034	0.01751	51.58	8.63	4.51	19.86
	100	0.01774	0.00053	0.02647	60.15	43.46	27.76	48.12

Table 6.2: Model Fit, Normal data with distinct parameters

Poisson data with high rate and common parameter								
Method	**Parameter**	**MAE**	**MSE**	**MREL**	**RAND**	**F1**	**JI**	**FM**
SPPM	70%	**0.03054**	**0.00422**	**0.03797**	49.14	42.08	26.65	51.24
	80%				76.54	59.65	42.50	63.71
	90%				**90.59**	**74.28**	**59.09**	**74.32**
SKATER	$c-3$	0.11544	0.02046	0.12527	74.76	54.92	37.86	58.11
	c	0.12579	0.03044	0.13527	75.50	54.12	37.10	56.62
	$c+3$	0.12618	0.03409	0.12778	77.03	54.06	37.04	55.77
AZP	$c-3$	0.14481	0.03003	0.15500	69.35	43.77	28.01	46.01
	c	0.10819	0.02484	0.11362	74.69	57.25	40.11	61.45
	$c+3$	0.13351	0.02751	0.14128	79.76	51.97	35.11	52.26
AZP_SA	$c-3$	0.17650	0.04653	0.21710	51.07	42.05	26.62	50.49
	c	0.07600	0.01620	0.08459	76.36	51.36	34.55	52.70
	$c+3$	0.11834	0.03139	0.12526	74.29	47.72	31.34	49.07
AZP_TABU	$c-3$	0.19176	0.06519	0.23379	49.84	36.52	22.34	42.84
	c	0.10617	0.02122	0.11570	73.60	51.01	34.23	53.47
	$c+3$	0.13359	0.02907	0.14364	81.11	48.89	32.35	48.90
AZP_RTABU	$c-3$	0.18137	0.05388	0.22075	53.29	41.53	26.20	48.90
	c	0.10348	0.02615	0.11161	78.00	47.37	31.03	47.60
	$c+3$	0.13224	0.03324	0.14001	78.60	50.97	34.20	51.45
ARISEL	$c-3$	0.13210	0.02655	0.14694	66.90	47.90	31.49	52.43
	c	0.10389	0.02524	0.11519	73.19	55.76	38.66	60.23
	$c+3$	0.12787	0.03623	0.13810	77.63	46.42	30.22	46.64
AMOEBA	*None*	0.16781	0.06013	0.18080	69.82	37.53	23.10	38.48
MAXP	10	0.08963	0.01851	0.10448	81.91	10.06	5.30	20.76
	100	0.12921	0.02821	0.15403	82.93	46.76	30.52	47.50

Table 6.3: Model Fit, Poisson data with high rate and common parameters

Poisson data with high rate and distinct parameter								
Method	**Parameter**	**MAE**	**MSE**	**MREL**	**RAND**	**F1**	**JI**	**FM**
SPPM	70%	**0.04759**	**0.00544**	**0.05465**	51.13	42.04	26.62	50.46
	80%				**89.02**	**70.40**	**54.32**	**70.41**
	90%				88.95	65.78	49.01	66.73
SKATER	$c-3$	0.09821	0.02133	0.10367	82.90	62.00	44.93	62.83
	c	0.10061	0.02474	0.10698	84.04	61.96	44.89	62.28
	$c+3$	0.10650	0.02619	0.11220	85.39	62.13	45.06	62.15
AZP	$c-3$	0.11725	0.02275	0.12573	72.38	46.57	30.35	48.37
	c	0.09930	0.01941	0.10935	71.55	50.88	34.12	54.29
	$c+3$	0.11223	0.02456	0.12804	74.61	43.48	27.78	44.05
AZP_SA	$c-3$	0.12101	0.02373	0.14110	70.97	39.75	24.81	40.73
	c	0.09564	0.01968	0.10301	76.76	56.78	39.65	59.50
	$c+3$	0.07958	0.01855	0.08658	83.88	56.34	39.22	56.35
AZP_TABU	$c-3$	0.10113	0.01917	0.11041	69.77	50.18	33.49	54.22
	c	0.14293	0.03248	0.17143	64.21	36.72	22.49	38.95
	$c+3$	0.09638	0.01827	0.10530	76.34	46.68	30.45	47.22
AZP_RTABU	$c-3$	0.11068	0.01971	0.12723	67.98	48.34	31.87	52.53
	c	0.10050	0.02104	0.11083	68.54	49.37	32.78	53.70
	$c+3$	0.12937	0.02724	0.14326	78.26	44.22	28.39	44.25
ARISEL	$c-3$	0.09152	0.01625	0.09966	76.59	48.94	32.40	49.74
	c	0.11073	0.02656	0.11981	79.64	51.77	34.93	52.07
	$c+3$	0.09866	0.02000	0.10814	76.19	51.87	35.01	53.40
AMOEBA	*None*	0.17683	0.05720	0.19835	67.94	35.11	21.29	36.13
MAXP	10	0.09528	0.01730	0.10998	81.93	10.68	5.64	21.19
	100	0.13384	0.02775	0.16170	80.19	41.26	26.00	41.56

Table 6.4: Model Fit, Poisson data with high rate and distinct parameters

Poisson data with low rate and common parameter								
Method	**Parameter**	**MAE**	**MSE**	**MREL**	**RAND**	**F1**	**JI**	**FM**
SPPM	70%	**0.07669**	**0.01905**	0.49370	65.43	78.99	65.27	80.78
	80%				65.98	79.17	65.52	80.87
	90%				**79.50**	**82.46**	**70.16**	**82.99**
SKATER	$c-3$	0.22172	0.12678	**0.36023**	49.73	52.20	35.32	53.72
	c	0.26417	0.16666	0.41019	46.72	47.16	30.85	49.27
	$c+3$	0.29476	0.23519	0.44595	45.43	44.73	28.81	47.16
AZP	$c-3$	0.24460	0.14382	0.74205	44.84	47.85	31.45	49.16
	c	0.23630	0.15310	0.60011	56.83	58.58	41.42	60.46
	$c+3$	0.27275	0.21656	0.78907	43.98	43.27	27.61	45.62
AZP_SA	$c-3$	0.22391	0.11542	0.74414	45.51	47.05	30.76	48.77
	c	0.18481	0.14430	0.66085	49.64	55.47	38.38	56.10
	$c+3$	0.17373	0.14279	0.67033	44.04	41.78	26.40	44.65
AZP_TABU	$c-3$	0.18362	0.12962	0.63674	51.98	60.42	43.29	60.58
	c	0.30429	0.18741	0.46918	45.31	38.46	23.81	43.40
	$c+3$	0.27051	0.19779	0.85028	47.51	45.23	29.23	48.42
AZP_RTABU	$c-3$	0.27233	0.17007	0.78498	44.19	44.95	28.99	46.86
	c	0.30857	0.20849	0.81854	43.09	43.67	27.93	45.59
	$c+3$	0.30699	0.21135	0.86227	43.84	35.09	21.28	40.60
ARISEL	$c-3$	0.29914	0.19268	0.81843	43.80	43.72	27.97	45.86
	c	0.28959	0.18852	0.43265	45.82	43.18	27.53	46.35
	$c+3$	0.30599	0.22116	0.88723	41.76	33.93	20.43	38.56
AMOEBA	*None*	0.53662	0.56779	0.66286	42.83	33.89	20.40	39.23
MAXP	10	0.16617	0.05139	0.60488	36.08	3.69	1.88	13.04
	100	0.12539	0.03611	0.68845	39.68	20.78	11.59	29.41

Table 6.5: Model Fit, Poisson data with low rate and common parameters

Poisson data with low rate and distinct parameter								
Method	**Parameter**	**MAE**	**MSE**	**MREL**	**RAND**	**F1**	**JI**	**FM**
SPPM	70%	**0.05172**	**0.00532**	**0.05553**	65.02	78.80	65.02	80.63
	80%				71.80	81.23	68.39	81.97
	90%				**82.34**	**84.77**	**73.57**	**85.40**
SKATER	$c-3$	0.17415	0.14741	0.19465	58.07	72.63	57.02	73.47
	c	0.21371	0.17941	0.24005	53.24	67.56	51.02	67.89
	$c+3$	0.24092	0.24516	0.27006	49.64	63.36	46.37	63.45
AZP	$c-3$	0.23537	0.14779	0.26733	43.52	53.24	36.28	53.39
	c	0.24854	0.17059	0.28081	42.97	38.19	23.60	41.85
	$c+3$	0.18657	0.19945	0.20594	48.30	51.36	34.56	52.69
AZP_SA	$c-3$	0.19384	0.11809	0.21444	45.09	38.37	23.74	43.20
	c	0.14145	0.11201	0.15493	48.34	59.72	42.57	59.72
	$c+3$	0.22061	0.22617	0.24595	39.32	29.60	17.37	34.37
AZP_TABU	$c-3$	0.20738	0.16752	0.22732	46.57	48.66	32.16	50.25
	c	0.21042	0.21353	0.22851	54.15	57.56	40.41	58.79
	$c+3$	0.30981	0.25766	0.34688	40.37	30.94	18.30	35.87
AZP_RTABU	$c-3$	0.24173	0.11225	0.26920	49.53	57.09	39.95	57.41
	c	0.26007	0.22601	0.29338	41.96	40.90	25.71	43.23
	$c+3$	0.28364	0.24032	0.32242	42.29	51.31	34.51	51.56
ARISEL	$c-3$	0.26916	0.18284	0.30273	42.89	46.43	30.23	47.59
	c	0.26754	0.19789	0.30433	40.72	48.16	31.72	48.62
	$c+3$	0.30933	0.26337	0.34726	42.22	37.86	23.35	41.28
AMOEBA	*None*	0.50147	0.55240	0.56333	42.28	34.78	21.05	39.38
MAXP	10	0.14022	0.03872	0.15519	35.97	3.46	1.76	12.44
	100	0.08065	0.01122	0.08758	40.55	23.89	13.57	31.97

Table 6.6: Model Fit, Poisson data with low rate and distinct parameters

6.2 Applications

In this section we show the results of applying our method to two case studies. The first application is the regionalization of the municipalities of Brazil according to their Human Development Index (HDI). The second application is the regionalization of the municipalities of the south of Brazil according to cancer mortality. These two applications exemplify the usage of the two different models introduced in Chapter 4.

6.2.1 Normal data: HDI

The first application we explore is the regionalization of a map of the Human Development Index (HDI) of municipalities in Brazil. The HDI is a statistic of life expectancy, education and income indices used to rank countries. The index used in this work is an adaption of the global HDI to the reality of the municipalities in Brazil. It was developed by the United Nations Development Programme (UNDP) in Brazil, joint with *Instituto de Pesquisa Econômica Aplicada* (IPEA) and *Fundação João Pinheiro* and uses data from the demographic census conducted by *Instituto Brasileiro de Geografia e Estatística* (IBGE).

The data is composed of the HDI of 5564 municipalities of Brazil. We consider the neighbourhood structure computed through the geographic adjacency to build the graph used in our algorithm. We assume for this data the model described in section 4.3, with the HDI of each municipality as the random vector $\boldsymbol{Y}$. In Figure 6.1 we show the data in form of a map.

For this experiment, we ran the MCMC for 10000 iterations, skipping the first 1000 samples as a *burn-in* period and *thinning* the result by taking only every 10th value. As the prior distribution of the parameters of each cluster, we use the Normal-Gamma with parameters $m = 0.65$, $v = 0.04$, $a = 100$ and $b = 1$. With this distribution, we concentrate the mass for the precision around 100, which yields a standard deviation of about 0.1 for the observations of the clusters. This distribution also favors the means for each cluster centered around 0.65, with a deviation of 0.5, which spans most of the range the HDI can take (which is from 0.0 to 1.0). This way, we expect that for each cluster, the observations fall in a range of 0.2 around its mean.

In Figure 6.2a we display a summary of the sampled partitions. This map was constructed taking into consideration the neighbouring municipalities which belonged to the same cluster in at least 80% of the sampled partitions.

Another visualization is present in Fig. 6.2b where each area is colored with

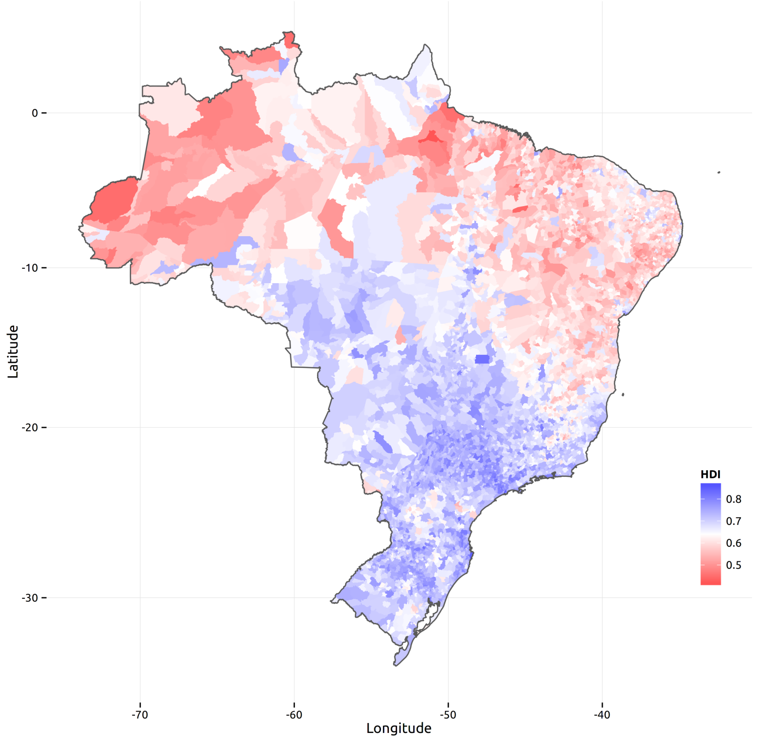
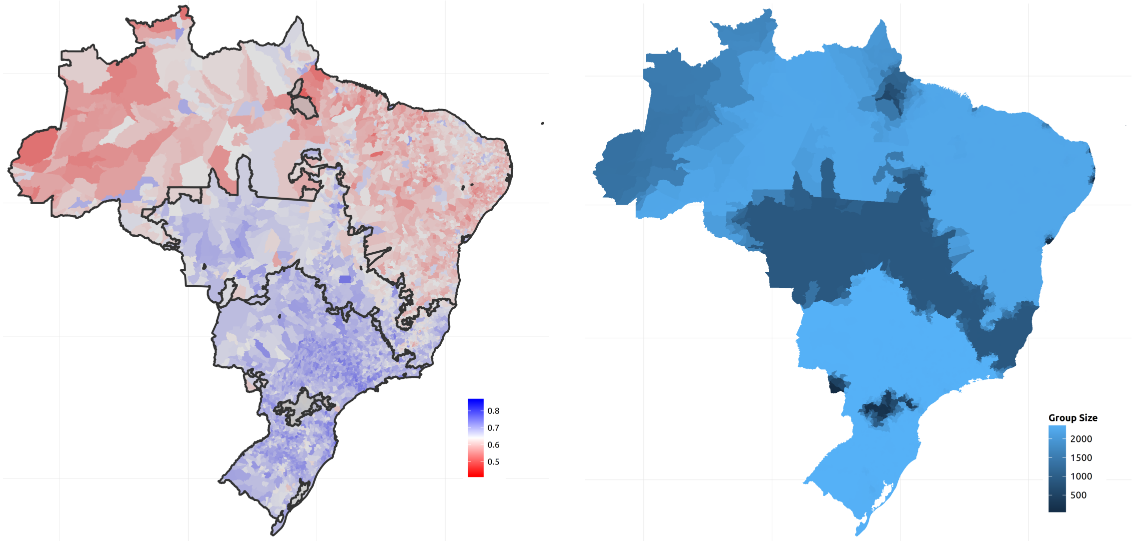


Figure 6.1: Map of municipalities of Brazil with HDI data



(a) Summary of the sampled partitions for Brazilian municipalities (HDI data)

(b) Average group size for each municipality in the sampled partitions

Figure 6.2: Regionalization of Brazilian municipalities according to their HDI

the average size of the cluster to which it belongs throughout the sample.

Through Figs. 6.2a and 6.2b we can see that the algorithm is able to group the data as we expected, and the groups are consistent through all the sampled partitions. These summary images show a partition which indicates three large groups with a number of smaller groups in the frontiers. Although the main groups are well defined, these images show that the separation between the clusters is not clearly defined, causing a certain level of noise. This may seem as some kind of problem of the algorithm but, in fact, this shows a natural characteristic of this kind of area where there is a transition between two distinct groups and frontiers are not really well defined.

In Figure 6.3 we show a sample of partitions generated by our algorithm. As it can be noticed, the main difference between samples is in general on the frontier between the clusters, which, as expected, is hard to be defined, since it is a transition area.

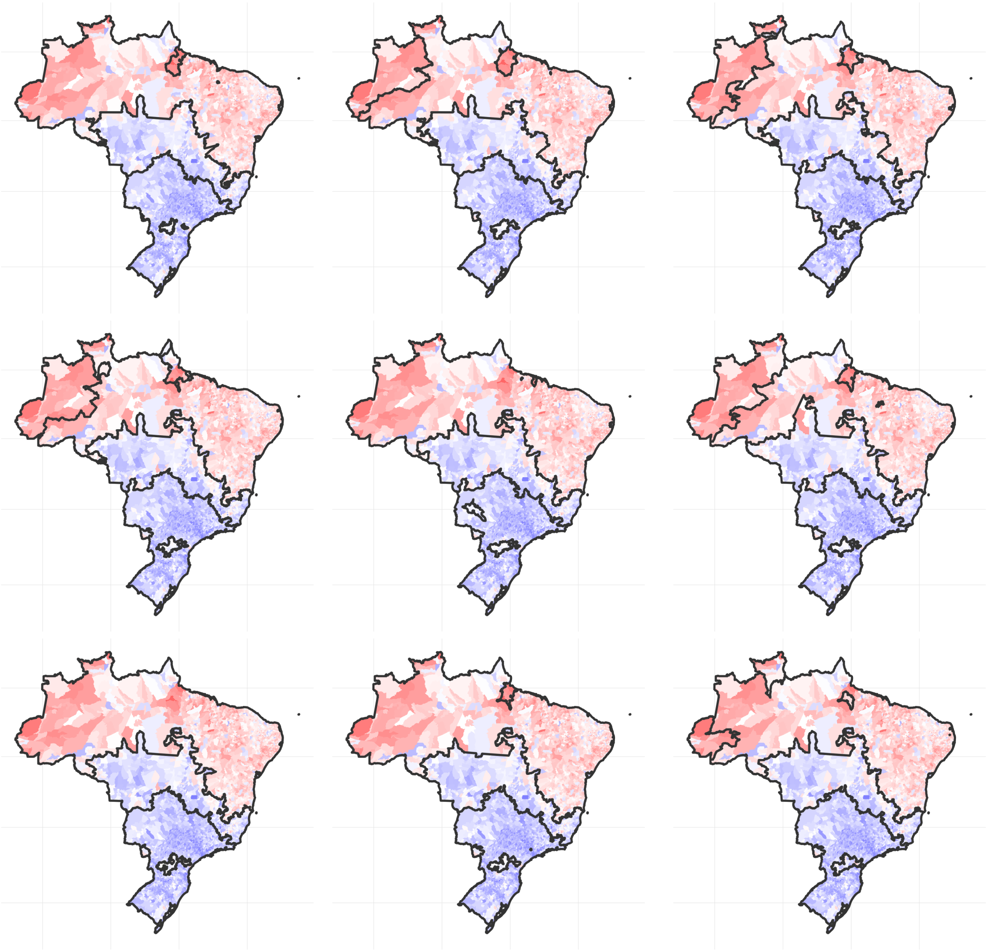


Figure 6.3: Some of the sampled partitions (HDI data)

Another interesting result is related to some tiny clusters present in the shore line which can be seen in both Figs. 6.2a and 6.2b. We checked the sampled partitions to find the areas which are frequently assigned to small clusters, that is, areas which are in clusters whose size is within 9000 squared kilometers in at least 90% of the sampled partitions. The interesting result is that those areas are the capital of states in northeast and their neighbouring cities. The northeast is a region where the HDI is generally lower, as can be seen in Fig. 6.1. The capital cities and their surroundings are regions where the HDI tends to be higher since they are the main economical region of the states as well as important tourism destinations.

6.2.2 Poisson data: Cancer deaths

The second application we studied is the regionalization of a map of deaths by cancer, in the south region of Brazil. In fact, two distinct regionalizations were performed. For the first, the data analyzed refers to deaths by lung cancer, while the second analyzed deaths by bladder cancer. These two types of cancers were selected duo to their different incidence. While both are common types of cancer, bladder cancer is almost an order of magnitude rarer than lung cancer. Due to this, the incidence rate for bladder cancer is more affected by small variations, which forms a scenario where the benefits of using a stochastic method instead of traditional approaches may be more easily seen.

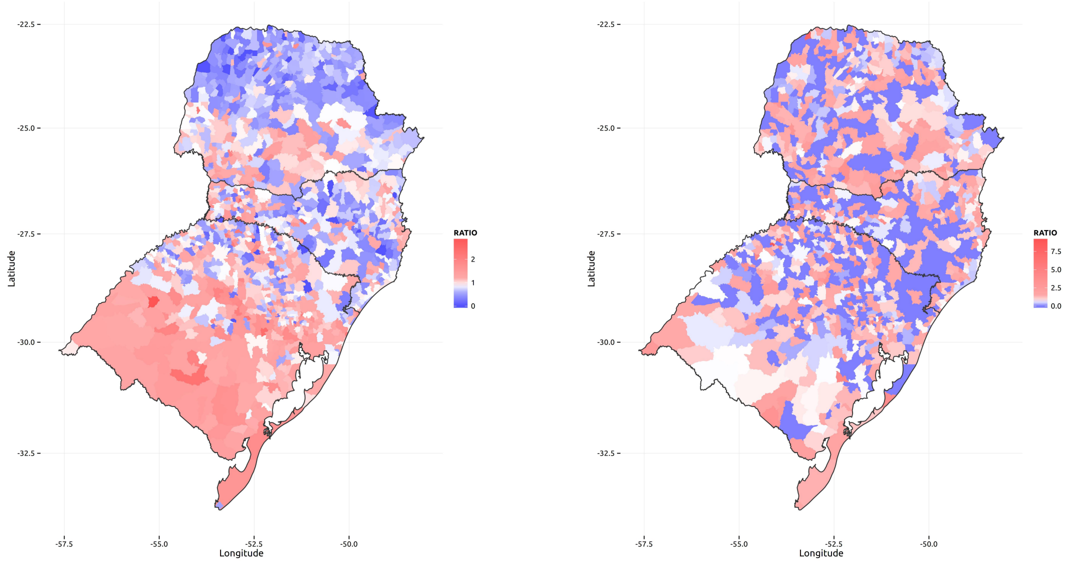
For both datasets, we obtained the number of fatalities by age group and gender, for each municipality, of the years 2008 - 2012 through DATASUS [3], the Department of Informatics of SUS (*Sistema Único de Saúde*), Brazil's publicly funded health care system. We also obtained demographic information of the same years, for the same age groups and gender, from the IBGE.

For each area we computed the expected number E_i of deaths, taking into consideration the demographics of each municipality and the number of deaths for each age group and gender. This value, together with the actual number of deaths in each area Y_i were used to perform the regionalization.

In Fig. 6.4 we show the ratio between the observed and the expected number of deaths by cancer. The reasoning behind the model is that the relative risk is the same within a cluster, that is, within a cluster the offset of the expected number of deaths is given by this common relative risk.

For this experiment, we ran the MCMC for 10000 iterations, skipping the first 1000 samples as a *burn-in* period and *thinning* the result by taking only every 10th value. For the distribution a priori of the parameters of each cluster, we use a Gamma with parameters $a = 1.1$, and $b = 1.1$. This distribution concentrates its mass around 1.1, with a variance of 0.91, so the relative risk is concentrated in values mostly be-

[3] http://datasus.saude.gov.br/



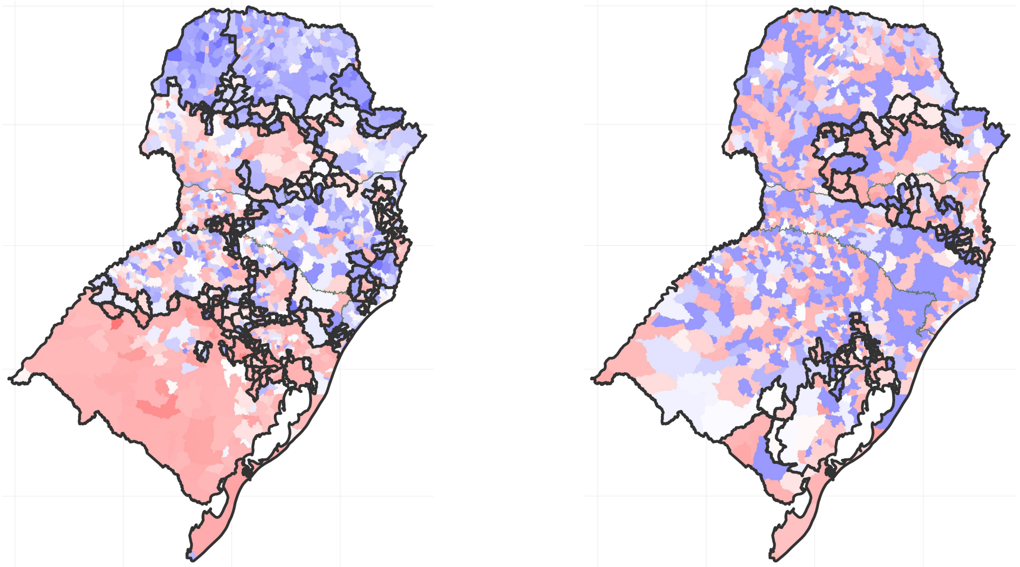
(a) Lung cancer

(b) Bladder cancer

Figure 6.4: Ratio between the actual and expected number of deaths by cancer

tween 0 and 2. This seems reasonable, as we expect that in a region the incidence rate deviates from the expected value by a factor of no more than 2.

In Fig. 6.5 we display a summary of the sampled partitions. This map was constructed taking into consideration the neighbouring municipalities which belonged to the same cluster in at least 85% of the sampled partitions.



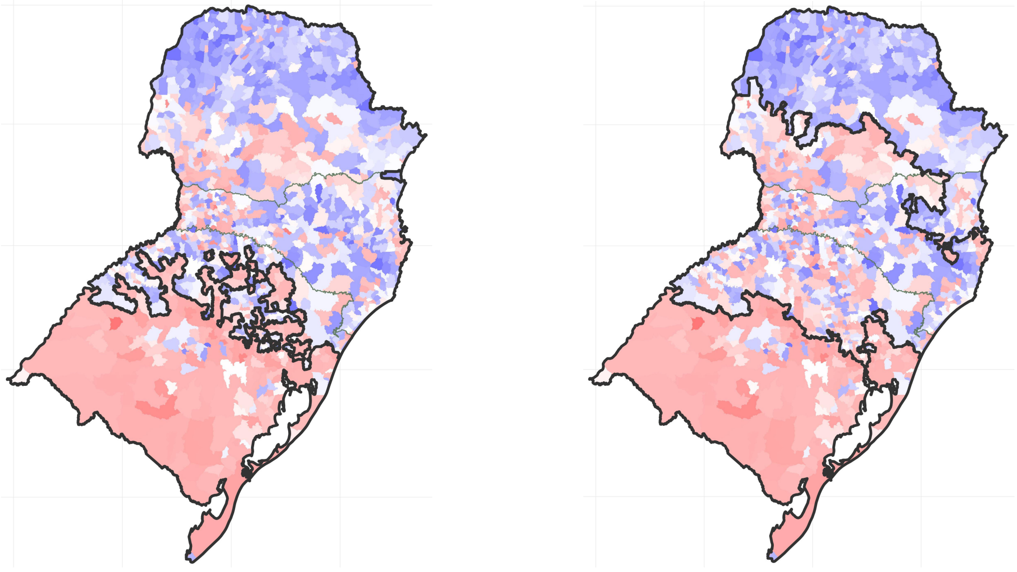
(a) Lung cancer

(b) Bladder cancer

Figure 6.5: Summary of the regionalization (Cancer deaths)

We also show the results of applying two other regionalization techniques to the datasets. In Fig. 6.6 we show the resulting map produced by the *ARiSEL* and *SKATER* techniques in the lung cancer map. The same is shown in Fig. 6.7 for the bladder cancer map. In all these images the methods were issued to compute 4 groups.

For the lung cancer, since the incidence is higher, the rates tend to be more stable than they are with the rarer bladder cancer. It is interesting to note how the *SKATER* method was still able to separate the top from the bottom of the map, where in the *ARISEL*, only the southern boundary was detected, yet with a more jagged line.


(a) ARiSEL

(b) SKATER

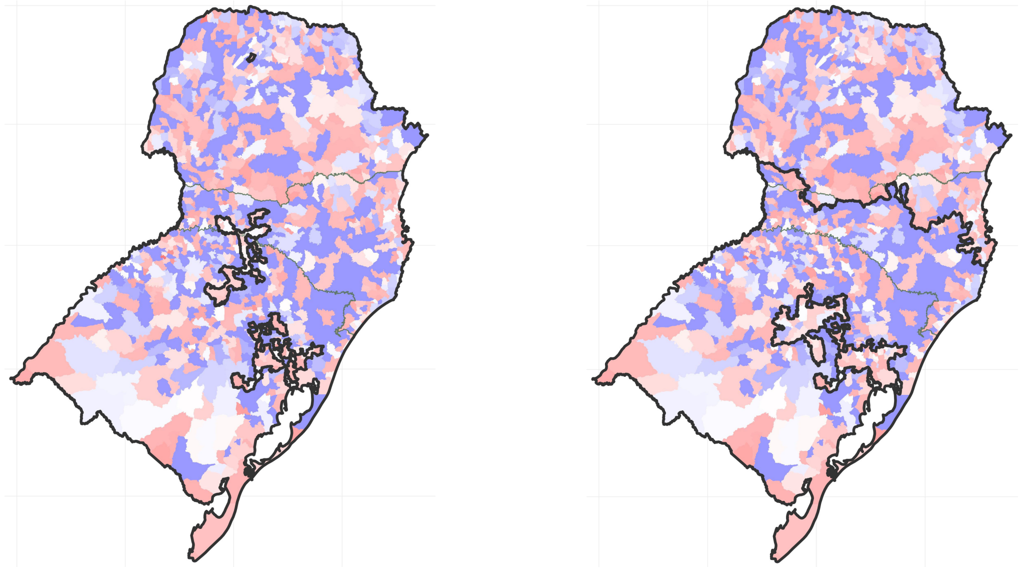
Figure 6.6: Other regionalization methods results for the lung cancer map

For the bladder cancer, however, neither method was able to detect the region in the northeast of the map. They seemed to be more sensitive to local variations in the rate. This is the practical exemplification of what we expected given that these methods don't use a statistical model and are more susceptible to this kind of problem where the population or the incidence rate is lower.

In Figure 6.8 we plot a map with the color of each area representing the average of area of the group to which it belongs throughout the sampled partitions.

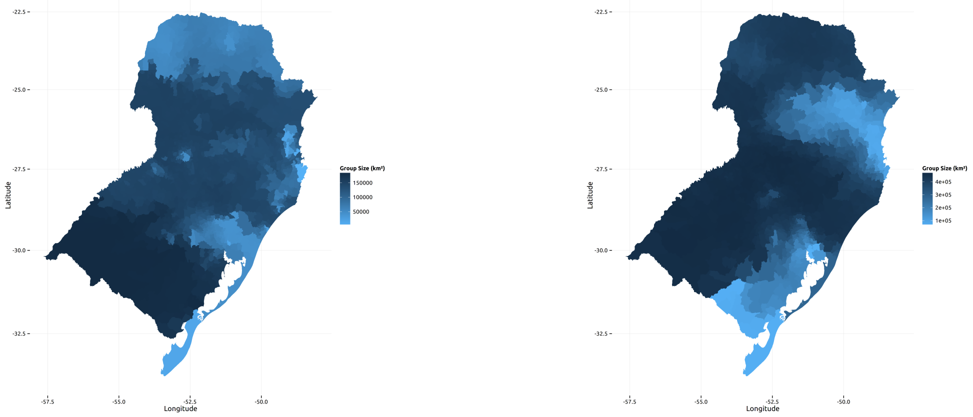
In Fig. 6.9 we show some of the actual partitions sampled in our algorithm for the lung cancer dataset. In Fig. 6.10 we show the same for the bladder cancer dataset.

In Fig. 6.5a we can notice how the SPPM separates the north and south groups as well as the portion on the right in the middle region of the map. Lung cancer is linked to smoking and the rate increases going south, which could be related to the colder climate. Also, it is interesting to note that the other methods either separates



(a) ARiSEL (b) SKATER

Figure 6.7: Other regionalization methods results for the bladder cancer map



(a) Lung cancer (b) Bladder cancer

Figure 6.8: Average size of the clusters for each area

only the south or fail to identify the right portion of the middle region of the map, which is where a few of the larger cities of the region are located, and presents a lower ratio than its western counterpart.

A more visible difference is in Fig. 6.5b, where we can notice how SPPM identified a cluster in the northeast of the map, which is precisely where Curitiba (capital of Parana state) and the most populated cities of the state of Santa Catarina are located. All the other methods failed to find this cluster and instead identified noisy small regions, which is a demonstration of how these methods are sensitive to the variations of the data.

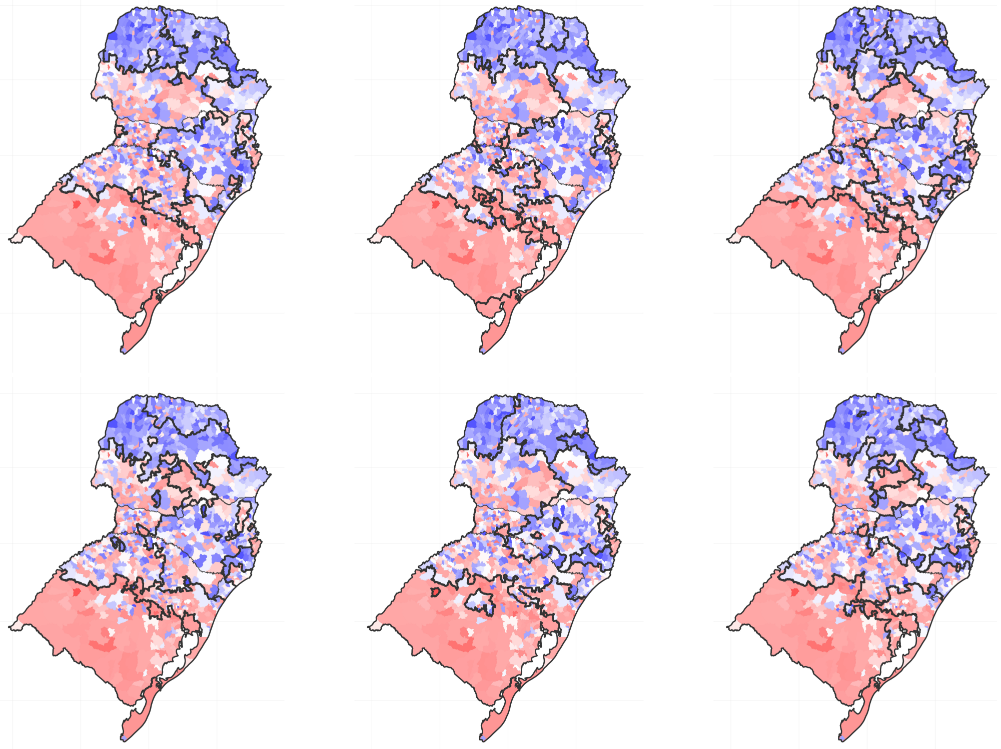


Figure 6.9: Some of the sampled partitions (lung cancer)

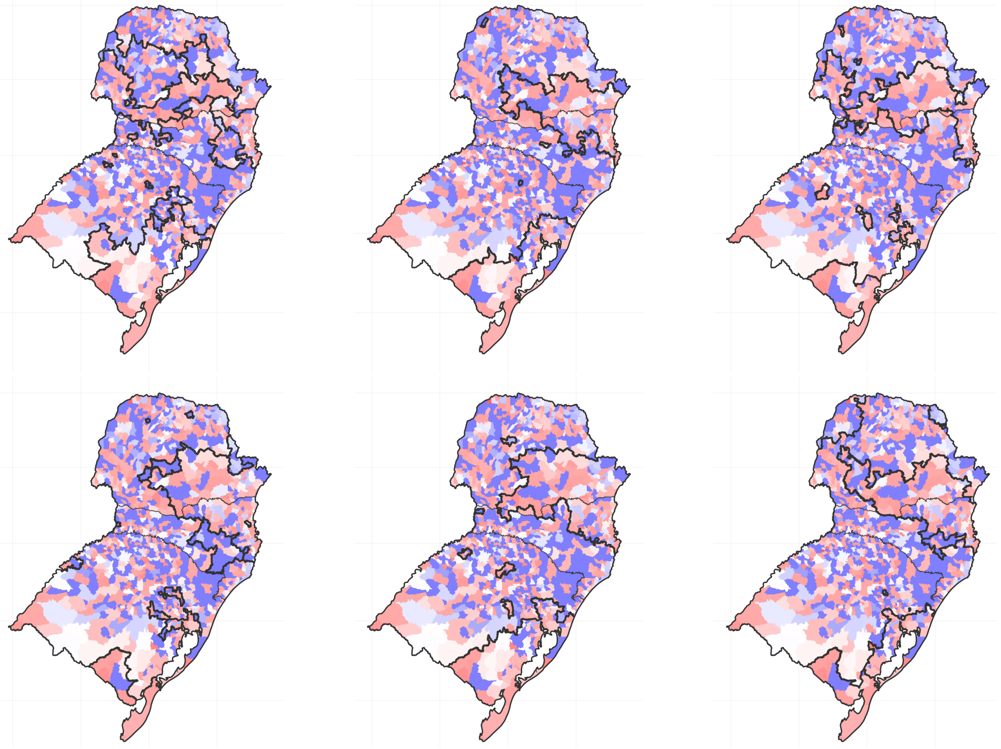


Figure 6.10: Some of the sampled partitions (bladder cancer)

Chapter 7

Conclusion

In this work we dealt with the problem of regionalization, an important type of clustering problem which arises in many areas. We reviewed the main techniques used to tackle this problem and how they fail to handle different characteristics of the problem.

We proposed a representation of the problem in terms of a graph and proposed a new stochastic model based on this representation. Our proposed model builds upon the well known Product Partition Model and we use a spanning tree as a tool to reduce the search space of the partitions of the data. With this, we presented a sampling algorithm for our proposed model, which is flexible enough to be adapted for variations of our model where different probability distributions are assumed for the data. Next, we introduced two specific versions of our model, one assuming a normal distribution for the data and the other assuming a Poisson distribution.

A third model was provided as an example of how a more complex model of the data can be assumed, but which presents also more challenges in terms of the sampling procedure. This promising model did not have as good performance as our product partition model and we did not present any results related to it. It remains as a promising proposal if one could overcome its deficiencies.

Finally, we presented an evaluation of our method both in a simulated study as well as through its application to real datasets. In the simulated study we compared our algorithm to available implementations of traditional algorithms used in the problem and showed how our results were consistently superior, particularly within the Poisson datasets, where a proper stochastic model presents a valuable gain in terms of the quality of the results. Then we applied our technique to perform the regionalizations of municipalities of Brazil based on a socio-economic index and of the municipalities of the south region of Brazil based on cancer mortality data.

We discussed how the results we obtained were suitable for the domain subjects and how, particularly with the Poisson model, by using a stochastic model rather than the raw data to drive the clustering process, we were able to obtain better and more meaningful results.

In conclusion, we proposed a stochastic model for the problem of regionalization that captures much of the prior reasoning one could have for its formation. We introduce the use of spanning trees to provide an effective sampling algorithm. Our model is flexible enough to accommodate different types of data and provided good results.

Bibliography

Aldstadt, J. and Getis, A. (2006). Using amoeba to create a spatial weights matrix and identify spatial clusters. *Geographical Analysis*, 38(4):327--343.

Altman, M. (1998). *Districting principles and democratic representation*. PhD thesis, California Institute of Technology.

Alvanides, S. and Openshaw, S. (1999). Zone design for planning and policy analysis. In Stillwell, J., Geertman, S., and Openshaw, S., editors, *Geographical Information and Planning*, Advances in Spatial Science, pages 299–315. Springer Berlin Heidelberg.

Anderson, C., Lee, D., and Dean, N. (2014). Identifying clusters in bayesian disease mapping. *Biostatistics*, page kxu005.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. (2003). An introduction to mcmc for machine learning. *Machine Learning*, 50(1-2):5–43. ISSN 0885-6125.

Assunção, R. M., Neves, M. C., Câmara, G., and da Costa Freitas, C. (2006). Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. *International Journal of Geographical Information Science*, 20(7):797–811.

Bação, F., Lobo, V., and Painho, M. (2004). Geo-self-organizing map (geo-som) for building and exploring homogeneous regions. In *Geographic Information Science*, pages 22--37. Springer.

Bação, F., Lobo, V., and Painho, M. (2005). Applying genetic algorithms to zone design. *Soft Computing*, 9(5):341--348.

Banerjee, S. and Gelfand, A. E. (2006). Bayesian wombling: Curvilinear gradient assessment under spatial process models. *Journal of the American Statistical Association*, 101(476):1487--1501.

Barry, D. and Hartigan, J. A. (1992). Product partition models for change point problems. *The Annals of Statistics*, 20(1):260--279.

Barry, D. and Hartigan, J. A. (1993). A bayesian analysis for change point problems. *Journal of the American Statistical Association*, 88(421):309--319.

Bernetti, I., Ciampi, C., and Sacchelli, S. (2011). Minimizing carbon footprint of biomass energy supply chain in the province of florence. *Italian Journal of Forest and Mountain Environments*, 66(4). ISSN 20363494.

Besag, J., York, J., and Mollié, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43(1):1--20.

Broder, A. (1989). Generating random spanning trees. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pages 442–447. IEEE.

Browdy, M. H. (1990). Simulated annealing: an improved computer model for political redistricting. *Yale Law & Policy Review*, pages 163--179.

Cattell, R. B. (1943). The description of personality: basic traits resolved into clusters. *The journal of abnormal and social psychology*, 38(4):476.

Denison, D. and Holmes, C. (2001). Bayesian partitioning for estimating disease risk. *Biometrics*, 57(1):143--149.

Duque, J. and Church, R. (2004). A new heuristic model for designing analytical regions. In *North American Meeting of the International Regional Science Association, Seattle*.

Duque, J. C., Anselin, L., and Rey, S. J. (2012). The max-p-regions problem. *Journal of Regional Science*, 52(3):397--419.

Duque, J. C., Dev, B., Betancourt, A., and Franco, J. L. (2011). *ClusterPy: Library of spatially constrained clustering algorithms, Version 0.9.9.* RiSE-group (Research in Spatial Economics). EAFIT University., Colombia.

Duque, J. C., Ramos, R., and Suriñach, J. (2007). Supervised regionalization methods: A survey. *International Regional Science Review*, 30(3):195--220.

Gangnon, R. E. and Clayton, M. K. (2000). Bayesian detection and modeling of spatial disease clustering. *Biometrics*, 56(3):922--935.

Garfinkel, R. S. and Nemhauser, G. L. (1970). Optimal political districting by implicit enumeration techniques. *Management Science*, 16(8):B--495.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):721–741. ISSN 0162-8828.

Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711--732.

Guo, D. (2008). Regionalization with dynamically constrained agglomerative clustering and partitioning (redcap). *International Journal of Geographical Information Science*, 22(7):801--823.

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

Hegarty, A. and Barry, D. (2008). Bayesian disease mapping using product partition models. *Statistics in medicine*, 27(19):3868--3893.

Hess, S. W., Weaver, J., Siegfeldt, H., Whelan, J., and Zitlau, P. (1965). Nonpartisan political redistricting by computer. *Operations Research*, 13(6):998--1006.

Hodges, J. S., Carlin, B. P., and Fan, Q. (2003). On the precision of the conditionally autoregressive prior in spatial models. *Biometrics*, 59(2):317--322.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264--323.

Knorr-Held, L. (2003). Some remarks on gaussian markov random field models for disease mapping. *OXFORD STATISTICAL SCIENCE SERIES*, pages 260--263.

Knorr-Held, L. and Raßer, G. (2000). Bayesian detection of clusters and discontinuities in disease maps. *Biometrics*, 56(1):13--21.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464--1480.

Lavine, M. L. and Hodges, J. S. (2012). On rigorous specification of icar models. *The American Statistician*, 66(1):42--49.

Lu, H. and Carlin, B. P. (2005). Bayesian areal wombling for geographical boundary analysis. *Geographical Analysis*, 37(3):265--285.

Macmillan, W. and Pierce, T. (1994). Optimization modelling in a gis framework: the problem of political redistricting. *Spatial analysis and GIS*, pages 221--246.

Madigan, D. and Raftery, A. E. (1994). Model selection and accounting for model uncertainty in graphical models using occam's window. *Journal of the American Statistical Association*, 89(428):1535--1546.

Martin, D. (1998). Optimizing census geography: the separation of collection and output geographies. *International Journal of Geographical Information Science*, 12(7):673–685. PMID: 12294535.

Martins-Bedé, F. T., Godo, L., Sandri, S., Dutra, L. V., Freitas, C. C., Carvalho, O. S., Guimarães, R. J., and Amaral, R. S. (2009). Classification of schistosomiasis prevalence using fuzzy case-based reasoning. In *Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part I: Bio-Inspired Systems: Computational and Ambient Intelligence*, IWANN '09, pages 1053--1060, Berlin, Heidelberg. Springer-Verlag.

Mehrotra, A., Johnson, E. L., and Nemhauser, G. L. (1998). An optimization based heuristic for political districting. *Management Science*, 44(8):1100--1114.

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.

Møller, J. and Waagepetersen, R. P. (1998). Markov connected component fields. *Advances in Applied Probability*, pages 1--35.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Nagel, S. S. (1965). Simplified bipartisan computer redistricting. *Stanford Law Review*, pages 863--899.

Openshaw, S. (1973). A regionalisation program for large data sets. *Computer Applications*, 3(4):136--147.

Openshaw, S. (1977a). A geographical solution to scale and aggregation problems in region-building, partitioning and spatial modelling. *Transactions of the institute of british geographers*, pages 459--472.

Openshaw, S. (1977b). Optimal zoning systems for spatial interaction models. *Environment and Planning A*, 9(2):169--184.

Openshaw, S. (1978). An optimal zoning approach to the study of spatially aggregated data. In *Spatial representation and spatial interaction*, pages 95--113. Springer.

Openshaw, S. (1984a). Ecological fallacies and the analysis of areal census data. *Environment and Planning A*, 16(1):17--31.

Openshaw, S. (1984b). The modifiable areal unit problem. Geo Abstracts University of East Anglia.

Openshaw, S. and Rao, L. (1995). Algorithms for reengineering 1991 census geography. *Environment and planning A*, 27(3):425--446.

Openshaw, S. and Wymer, C. (1995). Classifying and regionalizing census data. *Census Users Handbook. GeoInformation International, Cambrige, UK*, pages 239--268.

Reis, I. A., Câmara, G., Assunção, R., and Monteiro, A. M. V. (2007). Data-aware clustering for geosensor networks data collection. In *Anais XIII Simpósio Brasileiro de Sensoriamento Remoto*, pages 6059--6066.

Richardson, S. and Green, P. J. (1997). On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59(4):731--792.

Ricketts, T. C. (1997). *Using geographic methods to understand health issues*. Agency for Health Care Policy and Research, Dept. of Health and Human Services, US Public Health Service.

Robinson, W. (1950). Ecological correlations and the behavior of individuals. *American Sociological Review*, 15(3).

Rossiter, D. and Johnston, R. J. (1981). Program group: the identification of all possible solutions to a constituency-delimitation problem. *Environment and Planning*, 13:231--8.

Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications*. CRC Press.

Taylor, P. J. (1973). Some implications of the spatial organization of elections. *Transactions of the Institute of British Geographers*, pages 121--136.

Tryon, R. C. (1939). *Cluster analysis: correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality*. Edwards brothers, Incorporated, lithoprinters and publishers.

Vickrey, W. (1961). On the prevention of gerrymandering. *Political Science Quarterly*, pages 105--110.

Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577--584.

Wakefield, J. and Kim, A. (2013). A bayesian model for cluster detection. *Biostatistics*, 14(4):752--765.

Weaver, J. B. and Hess, S. W. (1963). A procedure for nonpartisan districting: development of computer techniques. *The Yale Law Journal*, 73(2):288--308.

Wilson, D. B. (1996). Generating random spanning trees more quickly than the cover time. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 296--303, New York, NY, USA. ACM.

Wise, S., Haining, R., and Ma, J. (1997). Regionalisation tools for the exploratory spatial analysis of health data. In Fischer, M. and Getis, A., editors, *Recent Developments in Spatial Analysis*, Advances in Spatial Science, pages 83–100. Springer Berlin Heidelberg.

Zoltners, A. A. (1979). A unified approach to sales territory alignment. *Sales Management: New Developments from Behavioral and Decision Model Research*, pages 360--376.

Zubin, J. (1938). A technique for measuring like-mindedness. *The Journal of Abnormal and Social Psychology*, 33(4):508.