Nonparametric clustering for spatio-temporal data

Ashwini Kolumam Venkatasubramaniam

A thesis submitted to the

University of Glasgow

for the degree of

Doctor of Philosophy

School of Mathematics and Statistics

January 2019

Abstract

Clustering algorithms attempt the identification of distinct subgroups within heterogeneous data and are commonly utilised as an exploratory tool. The definition of a cluster is dependent on the relevant dataset and associated constraints; clustering methods seek to determine homogeneous subgroups that each correspond to a distinct set of characteristics. This thesis focusses on the development of spatial clustering algorithms and the methods are motivated by the complexities posed by spatio-temporal data. The examples in this thesis primarily come from spatial structures described in the context of traffic modelling and are based on occupancy observations recorded over time for an urban road network. Levels of occupancy indicate the extent of traffic congestion and the goal is to identify distinct regions of traffic congestion in the urban road network.

Spatial clustering for spatio-temporal data is an increasingly important research problem and the challenges posed by such research problems often demand the development of bespoke clustering methods. Many existing clustering algorithms, with a focus on accommodating the underlying spatial structure, do not generate clusters that adequately represent differences in the temporal pattern across the network. This thesis is primarily concerned with developing nonparametric clustering algorithms that seek to identify spatially contiguous clusters and retain underlying temporal patterns. Broadly, this thesis introduces two clustering algorithms that are capable of accommodating spatial and temporal dependencies that are inherent to the dataset. The first is a functional distributional clustering algorithm that is implemented within an agglomerative hierarchical clustering framework as a two-stage process. The method is based on a measure of distance that utilises estimated

cumulative distribution functions over the data and this unique distance is both functional and distributional. This notion of distance utilises the differences in densities to identify distinct clusters in the graph, rather than raw recorded observations.

However, distinct characteristics may not necessarily be identified and distinguishable by a densities-based distance measure, as defined within the agglomerative hierarchical clustering framework. In this thesis, we also introduce a formal Bayesian clustering approach that enables the researcher to determine spatially contiguous clusters in a data-driven manner. This framework varies from the set of assumptions introduced by the functional distributional clustering algorithm. This flexible Bayesian model employs a binary dependent Chinese restaurant process (binDCRP) to place a prior over the geographical constraints posed by a graph-based network. The binDCRP is a special case of the distance dependent Chinese restaurant process that was first introduced by Blei and Frazier (2011); the binDCRP is modified to account for data that poses spatial constraints. The binDCRP seeks to cluster data such that adjacent or neighbouring regions in a spatial structure are more likely to belong to the same cluster. The binDCRP introduces a large number of singletons within the spatial structure and we modify the binDCRP to enable the researcher to restrict the number of clusters in the graph. It is also reasonable to assume that individual junctions within a cluster are spatially correlated to adjacent junctions, due to the nature of traffic and the spread of congestion. In order to fully account for spatial correlation within a cluster structure, the model utilises a type of the conditional auto-regressive (CAR) model. The model also accounts for temporal dependencies using a first order auto-regressive (AR-1) model. In this mean-based flexible Bayesian model, the data is assumed to follow a Gaussian distribution and we utilise Kronecker product identites within the definition of the spatio-temporal precision matrix to improve the computational efficiency. The model utilises a Metropolis within Gibbs sampler to fully explore all possible partition structures within the network and infer the relevant parameters of the spatio-temporal precision matrix. The flexible Bayesian method is also applicable to map-based spatial structures and we describe the model in this context as well.

The developed Bayesian model is applied to a simulated spatio-temporal dataset that is composed of three distinct known clusters. The differences in the clusters are reflected by distinct mean values over time associated with spatial regions. The nature of this mean-based comparison differs from the functional distributional clustering approach that seeks to identify differences across the distribution. We demonstrate the ability of the Bayesian model to restrict the number of clusters using a simulated data structure with distinctly defined clusters. The sampler is also able to explore potential cluster structures in an efficient manner and this is demonstrated using a simulated spatio-temporal data structure. The performance of this model is illustrated by an application to a dataset over an urban road network, that presents traffic as a process varying continuously across space and time. We also apply this model to an areal unit dataset composed of property prices over a period of time for the Avon county in England.

Acknowledgements

I would like to thank my PhD supervisors, Ludger Evers and Konstantinos Ampountolas, for the support and valuable advice that they have provided me through my time as a student at the University of Glasgow. I look back now and feel fortunate for the four years I spent here as a student. I am also grateful to Piyushimita (Vonu) Thakuriah for her advice and continued interest in my work.

I appreciate being funded by the Lord Kelvin Adam Smith scholarship that provided adequate financial support so that I could focus on working towards my PhD and learning new research techniques. I would also like to thank all the people I have met at the University of Glasgow, both in School of Mathematics and Statistics and in the Urban Big Data Centre.

I will always be grateful for the advice, encouragement and mentorship from Julian Wolfson at the University of Minnesota, Twin Cities. I first had the chance to do statistical research as a Master's student; he supervised my Master's thesis and played a significant role in convincing me to apply and work towards a PhD.

Lastly, I would like to thank my wonderful parents Girija and Koluman Venkatasubramaniam, my sister Arundhathi, my brother-in-law Sathya, and all my friends who have been cheering me on, even from many thousands of miles away.

Declaration

I, **Ashwini Venkatasubramaniam**, declare that this thesis titled, 'Nonparametric spatial clustering for spatio-temporal data' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at the University of Glasgow.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Contents

List of Algorithms

List of Figures

List of Tables

Chapter 1

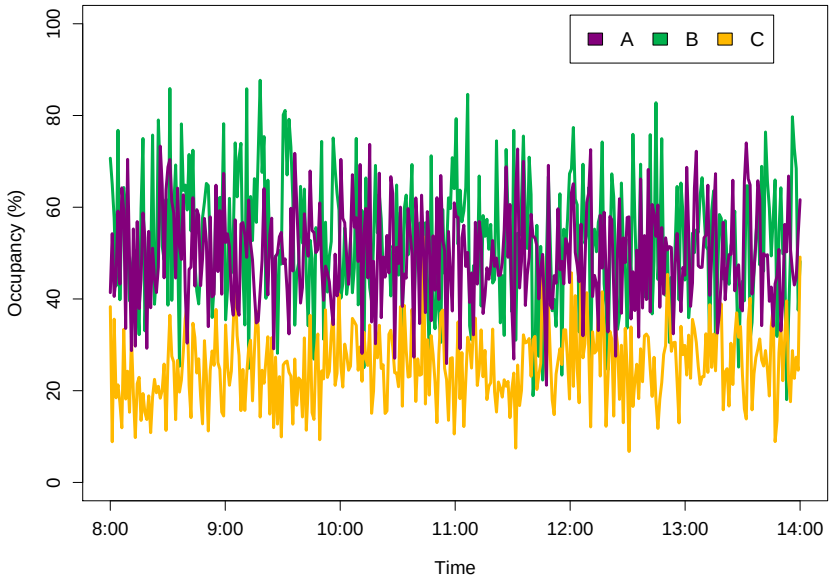
Introduction

1.1 Overview

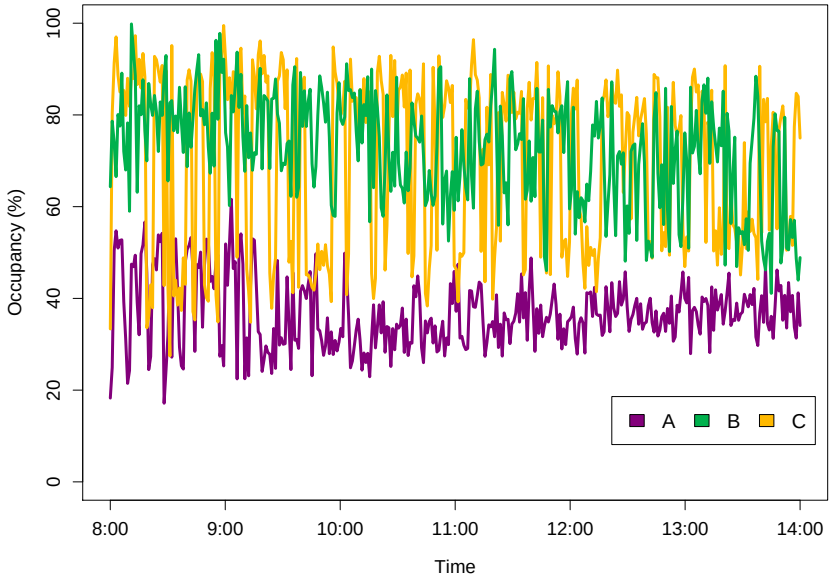
Clustering is an unsupervised learning method that seeks to identify homogeneous clusters in heterogeneous datasets. This thesis focusses on developing clustering algorithms that are able to identify spatially contiguous clusters by accommodating spatial, temporal and network dependencies in spatio-temporal data. The developed clustering methodologies are primarily motivated by examples from traffic modelling for an urban road network. More specifically, the examples focus on the spread of congestion across the network and highlight the algorithm's ability to identify regions that exhibit similar traffic congestion patterns. Differences in the spread of traffic congestion are indicated by the varying levels of occupancies across the urban road network and occupancy is the percentage of time that a location on the network is occupied by vehicles. Levels of traffic occupancy across an urban network vary over time and the development of appropriate clustering algorithms that adequately incorporate spatial and temporal dependencies enable the study of traffic congestion patterns. In general, spatio-temporal datasets are associated with observations recorded over time for vertices arranged as in a grid-style graph network; we focus on spatial structures where each vertex has a limited number of adjacent vertices. In this thesis, we demonstrate applications to observations recorded over time for junctions in an urban road network and also for observations over time associated with areal unit data.

The methods developed in this thesis seek to explore various scenarios of temporal patterns associated with a spatial structure. Temporal patterns at spatial locations can be represented by summary statistics, the underlying multi-modal distributions or can be adapted to appropriate transformations. Figure 1.1 displays temporal patterns that correspond to three distinct clusters (as displayed in Figure 1.2) for two different scenarios. For example, in Figure 1.1a, the observations over time have common variances but differ by the mean values. This differs from Figure 1.1b, where the observations over time for each cluster (identifiable by colour) vary by defined mean values and variance. The ability to adequately accommodate the underlying patterns improves the quality of clustering output and leads to meaningful clusters. For example, an algorithm that averages over a temporal pattern (for the dataset displayed in Figure 1.1b) would generate misleading clusters that do not represent differences in distribution. Spatio-temporal datasets are composed of multiple dependencies, pose multiple challenges to the development of spatial clustering methods and benefit from a variety of techniques to adequately accommodate the underlying complexities. In addition, the number of clusters need not necessarily be known and excessive reliance on personal input may lead to a biased and inaccurate selection. A clustering algorithm that is able to determine the number of clusters in a data-driven manner removes the need for this preliminary knowledge.

As an example, we utilise an urban road network in Downtown San Francisco. This network is composed of 158 junctions and road segments between the junctions, where observations are recorded over time for each junction. In Figure 1.2a, the network of interest is highlighted within the Downtown San Francisco area using a brown border. A darker line within the network highlights the Market Street and determines a clear division between the two regions within the network of interest. The corresponding graph network is described in Figure 1.2b. These 158 junctions are divided to form three clusters (A, B, C), where the clusters are represented by the different colours (purple, green, yellow). Cluster A is formed over the area below Market Street and represents a region with lower simulated occupancy levels. In comparison, clusters B and C over the Financial district area represent a concentration of higher vehicular occupancy levels. Cluster B is formed over both the top right and bottom



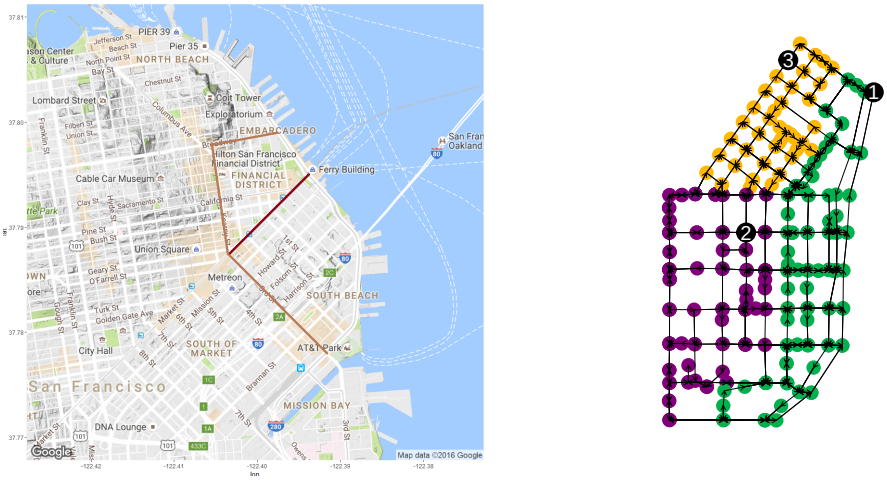
(a) Differences in the mean



(b) Differences in the mean and variance

Figure 1.1: Example scenario: temporal patterns for three different clusters (A, B and C).

right region of the network and includes regions above the Market Street (Embarcadero) and below the Market Street (South Beath, Rincon Hill). These clusters are differentiated by differences in the mean and the variance.



(a) Downtown San Francisco (b) Relevant graph network

Figure 1.2: Urban traffic network in Downtown San Francisco

In Figure 1.2, three junctions labelled as 1, 2 and 3 are highlighted in the network structure. Each of these junctions are located in three different cluster regions.

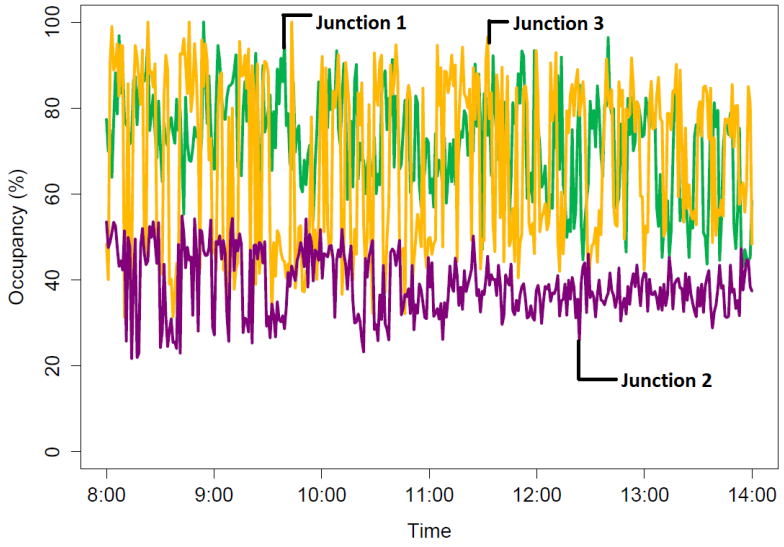


Figure 1.3: Occupancy observations over time for junctions 1, 2 and 3 (as highlighted in Figure 1.2)

More specifically, junction 1 is in cluster B, junction 2 is in cluster A and junction 3 is in cluster C. The corresponding occupancy observations for these three junctions are presented in Figure 1.3. These observations in Figure 1.3 represent temporal patterns that are similar to Figure 1.1b and highlight the differences between distinct clusters.

We present nonparametric clustering algorithms in this thesis that seek to identify spatially contiguous clusters using spatio-temporal data. More specifically, the clustering algorithms are introduced to determine spatially contiguous clusters that represent distinct temporal characteristics. We first introduce the *functional distributional clustering* algorithm developed within an agglomerative hierarchical clustering framework. The measure of distance is defined using cumulative distribution functions (CDFs), where CDFs are estimated using nonparametric kernel density estimators. To the best of our knowledge, a clustering approach that is *both* functional and distributional has not been previously introduced. This notion of distance is able to retain differences in distribution over time for each spatial location using densities and this method is particularly suitable for datasets that record consecutive 'jumps' for observations over time. We also introduce a relevant three-dimensional plot that is able to visualise these differences in densities over time for each cluster. However, this two-stage ad hoc clustering approach is not implemented within a formal statistical framework. A second method introduced in this thesis is a formal Bayesian approach, developed as a nonparametric spatial clustering method, to determine spatially contiguous clusters in a data-driven manner. This model seeks to accommodate the underlying spatial, temporal and network dependencies in a more comprehensive manner. This flexible Bayesian model first places a *binary dependent Chinese restaurant process* (binDCRP) over the graph as a prior; the binDCRP seeks to incorporate the geographical constraints imposed by the nature of the network. We also introduce a modification to the binDCRP that allows the formation of new clusters to be controlled and enables the number of clusters to be restricted. In this binDCRP-based model, the recorded occupancy observations over the network are assumed to follow a Gaussian distribution. In an urban road network, it is reasonable to assume that traffic occupancies in junctions are correlated to neighbouring junctions. To fully incorporate these spatial dependencies within the suggested clusters, a type of *conditional*

auto-regressive (CAR) model is utilised to accommodate spatial dependencies and a first order auto-regressive (AR-1) model for accommodating temporal dependencies. A relevant spatio-temporal precision matrix is defined and the availability of a unique observation for every space and time combination enables the utilisation of Kronecker product identities. These relevant identities are implemented within the data model and improve the computational efficiency of the model. The Metropolis within Gibbs sampler is utilised to explore all potential cluster structures within the network and also infers relevant parameters. The posterior suggests a potential partition structure composed of spatially contiguous clusters. In the Bayesian approach, the observations are assumed to follow a Gaussian distribution; this is unlike the functional distributional algorithm framework that is developed to model multi-modal distributions. The algorithms are motivated by different variations in temporal data and are developed to meet challenges posed by the complex nature of spatio-temporal datasets.

1.2 Thesis outline

The rest of the thesis is organised as follows. Chapter 2 provides a literature review of commonly used clustering methods and their relevance to spatial and temporal datasets. The literature review broadly describes partitional clustering, hierarchical clustering and relevant clustering evaluation methods. This review also highlights Bayesian inference, mixture models and nonparametric Bayesian approaches and their associated applications to clustering.

Chapter 3 introduces a spatial clustering algorithm for spatio-temporal data motivated by an example in traffic modelling. This two-stage clustering method labelled as the 'functional distributional clustering' is implemented within an agglomerative hierarchical clustering framework and utilises a unique measure of distance that is both functional and distributional. This method seeks to identify spatially contiguous clusters that are distinguished by differences in density functions and are able to account for variations in the underlying distribution (includes both mean and variances).

Chapter 4 describes the Chinese restaurant process (CRP), an alternative view of the Chinese restaurant process, the distance dependent Chinese restaurant process (ddCRP) and introduces a special case of the ddCRP for spatial data called the binary dependent Chinese restaurant process (binDCRP).

Chapter 5 introduces a flexible Bayesian model as a holistic approach to clustering spatio-temporal data. This mean-based nonparametric clustering algorithm determines the number of spatially contiguous clusters from the underlying data. The Bayesian model places a binDCRP prior over the spatial structure and assumes that the observations recorded over time follow a Gaussian distribution. We introduce a relevant Metropolis within Gibbs sampler to explore all possible clustering structures within the network and infer the parameters within the spatio-temporal precision matrix.

Chapter 6 demonstrates the performance of the binDCRP-based clustering using an application to simulated data. The simulated data is utilised to compare the cluster structures generated by the binDCRP based clustering approach and demonstrate the ability to restrict the number of clusters. The application to real data is illustrated using multiple examples, including occupancy observations recorded for an urban road network and for areal unit data associated with property prices that are recorded over a period of time.

Conclusions and future work are discussed in Chapter 7. Algorithms, figures and tables, introduced through the thesis, are presented as lists, before the first Chapter.

Chapter 2

Review of clustering methods

2.1 Introduction

Clustering is a fundamentally important unsupervised learning problem (i.e., the group labels are unknown) and clustering approaches typically seek to detect clusters (or groups) within a heterogeneous dataset. Clustering is often utilised as a tool for exploratory analysis and serves as a means to better understand recorded datasets. An identified cluster is typically associated with a set of distinct characteristics and common applications of clustering include image segmentation (Orbanz and Buhmann, 2008, Zeng et al., 2014), clustering related genes from gene expression data (Lu et al., 2018, McDowell et al., 2018), social networks (Levine and Kurzban, 2006, van Dam and Van De Velden, 2015), and disease modelling (Anderson et al., 2014, Wakefield and Kim, 2013). The definition of a meaningful cluster is dependent on the context of the research question and the underlying data. The need for clustering approaches to accommodate a combination of varied constraints in complex datasets demands the development of bespoke clustering algorithms. For a general description and more detailed comparison of existing clustering algorithms, see Alpaydin (2009), Friedman et al. (2001). This chapter summarises several relevant and commonly used clustering methods. Section 2.2 briefly discusses deterministic clustering approaches (hierarchical, non-hierarchical, spectral), its implementation and evaluation of the clustering output. Section 2.3.1 summarises Bayesian inference and discusses finite mixture models and non-parametric Bayesian methods.

2.2 Deterministic clustering

This section reviews both hierarchical and non-hierarchical clustering algorithms including K-means, agglomerative hierarchical clustering and spectral clustering.

2.2.1 Hierarchical clustering

Hierarchical clustering (Johnson, 1967) is a widely used unsupervised learning method and produces not just one clustering but a family of clusterings. This method uses a series of successive mergers to cluster a set of objects. Hierarchical clustering defines a measure of dissimilarity between groups of data objects to generate a hierarchy of clusters. For a given choice of dissimilarity measure (e.g., Euclidean) between objects, the dissimilarity matrix D is defined as

$$D = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1N} \\ d_{21} & d_{22} & \dots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \dots & d_{NN} \end{pmatrix}$$

where $d_{ij} = d(x_i, x_j)$ is the distance between objects x_i and x_j. The dissimilarity defined in the matrix D between clusters (that are composed of such objects) is quantified in multiple ways. This includes single linkage, complete linkage and group average, among other choices. Formally, let two clusters be denoted as G and H and the distance between the two clusters are as below. The single linkage $d_{SL}(G, H)$ defines the dissimilarity of the closest pair of objects in clusters G and H such that

$$d_{SL}(G, H) = \min_{i \in G, j \in H} d_{ij}.$$

The complete linkage $d_{CL}(G, H)$ defines the dissimilarity of the furthest pair of objects such that

$$d_{CL}(G, H) = \max_{i \in G, j \in H} d_{ij}.$$

Broadly, hierarchical clustering can be divided into agglomerative and divisive methods. Single linkage and complete linkage are one among several agglomerative hierarchical clustering methods. In an *agglomerative* hierarchical clustering method, each object is initially assigned to its own cluster and iteratively merged at each level to the two closest clusters.

This process continues until every object belongs to a single cluster and the algorithm generates a sequence of groupings. Similarly, in a *divisive* method all objects initially belongs to a single cluster and clusters are formed by dividing the data at each iteration. The process continues until each leaf has a single object such that clusters correspond to a set of singletons. (The agglomerative hierarchical clustering algorithm adapted to accommodate spatial and temporal constraints is described in detail in Chapter 3.)

The choice of dissimilarity metric and linkage within the hierarchical framework influences the clustering output and relies on the nature of application. Both hierarchical and non-hierarchical clustering approaches are commonly based on a notion of distance or dissimilarity measure that is defined between a set of objects. A class of distance measure, called the Euclidean distance, is based on the locations of the objects. Non-Euclidean distance measures are not based on the location of objects and the notion of average between the objects need not necessarily be defined. Instead, this class of distance is based on the properties of the objects. Examples include the Jaccard distance, Cosine distance and the Edit distance. A more detailed comparison of distance measures can be found in Ackermann et al. (2010), Hassan et al. (2014), Jaskowiak et al. (2014), Shirkhorshidi et al. (2015). In this thesis, we focus on utilising distance measures that summarise the underlying data associated with an object. A hierarchical clustering result can be visualised as a *dendrogram*, where inner nodes represent nested clusters with varying number of objects that belong to each cluster. In other words, a dendrogram organises clusters in a hierarchical manner to provide a useful summary of the data. The hierarchical clustering algorithm provides minimal guidance towards choosing the optimal number of clusters or the level at which to cut the dendrogram. Different decisions about dissimilarities and choices about the cluster structure of interest can often lead to vastly different dendrograms. Figure 2.1 displays a dendrogram that represents a hierarchy of clusters and different clusters can be identified depending on the decisions about the desired partition structure.

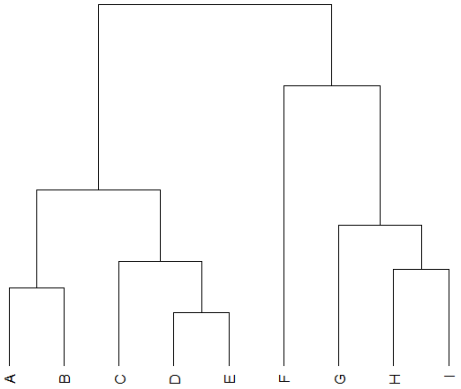


Figure 2.1: Dendrogram that visually represents a hierarchical organisation of clusters

Hierarchical clustering algorithms are commonly adapted to spatial datasets by defining a measure of distance that accommodates the geographical constraints. Spatially adjusted hierarchical clustering methods are motivated by many applications (Dumont et al., 2018, Zhang et al., 2017, Zhu and Guo, 2014) and various definitions for the distance measure are defined within the method.

2.2.2 Non-hierarchical clustering

Non-hierarchical (or flat) clustering approaches partition a given set of objects into distinct groups based on a defined distance or dissimilarity measure. K-means (Hartigan and Wong, 1979) is a commonly used partitional clustering method that seeks to cluster a dataset of N unlabelled objects into K user-specified clusters. The K-means method requires K number of clusters to be specified and an appropriate distance measure between objects to be defined. The algorithm is dependent on an initial choice of K and the N objects are partitioned into K distinct clusters. In each iteration, an object is assigned to a cluster that has the closest defined centroid, over all clusters. The cluster centroid is then updated to include the new object (if assigned to a new cluster). This is repeated until all objects simply remain in the same cluster structure. This results in a set of clusters that are well-separated. Figure 2.2 displays an application of K-means to a given set of data objects, when an initial choice of two clusters is made. In Figure 2.2, the objects are initially assigned to two specified clusters and successive iterations then assign the objects to the eventual

clustering output highlighted in blue and red.

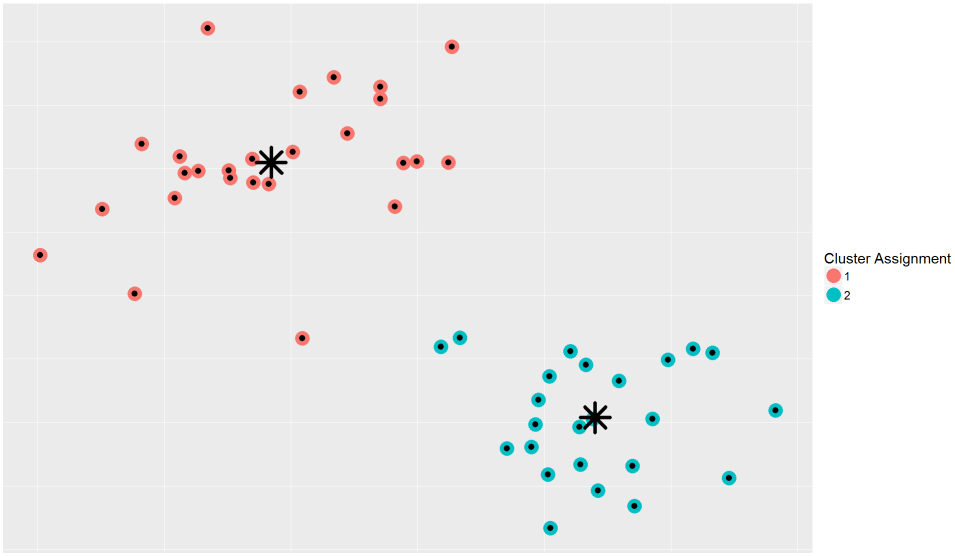


Figure 2.2: K-means clustering for initial choice of two clusters.

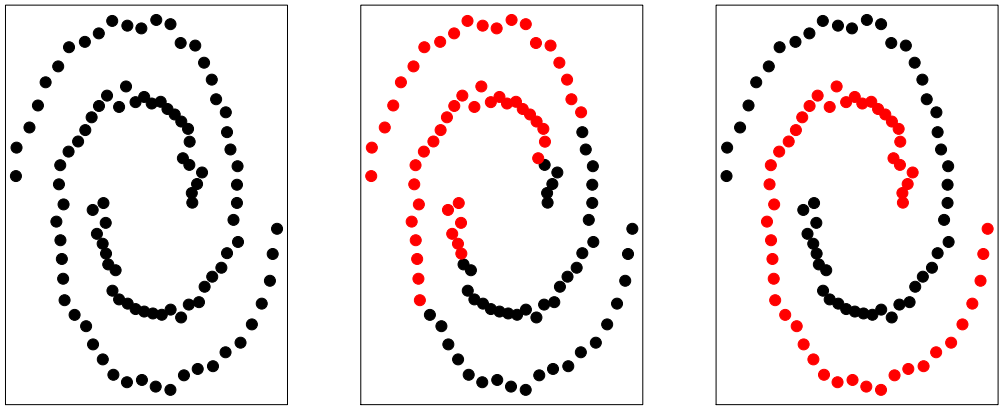
Spatially adjusted K-means approaches have been developed in combination with other methods to incorporate the geographical constraints of the underlying data (Ilea and Whelan, 2006, Mignotte, 2011, Xie et al., 2015); such approaches have often been motivated by research questions in image segmentation. However, the reliance of the algorithm on the initial choice of number of clusters and the dependence on the ability to compute a selected summary value reduces the flexibility of the algorithm. In addition, K-means struggles to detect non-spherical clusters (see spectral clustering in Section 2.2.3) and can lead to a misleading clustering result. Both K-means and hierarchical clustering methods are commonly combined with other algorithms to improve the quality of the clustering output.

2.2.3 Dimensionality reduction

Spectral clustering treats the process of clustering as a graph partitioning problem without introducing assumptions about the form of the clusters. In spectral clustering, each element in the similarity matrix S_{ij} represents how similar data object i is to data object j. The similarity matrix is then transformed to an eigenvector domain that allows the eigenvectors to provide an ability to identify the most significant features within a dataset of objects.

The clusters are identified and labelled using a clustering algorithm such as K-means. The objects in the dataset are mapped to a low-dimensional space such that they are separated; this enables them to then be easily clustered. For a more detailed explanation of spectral clustering, see Ng et al. (2002), Von Luxburg (2007).

Spectral clustering has numerous applications (Bach and Jordan, 2006, Higham et al., 2007) and seeks to cluster data with convex boundaries that may not necessarily be identified by other methods. Spectral clustering methods are more flexible and capture many geometric shapes. This is a class of methods that is based on eigendecompositions of the dissimilarity matrices. They have shown superior empirical performance as compared to many competing algorithms such as K-means (For a detailed comparison of spectral clustering methods to K-means, see Verma and Meila (2003)). In Figure 2.3, data objects that are arranged as a spiral are not recognised accurately by K-means but are distinguished by the spectral clustering method.



(a) Two spirals (b) K-means (c) Spectral clustering

Figure 2.3: K-means fails for datasets that are not linearly separable and is unable to identify two spirals. As demonstrated, spectral clustering works well in this case.

2.2.4 Clustering evaluation

Evaluation of clustering results is a complex process that can be performed internally as well as externally. Differences in the chosen number of clusters can be due to a variety of reasons within the clustering process. A well-performing clustering algorithm results in clusters with good inter-cluster separation and intra-cluster homogeneity. Many evaluation methods can be used to assess the quality of clustering algorithms based on internal criterion such as the Davies-Bouldin index (Davies and Bouldin, 1979), Dunn index (Dunn, 1973) and Silhouette index (Rousseeuw, 1987). These measures are also described in Chapter 3. Internal indices can be used to choose the best clustering algorithm as well as the optimal number of clusters. Since availability of information about datasets is often limited, internal validation methods are more useful. External indices are based on information about the data, e.g., the optimal number of clusters are useful for selecting the best clustering method. External evaluation measures compare the clustering output to a given true cluster structure. For example, Rand index (Rand, 1971), Adjusted Rand Index (Hubert and Arabie, 1985), Jaccard index (Downton, 1980), etc. For a comprehensive review of clustering evaluation approaches, see Ansari et al. (2015), Kovács et al. (2005), Milligan and Cooper (1987), Petrovic (2006).

2.3 Bayesian clustering

Hierarchical and non-hierarchical clustering approaches briefly described in Section 2.2 are better suited for clusters that are well separated. However, they do not provide an assessment of clustering uncertainties. Model-based clustering methods provide an approach that utilises formal principles of statistical inference and clustering analysis in this framework is based on a probability model. *Mixture models* are commonly used for the purpose of clustering, are able to accommodate overlaps associated with clusters and these models are suitable for data that cannot necessarily be represented by a simple distribution. Instead, a population is treated as composed of several distinct sub-populations. In this section, we briefly describe finite mixture models with a focus on its implementation in a Bayesian

framework and also discuss nonparameteric Bayesian models. In Section 2.3.1, we first provide a brief introduction to Bayesian inference. A more comprehensive review of Bayesian inference can be found in Gelman et al. (1995), Robert et al. (2010), Rogers and Girolami (2016).

2.3.1 Bayesian Inference

The primary goal of statistical inference is to learn about the population parameter θ. In both Bayesian and frequentist approaches to inference, we seek to make inferences about a population parameter θ and a likelihood $p(x \mid \theta)$ over the data x. In the frequentist approach, the population parameter θ can be learned using method of moments or the maximum likelihood and associated with point estimators, their variances and confidence intervals. However, unlike the frequentist approach, the Bayesian framework treats the population parameter θ as a random variable and this enables a probability distribution to be specified for the parameter θ. Bayesian inference is concerned with the calculation of the posterior distribution of unknown quantities, given both data and the prior opinions on those parameters. Accordingly, the Bayesian framework paints a more comprehensive picture of the underlying uncertainty.

The prior represents our beliefs of the distribution of θ before observing information about the data x. $p(\theta)$, the prior, is the probability of the population parameter θ. The prior varies according to the knowledge of the relevant unknown parameter. The *likelihood*, as a function of parameters, is the probability of x conditioned on θ and determines the probability of observing the data x under different values of the parameter θ. The *posterior distribution* represents the beliefs of the distribution of the parameter θ, after observing the data. The posterior distribution for θ is calculated as follows using the Bayes theorem:

$$p(\theta \mid x) = \frac{p(\theta)p(x \mid \theta)}{p(x)} \propto p(\theta)p(x \mid \theta). \tag{2.1}$$

The above rule can be re-written as:

$$\text{Posterior} \propto \text{Prior} \times \text{Likelihood}$$

It is of interest to utilise a prior that has the same functional form as the likelihood, so that the posterior distribution belongs to the same family of distributions as the prior distribution. Such priors are referred to as *conjugate* priors and enable the posterior to be evaluated in a simple manner. However, in many situations, the posterior is intractable and the posterior is approximated using sampling methods; intractable calculations are replaced by relevant simulated approximations. The most commonly used family of sampling methods is *Markov Chain Monte Carlo* (MCMC) simulations (for a complete review, see Gamerman and Lopes (2006)). MCMC methods were introduced by Gelfand and Smith (1990), Tanner and Wong (1987) as an alternative to numerical integration and can be traced back to Metropolis et al. (1953) and Hastings (1970).

2.3.1.1 Simulation

The posterior is generated by prior assumptions and the likelihood function defined over the data. However, its exact computation is often dependent on cumbersome integrations. Instead, one has to adopt other simulation-based strategies to obtain the posterior. *Monte Carlo* methods are simulation-based approximation techniques that are motivated by the law of large numbers and refers to any method that utilises random sampling. However, generating independent and identically distributed samples is not necessarily feasible. *Markov chain Monte Carlo* (MCMC) methods are capable of drawing samples from the posterior. These draws are dependent and form a Markov chain. More formally, a *Markov chain* is said to be a sequence of random variables for which:

$$p\left(\theta^{(t+1)} \mid \theta^{(0)}, \theta^{(1)}, \ldots, \theta^{(t)}\right) = p\left(\theta^{(t+1)} \mid \theta^{(t)}\right)$$

This is such that that the distribution of $\theta^{(t+1)}$ depends on only the previous draw $\theta^{(t)}$ and is independent of all other draws θ^0, θ^1 $\ldots \theta^{(t-1)}$. The probability of each current draw is conditionally dependent on the previous draw.

2.3.1.2 Metropolis-Hastings

The *Metropolis-Hastings* algorithm (Hastings, 1970) constructs a Markov chain such that for a given state $\theta^{(t)}$, the candidate state $\theta^{(t+1)}$ is drawn from a proposal distribution

$q(\theta^{(t+1)} \mid \theta^{(t)})$. This candidate state $\theta^{(t+1)}$ is accepted with probability r such that

$$r = \min\left\{1, \frac{p(\theta^{(t+1)})q\left(\theta^{(t)} \mid \theta^{(t+1)}\right)}{p(\theta^{(t)})q\left(\theta^{(t+1)} \mid \theta^{(t)}\right)}\right\}, \tag{2.2}$$

where $p(\theta^{(t+1)})$ is the target distribution at the state $\theta^{(t+1)}$. When the state $\theta^{(t+1)}$ is rejected, the chain continues to remain at the current state $\theta^{(t)}$. In order to accept the candidate state with probability r, the acceptance probability r is compared to a random variable u that follows a uniform distribution, $Unif(0,1)$, such that the candidate state $\theta^{(t+1)}$ is accepted if $u < r$. A Metropolis-Hastings step is summarised as below.

Algorithm 1: Metropolis-Hastings step

- Draw the candidate $\theta^{(t+1)}$ from $q(\theta^{(t+1)} \mid \theta^{(t)})$
- Compute $r = \frac{p(\theta^{(t+1)})q(\theta^{(t)}|\theta^{(t+1)})}{p(\theta^{(t)})q(\theta^{(t+1)}|\theta^{(t)})}$
- Accept the candidate state with probability $\min\{1, r\}$, otherwise remain at θ^t when $\theta^{(t+1)} = \theta^{(t)}$.

In general, the Metropolis-Hastings algorithm (Hastings, 1970) is a generalisation of other commonly utilised MCMC algorithms. This includes the Metropolis algorithm (Metropolis et al., 1953), where the proposal distribution is symmetric such that $q\left(\theta^{(t+1)} \mid \theta^{(t)}\right) = q\left(\theta^{(t)} \mid \theta^{(t+1)}\right)$.

2.3.1.3 Gibbs sampling

A special case of the Metropolis-Hastings algorithm is introduced as the *Gibbs sampler* such that the proposal is always accepted. Each iteration of the Gibbs sampler cycles through the conditional distribution of all the parameters. In each iteration, new parameters are generated and the defined conditional distributions to be utilised for the next iteration are updated.

The sampler is suitable for situations where the joint distribution of the parameters of interest is difficult to sample from. For example, let $p(\theta_1, \theta_2, \theta_3)$ be a joint distribution that is difficult to sample from. However, the conditional distributions $p(\theta_1 \mid \theta_2, \theta_3)$, $p(\theta_2 \mid \theta_1, \theta_3)$

and $p(\theta_3 \mid \theta_1, \theta_2)$ are possible to simulate from and are often referred to as *full conditionals*. The advantage of Gibbs sampling is that it simplifies a complex high-dimensional problem and breaks it down into simpler low-dimensional problems. Formally, the algorithm is described as follows such that θ consists of b blocks at iteration t.

Algorithm 2: Gibbs sampling step

- Draw $\theta_1^{(t+1)}$ from $p\left(\theta_1 \mid \theta_2^{(t)}, \theta_3^{(t)}, \ldots, \theta_b^{(t)}\right)$
- Draw $\theta_2^{(t+1)}$ from $p\left(\theta_2 \mid \theta_1^{(t+1)}, \theta_3^{(t)}, \ldots, \theta_b^{(t)}\right)$
- ...

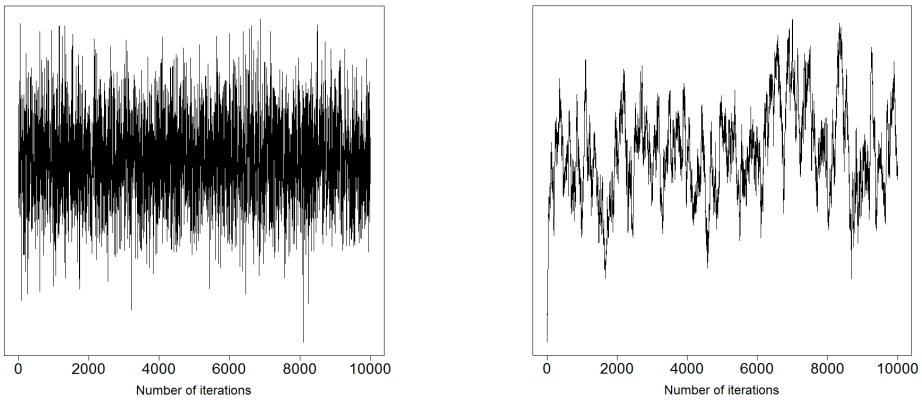
This corresponds to one iteration of the Gibbs sampler and produces a single draw of $\theta^{(t+1)}$; the above set of iterations are repeated many times.

This completes one iteration of the sampler and produces one draw of $\theta^{(t+1)}$. The Gibbs sampler is based on a property of the full conditionals as specified by the Hammersley-Clifford theorem; the full conditionals fully specify the joint distribution. However, it is not to be assumed that a set of proper well-defined conditional distributions will determine a marginal distribution. As a variation of Gibbs sampling, the Metropolis-Hastings sampler is utilised within the Gibbs sampler for updates of one or more of the conditional distributions. Such an approach is referred to as the Metropolis within Gibbs sampler. Other variations include the blocked Gibbs sampling and the collapsed Gibbs sampler.

2.3.1.4 MCMC convergence

An integral step of every MCMC is to check for evidence of *convergence*; Markov chains typically do not converge to the posterior distribution immediately. Sampling output from the posterior is visualised as a *trace plot* and relevant trace plots are typically examined to visualise the performance of the MCMC. Ideally, trace plots should look like 'fat hairy caterpillars' and Figure 2.4a displays an example of such a trace plot. An example of poorer performance of the MCMC is displayed in Figure 2.4b. It is often desirable to discard the beginning values of the Markov chain and this idea is known as *burn-in*. This eliminates

dependency on arbitrary initial values from the results. Burn-in is not necessarily required since a chain that is run long enough can reduce the impact of the initial values and lead to the same result. However, the removal of the initial values of the chain speeds up the process of achieving a valid result.



(a) (b)

Figure 2.4: Visual analysis: Trace plot

Besides visual diagnostics, more formal statistical diagnostic tests are also utilised to assess chain convergence. A popular statistic is the Gelman-Rubin diagnostic test (Brooks and Gelman, 1998, Gelman et al., 1992) that is based on analysing multiple simulated Markov chains. This is done so by a comparison of variances within each chain and the variance between chains for each model parameter. Extensive deviations between the estimated inter-chain and intra-chain variances indicate poor convergence of the chains. Examples of other diagnostic tests include the Geweke test (Geweke et al., 1983) and the Raftery and Lewis test (Raftery and Lewis, 1991). For a comparative review of convergence diagnostics, see Cowles and Carlin (1996).

Consecutive draws to generate a posterior sample from the MCMC chain can be highly correlated. This is referred to as *auto-correlation* and measures the dependency among the chains. A proposal being rejected very minimally is not necessarily positive since it indicates that the proposals are too cautious. This represents very small movements around

the posterior distribution and also leads to high auto-correlation. High auto-correlation diminishes the effectiveness of the number of samples, the goal of MCMCs is to simulate i.i.d. samples drawn directly from the target distribution. Ideally, samples should show low correlations, lower auto-correlation typically indicates better mixing of the chain and a faster rate of convergence.

2.3.2 Clustering

A *probability-based approach* to clustering overcomes many of the challenges that are found in the deterministic approaches to clustering (e.g., K-means). In this model, the data distribution is assumed to be a weighted sum of K component distributions. In the Gaussian mixture model (GMM), K component distributions follow a Gaussian distribution and each component corresponds to a cluster in the data. More formally, a GMM is given by $p(x \mid \Theta) = \sum_{l=1}^{K} \alpha_l p(x \mid \theta_l)$ where K denotes the number of Gaussian sources in the GMM, α_l is the weight of each Gaussian and $\theta_l = (\mu_l, \Sigma_l)$ represents the relevant parameters. The inference of parameters in Gaussian mixture models utilises computationally intensive methods such as MCMC methods, EM algorithm, etc. Finite mixture models have been extended and applied to datasets that pose spatial and temporal constraints (Blekas et al., 2004, Zhang et al., 2007). In parametric models (e.g., Gaussian mixture models), selecting the number of parameters is often difficult. Nonparametric or Infinite mixture models (Rasmussen, 2000) have several advantages over the finite mixture models; primarily that the number of clusters is determined from the dataset. Bayesian approaches to mixture modelling allow for the complex structure to be simplified through the use of latent variables. The Bayesian approach has been argued to be particularly suitable for scenarios where the number of components is unknown (Richardson and Green, 1997). In general, nonparametric does not mean 'no parameters', rather it means that the number of parameters grows with the number of data points. For example, a growing number of parameters can be in the context of more friend groups in social networks (Lim et al., 2016) or various representations within image segmentation results (Ghosh et al., 2011, Orbanz and Buhmann, 2008), etc. Such models have an infinite capacity to include number of clusters

and numerous nonparametric Bayesian models can be derived by starting with a standard parametric model. Nonparametric Bayesian models are an approach to constructing very flexible models and are able to better deal with the complexities of real data.

A common assumption is *independence* such that the joint probability can be expressed as the product of the probabilities of each data object.

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

Exchangeability is a weaker assumption that nonparametric Bayesian methods exploit. A data sequence is said to be *infinitely exchangeable*, if the distribution of any N data points is not altered when permuted. For any permutation σ,

$$p(x_1, \ldots, x_N) = p(x_{\sigma(1)}, \ldots, x_{\sigma(N)}) \tag{2.3}$$

In many statistical analyses, the random variables are independent and identically distributed (i.i.d.) and i.i.d. random variables are always infinitely exchangeable. However, an infinitely exchangeable sequence is not necessarily i.i.d. and this broader concept is used to define the De Finetti's theorem. The *De Finetti's theorem* states that a sequence $(x_1, \ldots, x_N)$ is infinitely exchangeable if and only if, for all N,

$$p(x_1, \ldots, x_N) = \int \prod_{i=1}^{N} p(x_i \mid \theta) p(\theta) d\theta \tag{2.4}$$

The motivation to utilise parameters and likelihoods and to place priors is justified by the De Finetti's theorem. The prior need not be finite dimensional and this provides a justification for non-parametric Bayesian priors.

A mixture model with infinite number of components is applied to a finite training set and results in only a finite number of components being used to model the data. As a Bayesian, model, the most common prior to use is the Dirichlet process (DP). The DP also induces a distribution over the partition of integers called the Chinese restaurant process (CRP). The Chinese restaurant process (CRP) (Pitman et al., 2002) is a discrete time stochastic

process that is utilised to produce exchangeable data and illustrates a generative model for data (this process is described in Chapter 4). The literature for Dirichlet processes and the alternatives to defining the DP is extensive and we do not attempt to provide an overview in this chapter. For a more comprehensive review, see Teh (2011). The generative model for observations from a CRP partitioning is referred to as a Dirichlet Process Mixture model (DPMM) (Rasmussen, 2000). This process is also known as the infinite Gaussian mixture model, such that the number of clusters can arbitrarily grow, to better accommodate data as needed, along with the assignment of relevant data points. Finite representation of infinite clusters avoid the problem posed by infinite number of parameters for inference.

However, the infinite Gaussian mixture model and relevant extensions have multiple restrictions. More specifically, the classical CRP does not have the ability to incorporate dependency information from non-exchangeable data. Non-exchangeable data includes datasets that have constraints (e.g., measurements at different geographic locations, observations recorded over time, network connectivity) such that the order does matter. Exchangeability is an assumption that is beneficial for many reasons, however, the data in many domains are not exchangeable. The traditional CRP based mixture model cannot incorporate such non-exchangeability. Models built on an assumption of exchangeability have limited applications and lifting the restrictions for domain specific applications can lead to very complex models. The distance dependent Chinese restaurant process (ddCRP) was introduced by Blei and Frazier (2011) to represent an alternative strategy for modelling non-exchangeability. This process expands the available infinite clustering methods and allows for numerous non-exchangeable distributions as priors on partitions. A detailed review of other priors related to the ddCRP as well as other priors adapted to deal with non-exchangeability is presented in Blei and Frazier (2011). The ddCRP has been utilised previously for image segmentation (Ghosh et al., 2011), clustering in combination with spectral methods for dimension reduction (Socher et al., 2011), a hierarchical generalisation (Ghosh et al., 2014), partitioning voxel measurements (Janssen et al., 2016) and clustering in phylogenetics (Cybis et al., 2018). A summary of the CRP, the ddCRP and a special case of the ddCRP that allows for the modelling of spatial constraints in a network is described

in Chapter 4.

Chapter 3

Functional Distributional Clustering

3.1 Introduction

Clustering is an unsupervised learning method that seeks to identify clusters with homogeneous characteristics. Conventional methods of this exploratory approach include hierarchical (Ward Jr, 1963) and partitional (e.g., K-means (MacQueen et al., 1967), etc.) techniques. A brief description of these deterministic clustering methods is included in Chapter 2. Hierarchical methods (divisive or agglomerative process) generate a set of clusters in which smaller clusters are nested within larger clusters and a dendrogram illustrates the arrangement of clusters generated by the clustering framework. On the other hand, K-means is a partitioning process which assigns objects to a pre-specified number of clusters. The ability of distance-based clustering methods such as hierarchical clustering and K-means to identify distinct clusters in heterogeneous data depends on the distance or dissimilarity measure (Hartigan, 1975, Kaufman and Rousseeuw, 2009, Shirkhorshidi et al., 2015). This chapter focusses on the development of a spatially adjusted clustering method within a hierarchical framework. Spatially adjusted clustering algorithms based on commonly used distance measures (e.g., Euclidean) might fail to preserve characteristics about the underlying data. Distance measures determined using nonparametric estimators do not make any assumptions about the distribution (e.g., Gaussian) of the data. Assuming

Gaussianity implies that differences between clusters should manifest in differences between means. However, domain knowledge might suggest that differences occur not just in the mean but also other aspects of the distribution such as spread and dispersion. A histogram is a widely used non-parametric density estimator and histogram clustering methods define a distance measure in numerous ways. For example, Kim and Billard (2013) define a distance between cumulative density functions using non-overlapping subintervals rather than individual observations and Irpino and Verde (2006) proposed a distance using the Wasserstein metric within an agglomerative hierarchical clustering framework. A measure of distance determined over data divided into classes (e.g., intervals and histogram-valued observations) enables a clustering algorithm to identify clusters that possess varying analytic characteristics but correspond to the same mean or median value. Kernel density estimators are visually similar to smoothed histograms and unlike histograms are smooth and provide a continuous representation. Numerous clustering algorithms exist for functional data (Jacques and Preda, 2014, Tarpey and Kinateder, 2003) and relevant measures of dissimilarity for functional data (Chen et al., 2014, Tzeng et al., 2016) have been defined between curves (e.g., over a time domain) to determine subgroups of representative curves that differ in shape and variation. However, fewer spatially adjusted clustering algorithms have been developed for functional data (Delicado et al., 2010, Giraldo et al., 2012, Haggarty et al., 2015, Secchi et al., 2011).

This chapter proposes a *functional distributional* clustering algorithm in an agglomerative hierarchical clustering framework such that a *unique* measure of distance is defined between conditional cumulative distribution functions (CDFs), where CDFs are estimated at different locations in space. A distance measure, defined using cumulative distribution functions rather than probability density functions (for a review see Cha (2007)), enables spread and dispersion (e.g., temporal patterns) in the distribution to be retained such that comparisons are not restricted to differences in summarised values (e.g., mean). This hierarchical clustering algorithm defines a measure of distance that is both functional and distributional since it defines a conditional cumulative distribution function using local averages of cumulative distribution kernels and density kernels over the recorded time series

and seeks to identify spatially contiguous clusters that correspond to groups of curves with distinct characteristics. *To the best of our knowledge, a clustering approach that is both functional and distributional has not been previously introduced.* This chapter focusses on the development of a relevant approach within a spatially adjusted agglomerative hierarchical clustering framework. The study seeks to highlight the use of hierarchical clustering as an exploratory tool and demonstrates the ability to visualise the differences in distribution over time, within the spatially contiguous clusters, using a series of three-dimensional plots.

For our analysis, we assume that a network is composed of sensors that are arranged in a way that can be used to define a neighbourhood structure, or to be more precise, an adjacency matrix. Formally, we assume that the network of sensors can be represented as an undirected graph with sensors as vertices and edges linking neighbouring sensors. Let $G = (V, E)$ be a graph, where V is a set of vertices and E is a collection of edges. Assume that $V = \{v_1, \ldots, v_N\}$ and the adjacency matrix of graph G is a square matrix $\mathbf{W}$ with elements $W_{ij} = 1$ if $\{v_i, v_j\} \in E$ (i.e., if there is an edge between vertices v_i and v_j) and $W_{ij} = 0$ otherwise. The examples in this chapter come from traffic modelling, where we assume that the urban road network is made up of junctions and road segments that link relevant junctions. Occupancy is the percentage of time that a location on the road is occupied by vehicles and a measurement of occupancy that describes congestion is available for each junction and unit of time. Junctions which are joined directly by a road segment are considered to be adjacent and our objective is to identify contiguous areas of similar traffic patterns. The development of clustering algorithms for urban road networks, using recorded occupancy observations, is of fundamental importance to traffic operators and traffic control centres (TCC) (Ji and Geroliminis, 2012, Saeedmanesh and Geroliminis, 2016, 2017). This is because the shape of aggregated models is affected by the spatio-temporal distribution of congestion in traffic networks; these models are used for traffic monitoring by traffic engineers and TCC. Traffic congestion usually propagates upstream in the network, to random locations, which can lead to noisy measurements being recorded by traffic sensors about adjacent regions. The spatial sensing area cannot necessarily be characterised by a regular disk or a radius. Therefore, geometric assumptions related to a measure of distance (e.g.

Euclidean) are not necessarily appropriate for monitoring spatio-temporal phenomena (such as traffic flow in urban traffic networks). Clusters obtained from the proposed method can be used from traffic control centres to apply innovative traffic management policies such as traffic signal control and perimeter control in order to mitigate traffic congestion and improve mobility (see e.g., Aboudolas and Geroliminis (2013), Aboudolas et al. (2009), Daganzo (2007)).

Figure 3.1 presents different scenarios that describe the distribution of occupancy observations for an individual junction. Successive jumps in occupancy levels over a period of time would be lost by clustering methods that fail to accommodate the distribution of occupancy levels and only include summary values. Instead, the distance measure incorporates distributions by utilising functions that are defined to account for temporal patterns. Figure 3.1a displays occupancy data, where levels of occupancy range between 0% and 100% and the overall mean value is 47%. In addition, Figure 3.1b, Figure 3.1c and Figure 3.1d also display occupancy observations that have a mean value of 47%. A clustering algorithm applied to the overall mean of these time series would be unable to detect differences between the four scenarios. Instead, the measure of distance in our approach, that is both functional and distributional, is able to account for dynamics in the recorded observations and retain information about underlying temporal patterns.

For example, Figure 3.1a and 3.1b have the same mean function over time: in both figures, the mean is around 30% for the first three hours and around 60% afterwards. The difference between the two scenarios is in the dispersion. Thus a clustering algorithm that is based on the mean function would not be able to detect a difference between these two scenarios. The marginal distribution of occupancies (ignoring time) is the same for scenarios displayed in Figure 3.1a and 3.1c. Only a clustering method that can capture temporal dynamics can differentiate between these two scenarios. The proposed method is both functional and distributional, so it would be able to detect differences between any of the four scenarios. Figure 3.2 displays a series of three-dimensional plots to describe the scenarios in Figure 3.1a, Figure 3.1b, Figure 3.1c and Figure 3.1d. Each plot displays a unique set of curves that

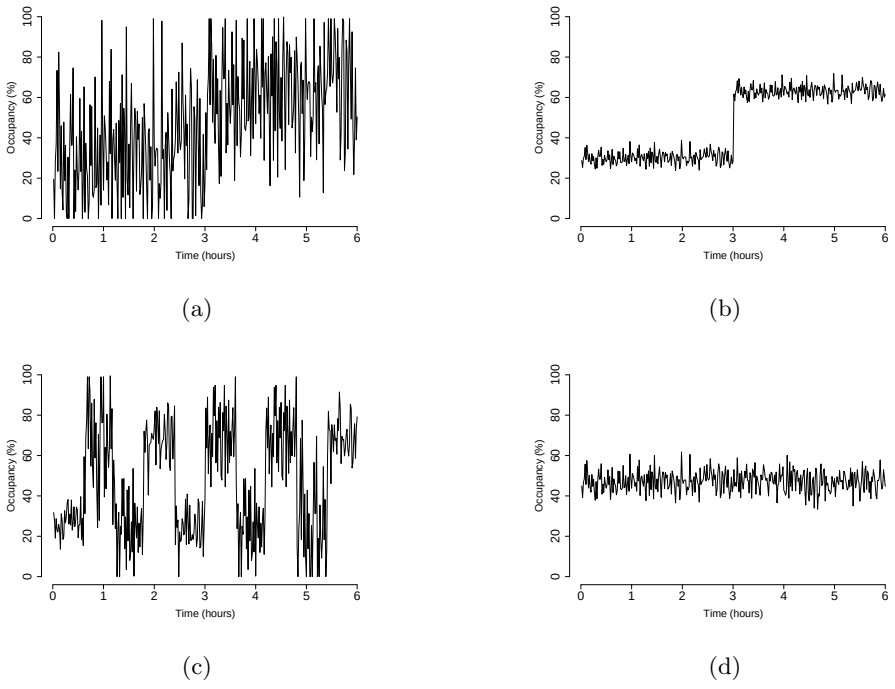


Figure 3.1: Each scenario displays occupancy measurements recorded over six hours (21600 seconds) for an individual junction. The mean value over observations in each scenario is $\sim 47\%$

vary in shape, concentration of occupancies and range of values over time. The differences within each plot in Figure 3.2 highlights the need for our functional distributional approach as opposed to other available hierarchical clustering algorithms. The example in this chapter focusses on observations associated with a grid style urban traffic network and is described in the context of traffic modelling. However, the functional distributional clustering method is applicable to any dataset associated with a spatial structure (e.g., data recorded over grid style networks or areal unit data for maps) and where observations are recorded over time for each unit.

The rest of the chapter is organised as follows. Section 3.2 provides a background of density estimation and summarises histograms and the kernel density estimator. Section 3.3 proposes the functional distributional clustering algorithm and the relevant bandwidth selection method. This section also describes methods to choose the optimal number of clusters and a measure of clustering similarity between identified clusters and a given set

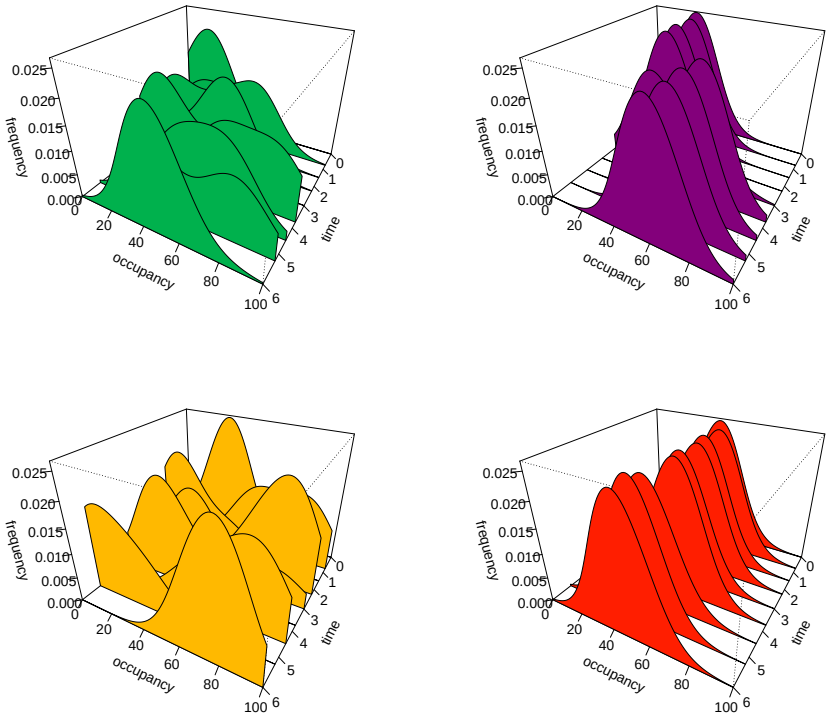


Figure 3.2: Series of three-dimensional plots corresponding to the scenarios displayed in Figure 3.1.

of 'true' clusters. The functional distributional clustering algorithm and corresponding components described in this chapter are available for implementation in the R package **FdiClust** at https://github.com/AshwiniKV/FdiClust. Section 3.4 presents an application of this algorithm to pre-defined data generated from an accurate micro-simulator for a 2.5 square miles network area in downtown San Francisco, CA. The simulation study evaluates the performance of the algorithm by comparing pairs of clusters obtained using distance measures with and without CDFs. The simulation study is also extended to include a comparison to clusters obtained from algorithms that use functional data analysis (FDA) related techniques within a spatially adjusted agglomerative hierarchical clustering framework. We show that the functional distributional clustering algorithm beats FDA-based approaches (that use principal component analysis (PCA) or the B-spline basis) as well as a recently developed Ward-like hierarchical clustering method named ClustGeo. In Section 3.5, we illustrate the application of this algorithm to real data for the same traffic

network and duration, but with no knowledge of underlying 'true' clusters. The clustering algorithm serves as an exploratory tool, where the clustering output and the series of three dimensional plots are used to highlight differences in the distribution across time. Finally, in Section 3.6, we summarise the algorithm and highlight its advantages and disadvantages.

3.2 Background: Density estimation

Density estimation is an important topic in statistical research and numerous approaches to density estimation exist including Parzen windows, histograms and kernel density estimators. A primary component for the development of the functional distributional clustering algorithm is the utilisation of non-parametric estimators. A non-parametric density estimator, unlike a parametric estimator seeks to estimate the density directly from the data without assuming a particular functional form for the underlying data. Let there exist a random variable X, where $f(x)$ is the probability density function (PDF) of X. The PDF $f(x)$ satisfies two conditions namely, $f(x) \geqslant 0$ and $\int f(x)dx = 1$. In general, determined probabilities relate to the area under the defined PDF and can also be represented using a cumulative distribution function (CDF). Let the CDF be defined as $F(x) = \int_{-\infty}^{x} f(u)du$ and $F(b) - F(a) = \int_{a}^{b} f(x)dx$. In this section, we focus on the ability to construct the probability density estimate over the set of data points.

3.2.1 Histograms

The simplest form of a nonparametric estimator of a probability distribution is the *histogram*. A histogram is constructed by considering equal sub-intervals from the relevant data i.e., bins and the relevant end point of the bins. In other words, the histogram requires two parameters to be defined, the bin width and the starting position of the first bin. To construct a simple example, first assume that $X_i \in [0, 1]$ such that $p(x)$ is non-zero within [0, 1]. The histogram seeks to partition the set [0, 1] into M bins and this leads to a partition as

$$B_1 = \left[0, \frac{1}{M}\right), B_2 = \left[\frac{1}{M}, \frac{2}{M}\right), \ldots, B_{M-1} = \left[\frac{M-2}{M}, \frac{M-1}{M}\right), B_M = \left[\frac{M-1}{M}, 1\right]$$

The density for a point $x \in B_l$ is estimated from the histogram as

$$\hat{p}(x) = \frac{\text{number of observations within } B_l}{n} \times \frac{1}{\text{length of bin}} = \frac{M}{n} \sum_{i=1}^{n} I(X_i \in B_l)$$

This density estimator assigns an equal density value to points within the bin. The bin B_l contains x and the ratio of observations within this bin is $\frac{1}{n}\sum_{i=1}^{n} I(X_i \in B_l)$. The ratio of observations should be equal to the density estimate times the bin length $\frac{1}{M}$.

Histogram-based clustering techniques have been implemented in combination with many methods (e.g., hierarchical, support vector machine, K-means, etc.); such methods have been primarily motivated by problems in image segmentation research. However, the histogram has several drawbacks; primarily the density estimate depends on the starting position of the bins. In addition, the discontinuities of the estimate are not dependent on the underlying density; making understanding the structure of the data challenging. For a more comprehensive review of histograms and its utilisation in clustering, see (Freedman and Diaconis, 1981, Tsai and Chen, 1992).

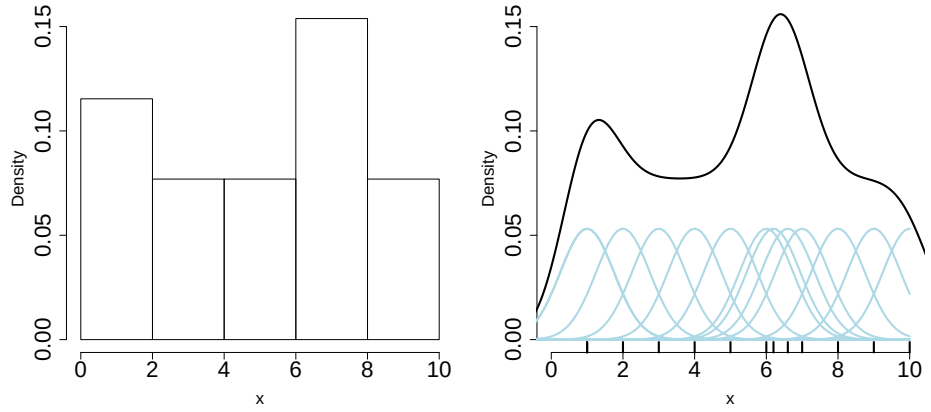
3.2.2 Kernel density estimators

The *kernel density estimator* serves as a way to alleviate the problems posed by the commonly used histogram method. Kernel density estimators are non-parametric density estimators that do not have a fixed functional form and determine an estimate over all available data points. The kernel density estimator removes the dependence on the starting points of the bins (in histograms) by defining a kernel function at each data point.

Formally, kernel density estimators smooth the contribution of each observed data point over the local neighbourhood of the relevant data point. Let the kernel function be denoted as K and its bandwidth be denoted as h. A random sample of data exists from an unknown distribution and is said to have a probability distribution function denoted by $f(x)$ and a cumulative distribution function $F(x)$. The estimated density at any point x can also be

written as:

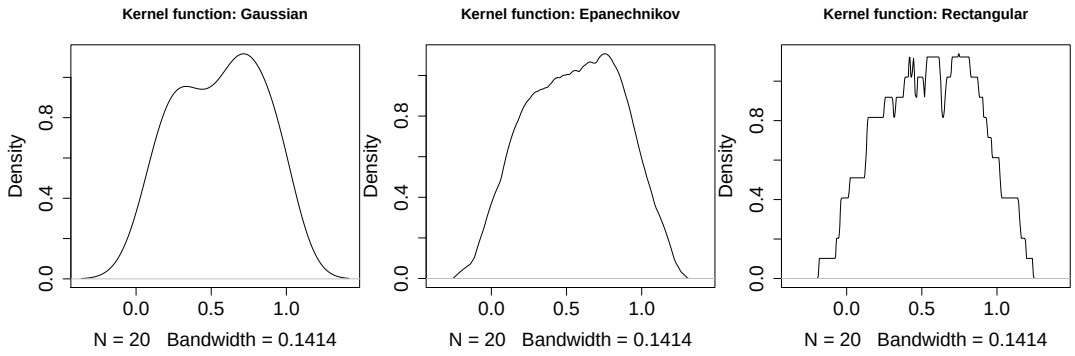
$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x(i)}{h}\right) \tag{3.1}$$



(a) Histogram (b) Kernel density curve

Figure 3.3: Commonly used nonparameteric estimators

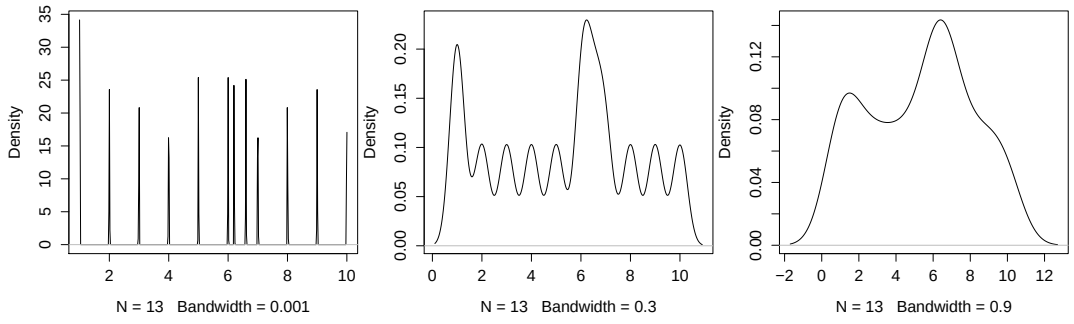
Let $\int K(t)\mathrm{dt} = 1$ and $K(x) \geqslant 0$, i.e., $\forall$ -$\infty < x < \infty$ K is a non-negative function that is symmetric around zero and integrates to 1. The Gaussian kernel function is most commonly chosen but other kernel functions including Uniform, Triangle, Epanechnikov, Rectangular, etc can also be chosen. Figure 3.4 displays densities estimated using different kernel functions, namely the Gaussian, Epanechnikov and Rectangular kernel functions.



(a) Gaussian (b) Epanechnikov (c) Rectangular

Figure 3.4: Estimated densities using different kernel functions

The quality of a kernel estimate is directly affected by the choice of the value of bandwidth h. This leads to a need to choose the appropriate bandwidth such that it is not too small or not too large. Small values of the bandwidth h lead to estimates that are spiky, while larger values of the bandwidth h obscure the structure of the underlying data. In other words, small values of the bandwidth undersmooth the data, larger values oversmooth the data. In Figure 3.4, the plots display density curves at different values of the bandwidth ($h = 0.001, h = 0.3$ and $h = 9$).



(a) (b) (c)

Figure 3.5: Density curves at different bandwidth values

Ideally, a formal process of choosing the bandwidth h minimises the error between the estimated density and the true density. A natural measurement of discrepancy for estimation at a single point x is defined using the mean square error (MSE). In this chapter, the functional distribution clustering method utilises a distance measure that is defined using the kernel density estimator and the rest of the chapter describes this method and its relevant components.

3.3 Clustering model

This section proceeds in two stages to set out the proposed functional distributional clustering method that identifies spatially contiguous clusters across the network and incorporates temporal patterns of recorded observations. The first stage utilises a hierarchical agglomerative clustering algorithm and generates a series of cluster configurations. The clustering

algorithm is built on a measure of distance that is defined using estimated conditional cumulative distribution functions (CDFs) for each cluster. The measure of distance is determined utilising functions calculated over individual observations rather than aggregated summary values. In the second stage, we use a clearly defined criterion to determine the optimal number of clusters and generate a distinct partition structure of the network. We also describe a measure of clustering similarity to examine the accuracy of identified clusters.

3.3.1 Hierarchical agglomerative clustering algorithm

In a hierarchical clustering approach, a partition occurs at each level to determine non-overlapping clusters. Let observations for sensor j at time t_i be denoted by x_{ij}, where $i = 1, \ldots, n$ and $j = 1, \ldots, N$ and Table 3.1 describes the recorded observations.

i	1	2	3	...	n
times (t_i)	t_1	t_2	t_3	...	t_n
sensor readings ($j = 1$)	x_{11}	x_{21}	x_{31}	...	x_{n1}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
sensor readings ($j = N$)	x_{1N}	x_{2N}	x_{3N}	...	x_{nN}

Table 3.1: Representation of the observations x_{ij} recorded over time $t_i = t_1 \ldots t_n$ for $j = 1 \ldots N$ sensors.

The probability density function (PDF) for observations relevant to sensor j at x_0 is estimated by $\hat{f}^{(j)}(x_0) = \dfrac{1}{nh_x} \displaystyle\sum_{i=1}^{n} \phi\left(\frac{x_{ij} - x_0}{h_x}\right)$, where the contribution of observation x_{ij} to an estimate at x_0 depends on how apart x_{ij} and x_0 are and $\phi(.)$ is a standard normal PDF. Hall et al. (2004) and Li and Racine (2008) propose that the probability density function (PDF) conditional on t_i is estimated at x_0 as

$$\hat{f}_{t_i}^{(j)}(x_0) = \frac{1}{h_x} \sum_{i=1}^{n} \phi\left(\frac{x_{ij} - x_0}{h_x}\right) w_{t_0}(t_i). \tag{3.2}$$

In Equation (3.2), $w_{t_0}(t_i) = \dfrac{\phi\left(\dfrac{t_0 - t_i}{h_t}\right)}{\displaystyle\sum_{\eta=1}^{n} \phi\left(\frac{t_\eta - t_i}{h_t}\right)}$, h_t is a bandwidth defined for time and h_x is a bandwidth which corresponds to recorded observations. The basic kernel density estimator

is modified to reasonably account for variation over time when applied to observations recorded over time. Let a set of clusters $\mathcal{C}_{l=1}$ be represented by $\mathcal{C}_1 = \{C_1, \ldots, C_k\}$. In an agglomerative hierarchical clustering framework, each cluster is initially composed of a single sensor; the set of clusters can accordingly be written as $\mathcal{C}_1 = \{C_1, \ldots, C_k\} = \{\{1\}, \{2\}, \ldots, \{N\}\}$. In this case, k is equal to N. At subsequent levels of the algorithm, clusters are consolidated and eventually form a single larger cluster ($k = 1$) composed of all N sensors in the network such that $\mathcal{C}_1 = \{C_1\}$. The conditional probability density function for a cluster C is determined over observations recorded for relevant sensors and is defined as $\hat{f}_{t_i}^{(C)}(x_0) = \frac{1}{|C|} \sum_{j \in C} \hat{f}_{t_i}^{(j)}(x_0)$. The estimator of the conditional cumulative distribution function (CDF) is defined as

$$\hat{F}_{t_i}^{(j)}(x_0) = \sum_{i=1}^{n} \Phi\left(\frac{x_{ij} - x_0}{h_x}\right) w_{t_0}(t_i), \tag{3.3}$$

where $\Phi(\cdot)$ is a standard normal CDF and $\hat{F}_{t_i}^{(C)}(x_0) = \frac{1}{|C|} \sum_{j \in C} \hat{F}_{t_i}^{(j)}(x_0)$. The relevant theoretical properties for the estimated conditional density functions are described in Fan and Yim (2004). A single observation for each sensor provides less information about temporal patterns compared to a single value from $\hat{F}_{t_i}^{(j)}(x_0)$ and the conditional CDF retains all the sensor readings over time.

Within the hierarchical clustering framework, a pair of clusters C_1 and C_2 are merged if they have the lowest distance compared to distances calculated for all other pairs of clusters. The distance d is built using a L_1 norm, rather than the more commonly used L_2 norm or squared L_2 norm and distance d is determined over estimated conditional CDFs rather than individual observations. Let the distance d between cluster C_1 and cluster C_2 at time t_i be defined as the area between the two CDFs, i.e.

$$d\left(\hat{F}_{t_i}^{(C_1)}(\cdot), \hat{F}_{t_i}^{(C_2)}(\cdot)\right) = \int \left|\hat{F}_{t_i}^{(C_1)}(x_0) - \hat{F}_{t_i}^{(C_2)}(x_0)\right| \, dx_0 \approx \Delta \sum_{s=1}^{S} \left|\hat{F}_{t_i}^{(C_1)}(\xi_s) - \hat{F}_{t_i}^{(C_2)}(\xi_s)\right| \tag{3.4}$$

for a regular grid $\xi_1, \ldots, \xi_S$ with $\xi_{s+1} - \xi_s = \Delta$.

Accordingly, let D be a distance matrix, where distance between cluster C_1 and C_2 in the

matrix is defined as the sum of the above distance over time $t_1, \ldots, t_n$.

$$D_{C_1,C_2} = \begin{cases} \sum_{i=1}^{n} d\left(\hat{F}_{t_i}^{(C_1)}(\cdot), \hat{F}_{t_i}^{(C_2)}(\cdot)\right) & \text{if } C_1 \sim C_2 \\ \infty \text{ otherwise} \end{cases} \tag{3.5}$$

and $C_1 \sim C_2$ indicates that calculating the distance between clusters is feasible only if a link exists between any two sensors in the clusters. This condition ensures that identified clusters are spatially contiguous and any two clusters are merged at each iteration such that they correspond to the lowest distance d. The CDFs corresponding to the clusters C_1 and C_2 are also merged as:

$$\hat{F}_{t_i}^{(C_1 \cup C_2)}(x_0) = \frac{|C_1|}{|C_1| + |C_2|} \hat{F}_{t_i}^{(C_1)}(x_0) + \frac{|C_2|}{|C_1| + |C_2|} \hat{F}_{t_i}^{(C_2)}(x_0). \tag{3.6}$$

Updated CDFs are then utilised to calculate the distance d at each subsequent iteration and this process continues until a single larger cluster containing every sensor in the network is obtained.

Algorithm 3: Functional distributional clustering

Input : Initialize $\mathcal{C}_{l=1}$, where $\mathcal{C}_1 = \{C_1 \ldots, C_k\}$. At this level, the Nth sensor belongs to the kth cluster, i.e., $\mathcal{C}_1 = \{C_1 \ldots, C_k\} = \{\{1\}, \ldots, \{N\}\}$.

Output: Hierarchical set of clusters, ζ.

if $|\mathcal{C}_l| > 1$ **then**

1. For all pairs of clusters, compute distance d as defined in Equation (3.5).
2. Set $\{C_1, C_2\} = \underset{C_1, C_2 \in \mathcal{C}_l}{\operatorname{argmin}}(D_{C_1,C_2})$ to identify the pair of clusters that correspond to the minimum distance.
3. Merge the pair of clusters C_1 and C_2 as $C_1 \cup C_2$.
4. Update $\mathcal{C}_l$ to $\mathcal{C}_l \backslash \{C_1, C_2\} \cup \{C_1 \cup C_2\}$ and $\hat{F}_{t_i}^{(C_1)}(x_0)$ and $\hat{F}_{t_i}^{(C_2)}(x_0)$ using Equation (3.6).

else

return ζ;

end

3.3.2 Bandwidth selection

This section addresses the selection of smoothing parameters or bandwidths to estimate the conditional PDF $\hat{f}_{t_i}^{(j)}(x_0)$ defined in Equation (3.2). A data driven method such as cross-validation (Bowman, 1984, Rudemo, 1982) selects the bandwidth that corresponds to the minimum of the expected loss function and avoids the arbitrary selection of bandwidths that can lead to under smoothing or over smoothing. We use an extended cross-validation method developed by Fan and Yim (2004) to select optimal bandwidths h_x and h_t and denote an estimated conditional PDF for a cluster C dependent on the bandwidths as $\hat{f}_{t_i}^{(C)h}(x_0)$. The integrated squared error (ISE) is defined as

$$ISE = \frac{1}{|\mathcal{C}_l|} \sum_{C \in \mathcal{C}_l} \left(\frac{1}{n} \sum_{i=1}^{n} \int \{\hat{f}_{t_i}^{(C)h}(x_0) - f_{t_i}^{(C)}(x_0)\}^2 \, dx_0 \right)$$

$$= \frac{1}{|\mathcal{C}_l|} \sum_{C \in \mathcal{C}_l} \left(\frac{1}{n} \sum_{i=1}^{n} \int \hat{f}_{t_i}^{(C)h}(x_0)^2 \, dx_0 - \frac{2}{n} \sum_{i=1}^{n} \int \hat{f}_{t_i}^{(C)h}(x_0) f_{t_i}^{(C)}(x_0) \, dx_0 \right.$$
$$\left. + \frac{1}{n} \sum_{i=1}^{n} \int f_{t_i}^{(C)}(x_0)^2 \, dx_0 \right).$$

The last term is not dependent on bandwidth h and accordingly can be ignored in the bandwidth selection process. A reasonable estimator of the ISE is

$$CV(h) = \frac{1}{|\mathcal{C}_l|} \sum_{C \in \mathcal{C}_l} \left(\frac{1}{n} \sum_{i=1}^{n} \int \hat{f}_{t_i}^{(C)h}(x_0)^2 \, dx_0 - \frac{2}{n|C|} \sum_{i=1}^{n} \sum_{j \in C} \hat{f}_{t_i,-ij}^{(C)h}(x_{ij}) \right), \tag{3.7}$$

where $\hat{f}_{t_i,-ij}$ estimates the conditional density function over the data sample, where the data sample excludes the observation x_{ij} for sensor j. The optimal bandwidth parameter corresponds to the minimum cross validation error $\hat{h} = \underset{h^*}{\text{argmin}} \; CV(h^*)$. The optimal bandwidths, i.e., h_x and h_t are determined through a grid search. One could argue that the bandwidth should be re-tuned for each update in the cluster structure; however, to reduce the computational footprint we determine the optimal bandwidth only at the beginning of the algorithm. Towards the end of the algorithm, clusters are substantially bigger and there could be scope to further reduce the bandwidths. We have found that using the same bandwidth throughout the algorithm usually gives similar clusterings.

3.3.3 Optimal number of clusters

A major challenge in clustering is the identification of the optimal number of clusters. In hierarchical clustering algorithms, the assignment of parameters to determine clusters often relies on the number of 'true' clusters, which may not necessarily be available or easily defined. Methods of cluster validation to determine the 'true' number of clusters include the CH index (Caliński and Harabasz, 1974), Dunn index (Dunn, 1973), Davies-Bouldin index (Davies and Bouldin, 1979), and the Silhouette index (Rousseeuw, 1987) and these methods seek to identify compact and well separated clusters, where clusters are deemed to be more distinct for smaller values of the index. In comparison to other methods, the time complexity for computation of the Davies-Bouldin index was found to be far lower than for the Silhouette method (Petrovic, 2006). Alternatively, the *gap statistic* (Tibshirani et al., 2001) compares within-cluster errors in the observed data to within-cluster errors calculated for data from an appropriate null reference distribution and removes the need for calculating validation scores. However, the need to bootstrap samples in the gap statistic approach leads to the method being rather computationally expensive and inefficient for calculating the number of clusters.

We modify the *clustering balance criterion* (Jung et al., 2003), a method similar to the Davies-Bouldin index, to compare the inter-cluster distances and intra-cluster distances in a computationally efficient manner for larger datasets. Let the aggregated CDF over all sensors in a cluster C be defined as $F_{t_i}^{(C)}(\cdot) = \frac{1}{|C|}\sum_{j\in C} F_{t_i}^{(j)}(\cdot)$. Using this definition, let $\Lambda = \sum_{i=1}^{n}\sum_{C\in\mathcal{C}_l}\sum_{j\in C} d\left(F_{t_i}^{(j)}(\cdot), F_{t_i}^{(C)}(\cdot)\right)$ be the intra-cluster distance sum calculated for all k identified clusters in $\mathcal{C}_l$. The inter-cluster distance sum is defined by $\Gamma = \sum_{i=1}^{n}\sum_{C\in\mathcal{C}_l} d\left(F_{t_i}^{(C)}(\cdot), F_{t_i}^{(C_0)}(\cdot)\right)$, where $F_{t_i}^{(C_0)}(\cdot) = \frac{1}{|\mathcal{C}_l|}\sum_{C\in\mathcal{C}_l} F_{t_i}^{(C)}(\cdot)$. Within an agglomerative hierarchical clustering framework, the intra-cluster sum Λ has zero distance for singleton clusters and this value is maximised when all sensors in the network belong to a single cluster. On the other hand, the inter-cluster sum Γ is minimised when all sensors belong to a single cluster and maximised when each sensor is a singleton cluster. Accordingly, the clustering balance is defined

as $\epsilon = \alpha\Lambda + (1-\alpha)\Gamma$, where weights α and $1-\alpha$ are assigned to Λ and Γ. In the examples, we used an α value of 0.5. The hierarchical clustering algorithm described above yields a sequence of nested partitions. We then retain the partition minimising the above modification of the clustering balance criterion, which is deemed to have optimal number of clusters.

3.3.4 Measure of clustering similarity

The optimal number of clusters determines objects within each cluster by utilising the constructed hierarchy of clusters. This set of defined clusters and their elements are compared against external criteria such as a pre-defined cluster structure or known set of labels. Let a set of sensors in the network be defined as $\mathcal{J} = \{1, 2, 3, \ldots N\}$ and $\mathcal{U}$ and $\mathcal{V}$ are two partitions of $\mathcal{J}$, where $\mathcal{U} = \{U_1, \ldots, U_u\}$ is defined as the set of u true clusters and $\mathcal{V} = \{V_1, \ldots, V_v\}$ represents a clustering result composed of v clusters. Let a be the number of pairs of sensors in $\mathcal{J}$ that are in the same cluster within $\mathcal{U}$ and the same cluster within $\mathcal{V}$, b be the number of pairs of sensors in $\mathcal{J}$ that are in the same cluster in $\mathcal{U}$ but not the same cluster in $\mathcal{V}$, c be the number of pairs of sensors in $\mathcal{J}$ that are not in the same cluster in $\mathcal{U}$ but in the same cluster in $\mathcal{V}$, and d be the number of pairs of sensors in $\mathcal{J}$ that are in different clusters for both $\mathcal{U}$ and $\mathcal{V}$. Similarity measures between clustering results and 'true' clusters can be calculated using a method called the *Rand index* (RI) (Rand, 1971). The Rand index is then defined as $\text{RI} = \frac{a+d}{a+b+c+d}$, where $a+d$ refers to the number of agreements between the clustering output of the developed algorithm and the given truth and $a + b + c + d$ includes both agreements and disagreements. Values of the RI lie between 0 and 1, where 0 represents little agreement and 1 represents strong agreement. However, the expected value of the RI for two random partitions does not necessarily take a constant value and the RI approaches an upper limit of unity as the number of clusters increases. A modified version of the RI was introduced by Hubert and Arabie (1985) to account for problems within the RI method and is called the *Adjusted Rand index* (ARI). In general, a larger ARI indicates a higher agreement between two partitions and the ARI has a maximum value of 1 but can also take negative values. This index is typically recommended as the choice for measuring agreement between any two clustering results even when the number of clusters are different

(Milligan and Cooper, 1986) and is computed using:

$$\frac{(a+b+c+d)(a+d) - \left[(a+b)(a+c)+(c+d)(b+d)\right]}{(a+b+c+d)^2 - \left[(a+b)(a+c)+(c+d)(b+d)\right]}. \tag{3.8}$$

3.4 Simulated occupancy data

In this section, the group of sensors arranged as a network correspond to junctions within an urban road network, where adjacent junctions are linked by road segments. An urban road network constitutes a network which can be represented as an undirected graph with junctions as vertices and road segments that link relevant junctions as edges.

3.4.1 Data

We simulate occupancy data over a 2.5 square miles network area in Downtown San Francisco, California composed of $N = 158$ junctions and 316 links to reflect a heterogeneous network composed of homogeneous clusters. Correlated occupancy data is generated in R version 3.4.2 (R Core Team, 2013) using a spatio-temporal precision matrix to define three distinct clusters in the network, where within each cluster in $\mathcal{C}_l$, a given state space model generates zero and one values corresponding to defined occupancy levels. We assume that each junction within an urban road network has a maximum of four links to adjacent junctions. The presence of a limited number of road segments between junctions in the network leads to a sparse spatial precision matrix modelled as a type of conditional auto-regressive (CAR) model (Leroux et al., 2000). The temporal precision structure is defined as a first order auto-regressive model (AR-1) and occupancy observations for each junction are recorded over a period of six hours (21600 seconds) with a sampling rate of 60 seconds. Figure 3.6 illustrates the simulated occupancy data to represent distinct clusters. Occupancy values (20 – 50%) displayed in purple for cluster A are typically lower and variations in jumps between successive observations reduce over time. The values (40 – 100%) plotted in yellow for cluster C are composed of both higher and lower values, with differences between successive observations reducing marginally over time. Occupancy values (70 – 100%) in green for cluster B are typically higher in the first three hours and display greater variation (50 – 90%) over the next three hours.

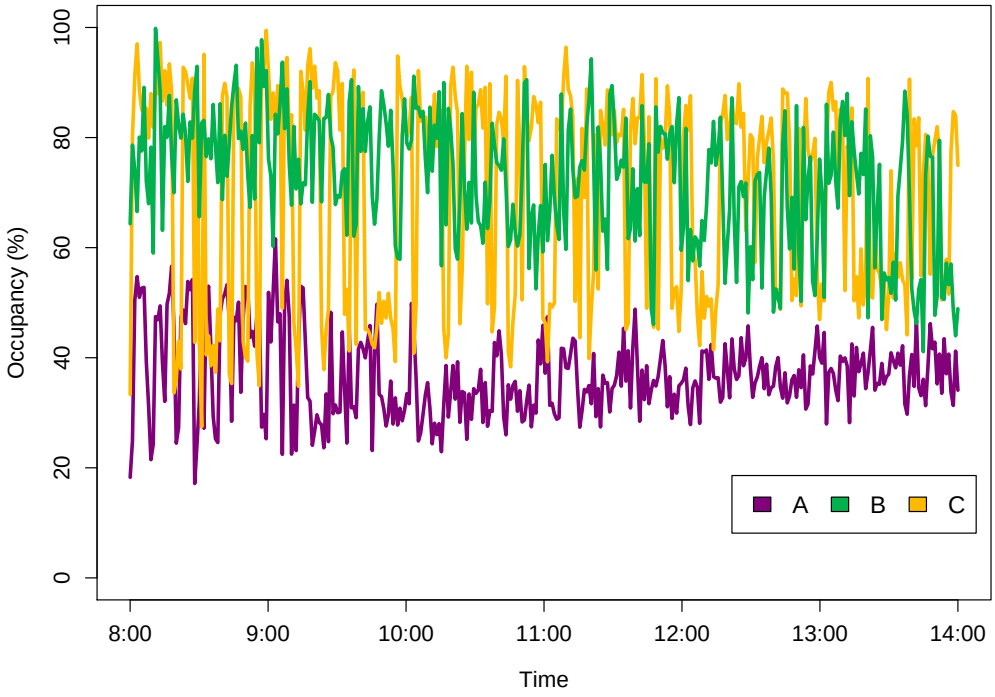
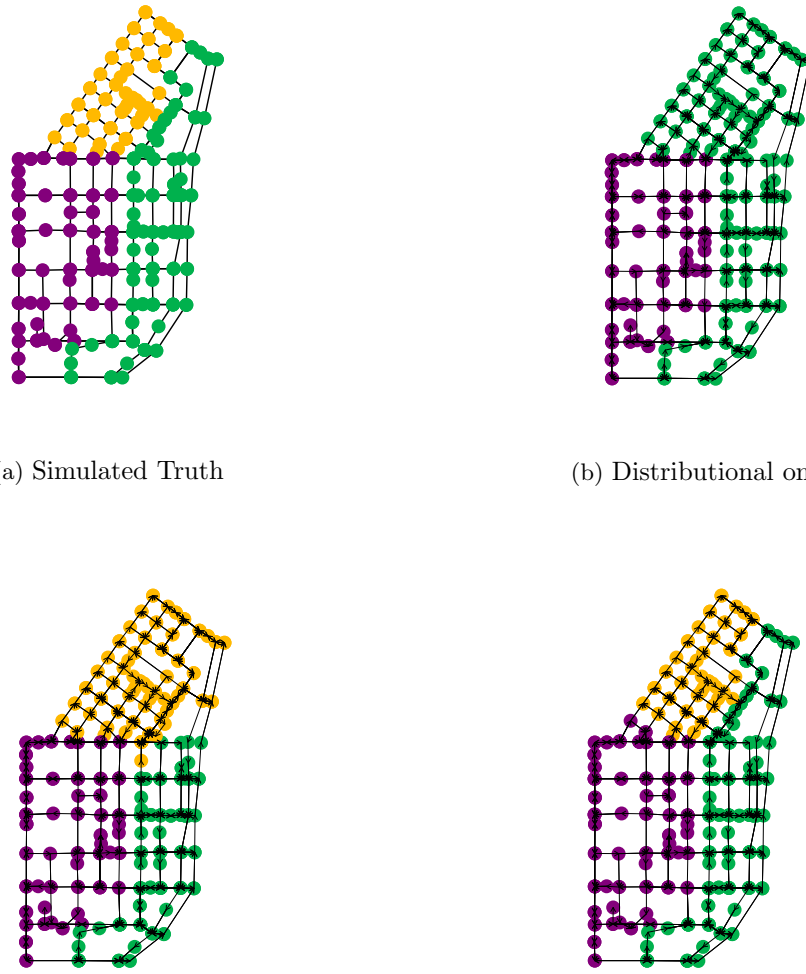


Figure 3.6: Occupancy measurements generated for three distinct clusters.

3.4.2 Results

The proposed algorithm introduced in Section 3.3.1 is applied to simulated occupancy observations generated within the urban network as described in Section 3.4.1. Each junction is initially treated as a singleton within the agglomerative clustering framework. The conditional CDF $F_{t_i}^{(C)}(x_0)$ for a cluster C is estimated over a sample of 360 observations (sampling rate of 60 seconds), where bandwidths $h_x = 10$ (occupancies recorded in %) and $h_t = 6$ (time in seconds) are selected using the extended cross validation method described in Section 3.3.2. Conditional CDFs are estimated for each cluster and stored outside individual iterations of the algorithm to improve the proposed algorithm's computational efficiency. The distance d is calculated between adjacent clusters using Equation (3.4) and (3.5) and individual clusters are merged at each iteration of the algorithm corresponding to the minimum distance. This process stops when all junctions belong to a single larger cluster and

we obtain a series of merged clusters from the hierarchical clustering algorithm.



(a) Simulated Truth

(b) Distributional only

(c) Functional Only

(d) Functional and Distributional

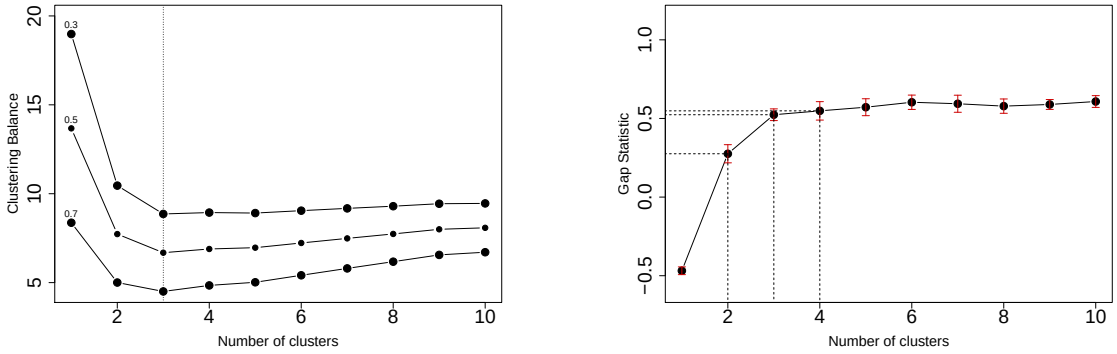
Figure 3.7: Clustering algorithm applied to data simulated in the network for a period of six hours.

Figure 3.7 displays networks with clusters identified by three different clustering algorithm scenarios and the defined 'true' clusters. These 'true' clusters in Figure 3.7a correspond to the simulated occupancy data in Figure 3.6. Figure 3.7b displays clusters identified when the distance measure uses Equations (3.2) and (3.3) with only observations over time and without the functions ϕ and Φ. Cluster C is not identified as distinct from cluster B and the *distributional only* algorithm is unable to determine the 'true' clusters. In particular,

the algorithm is unable to identify the cluster C which is composed of occupancy observations that successively jump between high and low values. Figure 3.7c depicts three clusters identified by the *functional only* algorithm, where Equation (3.3) is determined using observations aggregated over time. In Figure 3.7c, the identified clusters reflect the diminished ability of the algorithm to distinguish between cluster C and cluster B as compared to the clusters identified in Figure 3.7d. The clustered network in Figure 3.7d displays results of the *functional distributional clustering* algorithm that calculates $F_{t_i}^{(C)}(x_0)$ using all components in Equation (3.3). This algorithm is functional and distributional because distance measures are calculated using conditional CDFs for occupancy observations recorded over time. The clusters identified by the functional distributional algorithm are nearly equivalent to the three 'true' clusters displayed in Figure 3.7a. This indicates the ability of the functional distributional algorithm to recover the true spatially contiguous clusters when each cluster corresponds to a distinct distribution of occupancy observations.

The optimal number of clusters within the network is determined using both the commonly used gap statistic and a clustering balance criterion defined in Section 3.3.3. For each clustering algorithm, the gap statistic and clustering balance criterion are calculated for scenarios ranging from when the network has ten clusters to a scenario when the all the sensors belong to a single cluster. Figure 3.8a and Figure 3.8b display the clustering balance criterion and gap statistic against the corresponding number of clusters for results determined by the functional distributional clustering algorithm. The clustering balance criterion selects $k = 3$ for $\alpha = 0.5$ and for higher and lower values of α. The gap statistic chooses minimum k such that $\text{Gap}(k) \geqslant \text{Gap}(k+1) - s_{k+1}$ and this rule also determines that $k = 3$. However, determining bootstrap samples for the gap statistic is computationally expensive and we utilise the clustering balance criterion to determine the optimal number of clusters in Section 3.4.3 and Section 6.

To compare the clusters identified by the functional and distributional clustering algorithm to the 'true' clusters displayed in Figure 3.7a, we calculate the Adjusted Rand index (ARI) discussed in Section 3.3.4. ARI indicates agreement between a set of clusters $\mathcal{V}$ that is determined by the functional distributional clustering algorithm and a set of 'true' clusters $\mathcal{U}$ and is equivalent to 0.93. Similarly, $\mathcal{V}$ determined by the functional only algorithm results



(a) Clustering balance criterion

(b) Gap statistic

Figure 3.8: Methods to determine the optimal number of clusters.

in an ARI of 0.68 for three clusters and $\mathcal{V}$ determined by the distributional only algorithm leads to an ARI of 0.57 for two identified clusters. The functional only algorithm is unable to correctly identify all the junctions belonging to cluster B and the distributional only algorithm is able to only identify two out of three distinct clusters.

Figure 3.9 displays three-dimensional density plots for occupancy observations that correspond to clusters identified by the functional distributional algorithm in Figure 3.7d. These plots describe a relationship for each cluster between 100 occupancy observations (values between 0% and 100%), a time period of six hours (21600 seconds) with a sampling rate of sixty seconds and estimates for a Gaussian kernel density (over occupancy observations within the relevant cluster) with bandwidth equivalent to 15%. This value of bandwidth enables meaningful comparisons among curves within a cluster; lower values result in 'choppier' density curves that inhibit the ability to identify differences. In Figure 3.9, the sub-plot for cluster A represents observations with density levels between 0.015 and 0.025 but are concentrated at lower occupancy levels between 10% to 40%. There is also a steady increase in density values over six hours. The sub-plot for cluster B displays observations with density levels reaching approximately 0.020 and occupancy levels concentrated between 30% to 75%. This sub-plot also reflects the concentration of occupancy data for cluster B in Figure 3.7d towards higher levels over the first few hours and a decrease in concentration

reflected by lower density over the latter half of the time period. The sub-plot for cluster C represents varied density and occupancy levels through the observed time period. This corresponds to the variation identified within the cluster C in Figure 3.7d and reflects the ability of the clustering approach to adequately represent the differences in the shapes of curves and the spread of occupancy values over time described in Figure 3.6.

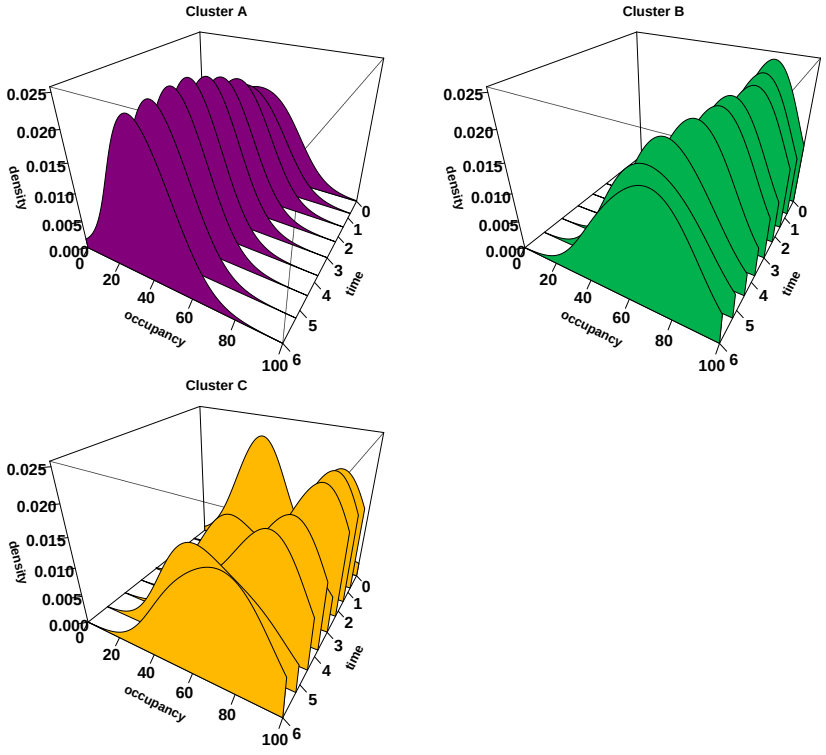


Figure 3.9: Three-dimensional density plots for distinct clusters determined using the functional distributional clustering algorithm.

Figure 3.10 summarises the distribution of occupancy observations for the three clusters A, B and C. The first boxplot, for cluster A, describes a range over lower levels of occupancies. The boxplots for cluster B and C are drawn over similar ranges of occupancies; cluster C does have a lower minimum occupancy level compared to cluster B. In addition, the boxplot for cluster C displays a skewed distribution as compared to the reasonably symmetric boxplot for cluster B.

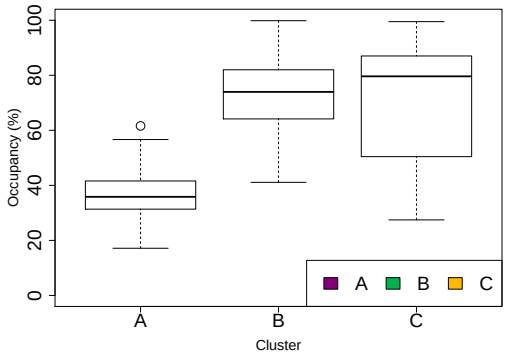


Figure 3.10: Boxplot that summarises the occupancy observations for three clusters A, B and C.

These results indicate the temporal patterns of simulated occupancy observations for the San Francisco grid style network on a weekday morning. Within the network, the traffic is simulated such that the initial traffic at the start of the time period is concentrated above Market street. More specifically, this is represented by the yellow and green cluster. The green cluster displays greater occupancy values in the network over the period of six hours. In the network, as displayed in Figure 3.11, the green cluster corresponds to areas both above (includes the area around the Embarcadero) and below the Market street divide (includes the East Cut and South Beach). Based on the geographical constraints of the network, the spread in traffic is expected towards the green cluster from the yellow cluster. The Financial district area is composed of both the yellow cluster and a part of the green cluster. This is reflected by the multiple density curves in the three dimensional plots displayed in Figure 3.9. The yellow cluster has higher occupancies (at values similar to the green cluster) between 8:00 am to 10:00 am; this typically indicates people's commute to work and resulting increase in vehicular traffic. The subsequent fall in occupancy before the spike in occupancy suggests the increase in traffic over lunchtime. This is unlike the region presented as the purple cluster that has fewer number of restaurants and retail areas and accordingly has far lower occcupancy levels.

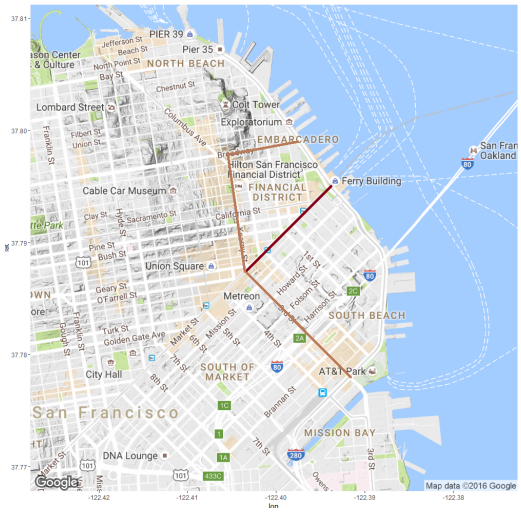


Figure 3.11: Downtown San Francisco Network with highlighted region of interest

3.4.3 Simulation study

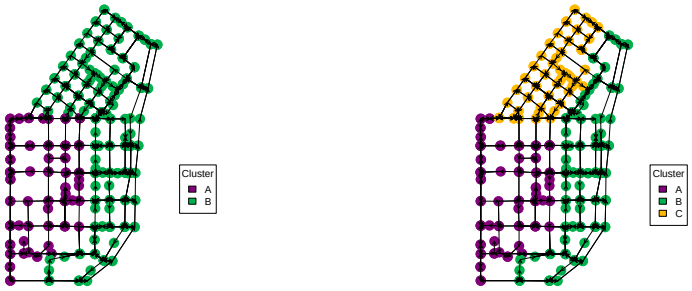
This section provides a quantitative analysis of the proposed functional distributional clustering algorithms to validate the clustering results in Section 3.4.2 for varied/various datasets. To this end, we simulated datasets as described in Section 3.4.1 with seeds from one to hundred to evaluate the developed algorithm's ability to identify clusters. The determined cluster structure is compared to the 'true' number of clusters as described in Figure 3.7a. For a given seed, the optimal number of clusters is determined using the defined clustering balance criterion. At the selected number of clusters, the ARI measures its agreement to the 'true' number of clusters. We average the ARI over all simulation results and present a comparison between the functional distributional algorithm, the functional only algorithm, and the distributional only algorithm. The mean and corresponding standard error of the ARI for all three algorithms are presented in Table 3.2. In addition, the 25th quantile, the median, and the 75th quantile of the determined optimal number of clusters are described for different algorithms.

	ARI		Number of Clusters		
Algorithm	Mean	SE	25th Q	50th Q	75th Q
Functional Distributional	0.85	0.174	3	3	4
Functional only	0.69	0.176	2	3	3
Distributional only	0.59	0.070	2	2	2

Table 3.2: Results aggregated over 100 simulations with varied seeds for the functional distributional clustering algorithm, functional only algorithm, and distributional only algorithm.

The functional distributional algorithm generates clusters that are reasonably similar to the defined 'true' clusters, as indicated by the aggregated ARI value equivalent to 0.85. The functional only algorithm has a lower mean ARI equivalent to 0.69 while the distributional only clustering algorithm struggles to identify three clusters with ARI equivalent to 0.59. This is reflected by the lower ARI and the suggested two optimal clusters. The functional distributional clustering method is also compared to other algorithms that utilise functional data analysis (FDA). The FDA-based approaches in this comparison study are implemented within a spatially adjusted agglomerative hierarchical clustering framework. In other words, the two-stage approach to clustering functional data is composed of a step using functional data analysis (FDA) techniques (Ramsay and Silverman, 2007) and a second step for the implementation of hierarchical clustering. More specifically, the study considers coefficients that describe the data, such that coefficients can be principal component scores resulting from principal component analysis (PCA) or coefficients from basis approximations (e.g., B-splines). In Figure 3.12a, the coefficients of basis expansions that represent functional data are utilised to define the distance and a clustering result shows two identified clusters. In Figure 3.12b, PCA within an agglomerative hierarchical clustering framework is implemented for spatio-temporal data to identify heterogeneity within the network. Figure 3.12b displays clustering output when principal component scores that explain highest percentage of variance are used within the clustering framework and the relevant clustering output has an ARI equal to 0.92.

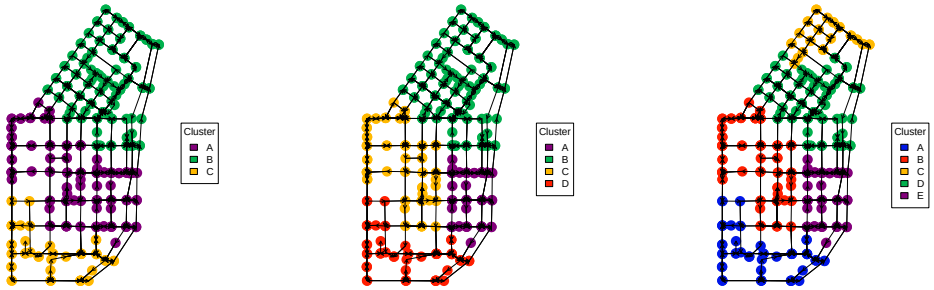
In Figure 3.13, we apply the ClustGeo (Chavent et al., 2017) method to the same simu-



(a) B-spline basis (b) Principal Component Analysis (PCA)

Figure 3.12: FDA bases approaches implemented within the agglomerative hierarchical clustering framework

lated dataset and set the parameter $\alpha = 0.99$ in order to encourage the incorporation of neighbourhood constraints. This Ward-like hierarchical clustering algorithm generates three clusters (in Figure 3.13a) and is evaluated against the true cluster structure to determine an ARI = 0.21. Similarly, an output with four clusters (in Figure 3.13b) leads to ARI = 0.33 and output with five clusters (in Figure 3.13c) leads to ARI = 0.30.



(a) (b) (c)

Figure 3.13: ClustGeo: Ward-like hierarchical clustering algorithm

We present simulation results for the FDA-based approaches and the ClustGeo in Table 3.3; the simulations are conducted exactly as mentioned earlier in this section. Table 3.3 presents the mean and corresponding standard error of the ARI for the B-spline based clustering approach, PCA based clustering approach and the ClustGeo method. The results of the B-spline based clustering approach indicate that it performs similar to the functional approach

results in Table 3.2. The PCA based clustering approach has performance that is nearly equal to the distributional approach.

	ARI		Number of Clusters		
Algorithm	Mean	SE	25th Q	50th Q	75th Q
Using coefficients of B-Spline basis	0.63	0.143	2	2	2
Using principal component (PC) scores	0.58	0.258	3	4	8
ClustGeo	0.32	0.023	4	4	5

Table 3.3: Results aggregated over 100 simulations with varied seeds for FDA-based clustering methods and the ClustGeo method.

3.5 Application

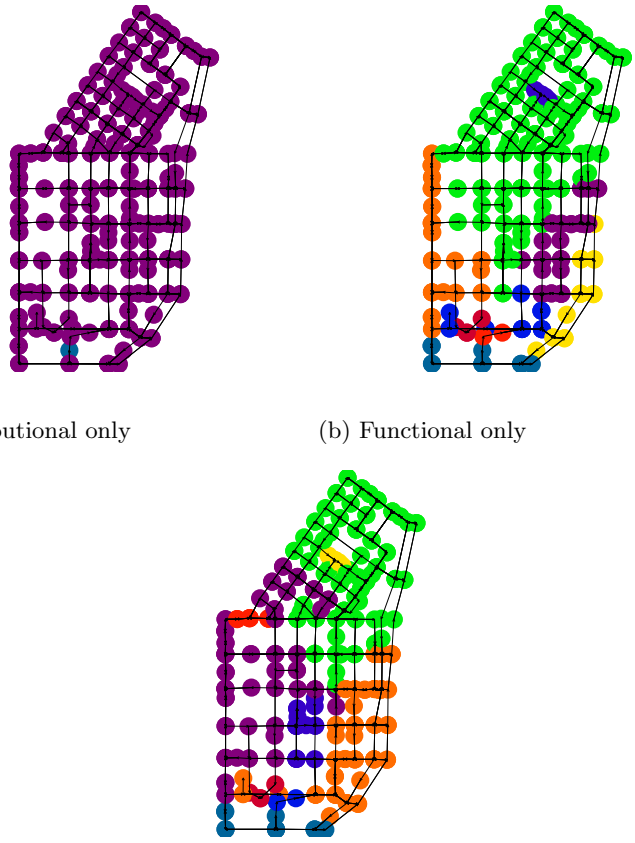
3.5.1 Occupancy data

To illustrate the functional distributional algorithm, we apply the developed clustering method to occupancy data generated for the 2.5 square miles network area in downtown San Francisco, CA. High resolution spatio-temporal data for urban road networks are not readily available in open data sources and so we use an AIMSUN microscopic traffic simulator to mimic relevant origin-destination traffic demand scenarios. These scenarios are simulated to broadly represent three different clusters. 120 observations are recorded over six hours (21600 seconds) with a sampling rate of 180 seconds and we seek to identify the differences in occupancy levels that reflect the spread of congestion across the network. Since data within the first two hours is limited to very low levels of occupancy across the network, the functional distributional algorithm is applied to 80 occupancy observations recorded between 10 am to 2 pm (14400 seconds). In general, it is expected that the traffic increases through the day with a peak in the concentration of occupancy around noon. This increase in occupancy could reflect the traffic at lunch time in the downtown area of San Francisco.

3.5.1.1 Results

In the described dataset, the underlying structure in the network for the 'true' number of clusters is unavailable and making assumptions of the partition structure is challenging. Rather, given the nature of occupancy data, we expect the functional distributional clustering algorithm would determine more relevant clusters than algorithms with differently defined distance measure scenarios. The clustering algorithm is implemented using the distance measure specified in Equation (3.5) and bandwidths are calculated using the extended cross validation method described in Section 3.3.2. Selected bandwidths for h_x and h_t are equivalent to 15 (occupancy in %) and 7.5 (time in seconds) and conditional functions are estimated over the sample of 80 occupancy observations. The clustering balance criterion suggests optimal number of clusters for the functional and distributional algorithm, functional only algorithm and distributional only algorithm. In Figure 3.14c, the functional distributional clustering algorithm partitions a network into nine clusters with three main clusters (green, purple, and orange). This is in contrast to the clusters obtained in Figures 3.14a and 3.14b, where the clustering balance criterion suggests a single larger cluster for the distributional only clustering algorithm and a main larger cluster along with several smaller clusters for the functional only clustering algorithm.

Figure 3.15 displays the corresponding density distributions for the clusters determined by the functional distributional clustering algorithm. Within a sub-plot for an individual cluster, Gaussian density curves (bandwidth equivalent to 15%) over relevant occupancy observations are displayed at defined time points (at 30 minute intervals) over the period of four hours (14400 seconds). This value of the bandwidth enables density curves to retain differences within each curve and allows for comparisons between clusters. Individual curves also describe the concentration of occupancy and their corresponding values between 0% and 100% through the day. The curves in the three-dimensional sub-plot for the green cluster have a higher magnitude in density levels as compared to the sub-plots for the orange and purple cluster. The sub-plot for the green cluster also displays variations in the concentration of occupancies and the range of occupancy values over four hours. The occupancy levels are concentrated at higher levels closer to mid-day before falling back to the



(a) Distributional only

(b) Functional only

(c) Functional and Distributional

Figure 3.14: Clustering results using micro-simulated data over four hours (14400 seconds).

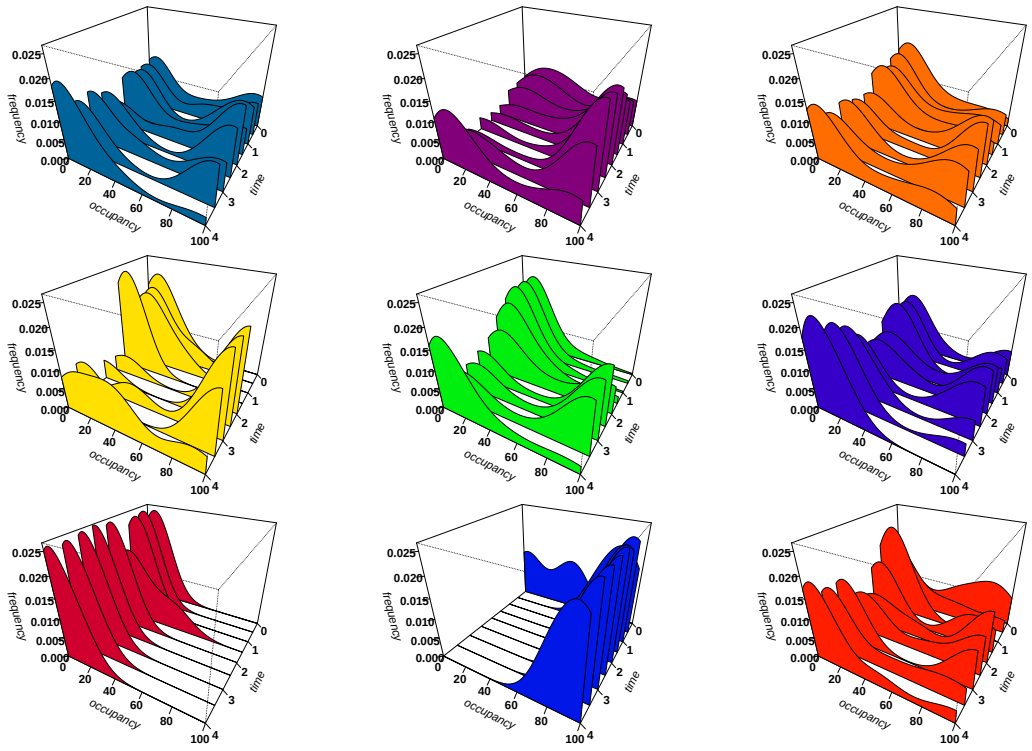


Figure 3.15: Three dimensional plots for the identified clusters in Figure 3.14c.

lower levels, as presented for earlier in the day. A similar set of variations can be viewed in the sub-plot for the yellow cluster but far more pronounced. The purple cluster has occupancy values that are concentrated at higher values for the majority of the time time period and the change in occupancy levels is not entirely concentrated around the middle of the day. On the other hand, the orange cluster has occupancy values that are concentrated far more equally at lower and higher values and with lower change in the distribution of density values through the period of four hours. The sub-plots on the third row display density curves and variations in occupancy levels that correspond to the smaller distinct clusters in the lower part of the network. In general, these plots are able to identify the expected overall traffic patterns within the San Francisco Network. The traffic is concentrated above the Market area in the morning with a spread in traffic to other part of the network through noon. The spread is gradual, reaching areas below the Market area and towards the east. The levels of occupancy steadily increase from 10:00 am (the start of this period from 10:00 am to 2:00 pm is marked as 0 in Figure 3.15), reach a peak around noon

and start to diminish by 2:00 pm. These differences in temporal patterns are differentiated by the functional distributional clustering method, as opposed to a static spatially-focussed clustering method.

3.6 Discussion

This chapter proposes a functional distributional clustering algorithm within an agglomerative hierarchical framework to identify spatially contiguous clusters using spatio-temporal data. The algorithm seeks to identify homogeneous regions within a heterogeneous network such that individual clusters reflect differences recorded in the readings of the sensor. In the traffic example studies, these clusters correspond to distinct temporal patterns in occupancy observations and congestion levels through the network. Within the framework of this clustering approach, the algorithm is both functional and distributional, such that a distance measure is defined utilising cumulative distribution functions, to account for temporal patterns present in the available data rather than summarised values. In this proposed non-parametric method, conditional CDFs are determined and stored outside individual iterations of the algorithm in order to improve the computational efficiency for larger datasets. This algorithm generates a hierarchy of clusters and decisions to estimate the optimal number of clusters in the network are dependent on defined methods. The simulation study demonstrates the superior ability of the functional distributional clustering algorithm in identifying 'true' clusters compared to the functional only, distributional only, FDA-based algorithms and a Ward-like hierarchical clustering method ClustGeo. We also applied this algorithm to real data to describe the clustering process when knowledge of the underlying 'true' clusters is limited. In general, the proposed method identifies spatially contiguous clusters that accommodate temporal patterns but do not change shape over time. In future work, we seek to extend the functional distributional clustering algorithm to be capable of identifying dynamic clusters and illustrate further applications.

Chapter 4

Binary dependent Chinese restaurant process

This chapter formally defines the binary dependent Chinese restaurant process (binDCRP) that is able to accommodate spatial constraints and encourage the discovery of connected segments. We utilise a modified version of this defined binDCRP as a prior in the flexible Bayesian clustering approach in Chapter 5. Section 4.1 formally describes the classical Chinese restaurant process (CRP) as a distribution over partitions. However, this process relies on the assumption of exchangeability, which can be unrealistic. Section 4.2 describes an alternative view of the Chinese restaurant process formulated to accommodate sequential data. The sequential CRP can be generalised to include the non-sequential case and this generalised view is called the distance dependent Chinese restaurant process (Blei and Frazier, 2011). This chapter also summarises the distance dependent Chinese restaurant process in Section 4.3; the ddCRP is capable of modelling random partitions of non-exchangeable data (both sequential and non-sequential). The clusters generated by the ddCRP are biased in nature such that each object is more likely to be clustered to data points that are identified as being nearer. Section 4.4 extends the non-sequential view of the ddCRP to accommodate the spatial constraints imposed by the geographical structure of the networks. This process is labelled the *binary dependent Chinese restaurant process*.

4.1 Chinese restaurant process (CRP)

The *Chinese restaurant process* (CRP) is a probability distribution over partitions (Pitman et al., 2002) described by specifying how a sample is drawn from it. The CRP is described using culinary metaphors and the restaurant is assumed to have an infinite number of tables with an infinite capacity to seat customers. Let a sequence of n customers, $\{1, \ldots, n\}$, enter the restaurant such that each customer sits at a randomly chosen table. A customer sits at an existing table k with probability proportional to the number of customers n_k already seated or at a new table with probability proportional to α. The real valued parameter α controls how often a customer sits at a new table. Formally, let z_i denote the table assignment for customer i and $z_{1:(i-1)}$ denotes customers already assigned to occupy K tables.

Definition 4.1.1 *The traditional CRP draws a table assignment z_i for customer i such that*

$$p(z_i = k \mid z_{1:(i-1)}, \alpha) \propto \begin{cases} \alpha \text{ for } k = K + 1 \\ n_k \text{ for } k \leqslant K \end{cases}$$

In Figure 4.1, the first customer is seated at Table 1 with probability $\frac{\alpha}{\alpha}$ and the second customer is seated with probability $\frac{1}{1+\alpha}$. This continues for subsequent customers and the table assignments of the customers $\{1, \ldots, 5\}$ are $z_1 = 1, z_2 = 1, z_3 = 2, z_4 = 1, z_5 = 2$. Each table in this framework has a dish that represents a combination of parameters.

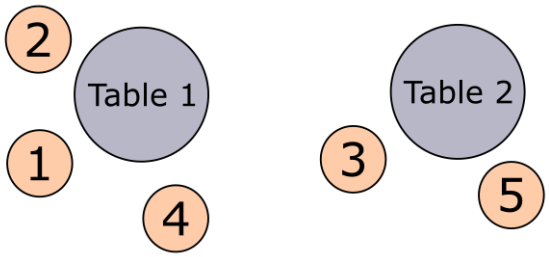


Figure 4.1: Chinese restaurant process (CRP)

To summarise, a potential partition structure of customers that enter and are seated at the restaurant is $\{\{1, 2, 4\}, \{3, 5\}\}$. The probability of seating customers in this order within the

restaurant is determined by:

$$\frac{\alpha}{\alpha} \cdot \frac{1}{1+\alpha} \cdot \frac{\alpha}{2+\alpha} \cdot \frac{2}{3+\alpha} \cdot \frac{1}{4+\alpha}$$

Formally, this can be defined for n customers that have been seated at K tables as:

$$\frac{\alpha^K \prod_{k=1}^{K} (n_k - 1)!}{\alpha \cdots (\alpha + n - 1)} \tag{4.1}$$

The probability of being seated in this order is proportional to $\alpha^K = \alpha^2$. In general, after n customers have been seated, the seating plan gives a partition of the customers over K tables. In this process, customers that enter the restaurant are identified with observations and observations associated with the same table belong to a cluster. The table selected to be seated at is chosen at random and this results in a process that lets the number of clusters be determined by the underlying data. The process is exchangeable, such that under any permutation of the ordering of customers, the probability of a specific partition of customers is invariant. However, this assumption of exchangeability is not reasonable for many clustering applications that rely on the ability to incorporate an underlying structure (e.g., spatial) or order in the data.

4.2 An alternative view of the CRP

An alternate approach to the CRP is called the *sequential* CRP and can also be used to describe a distribution over partitions. The sequential CRP is constructed with a dependence on the order in which customers arrive and a new customer chooses an existing customer as a friend to sit with. The sequential CRP can be viewed as a way to accommodate *temporal* constraints and tables are allocated as a deterministic function of the friendships between customers. Formally, let c_i be the ith customer assignment that denotes the customer with whom the ith customer is seated and $\mathbf{c}$ be the set of all customer assignments.

Definition 4.2.1 *The sequential CRP draws a customer assignment c_i as:*

$$p(c_i = j \mid \alpha) \propto \begin{cases} 1, \text{ if } j = 1, \ldots, i-1 \\ 0, \text{ if } j = i+1, \ldots, n \\ \alpha, \text{ if } j = i \end{cases}$$

In other words, the probability of choosing an existing customer as a friend is proportional to one and zero for choosing a customer yet to enter the restaurant. A customer can also choose to befriend themself with probability proportional to α. In Figure 4.2, a *link* from customer 4 to customer 1 indicates that customer 4 is friends with customer 1, but does not imply that customer 1 is also friends with customer 4. Customer 1 is the customer assignment for customer 4; customer 4 and customer 1 are seated together at table 1. A *self-link* at customer 1 indicates that the customer has chosen to befriend themself. Tables are allocated as a function of the friend allocations c_i. Let an induced table assignment be denoted as $z(c_i)$ and the set of all such table assignments is $z(\mathbf{c})$. The assignments are summarised as $c_1 = 1, c_2 = 1, c_3 = 3, c_4 = 1$ and $c_5 = 3$ and the induced table assignments $z(\mathbf{c})$ are $z(c_1) = 1, z(c_2) = 1, z(c_3) = 2, z(c_4) = 1, z(c_5) = 2$. A determined seating plan is represented by $\{\{1, 2, 4\}, \{3, 5\}\}$ and the probability of this arrangement is proportional to $\alpha \cdot 1 \cdot \alpha \cdot 1 \cdot 1 = \alpha^2$.

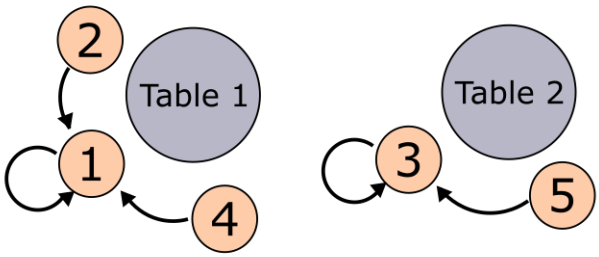


Figure 4.2: Sequential view of the Chinese restaurant process

The sequential CRP can recover the traditional CRP for the conditions described in definition (4.2.1). In the sequential CRP, the probability of being assigned to each of the other customers that are already seated at a table is proportional to one. Accordingly, the probability of sitting at a table is proportional to the number of customers already seated

there; this is the definition in a traditional CRP. In addition, the probability of a customer befriending themself and sitting at a new table is proportional to α. To summarise, the α parameter provides the ability to control the cluster size in both the sequential CRP and the traditional CRP.

The sequential CRP can also be generalised such that customers are seated with customers that enter the restaurant in any order. The generalised view includes both the sequential and the *non-sequential* case where customers have a choice between sitting by themselves, with customers that have yet to arrive and with customers that have already been seated. In other words, the seating plan probability is described in reference to customers befriending other customers without a focus on the order in which they arrive. This generalised view is called the distance dependent Chinese restaurant process.

4.3 Distance dependent Chinese restaurant process (ddCRP)

The *distance dependent Chinese restaurant process* (ddCRP) was first introduced by Blei and Frazier (2011) to accommodate non-exchangeable data. The probability of determining a seating plan in a restaurant is determined by friendships between customers and customers are eventually allocated to a table. In the ddCRP, a customer chooses another customer as a friend using a measure a_{ij} that deems customers i and j to be similar if they satisfy a notion of 'proximity'.

Definition 4.3.1 *The ddCRP draws a customer assignments c_i as:*

$$p(c_i = j \mid \alpha) \propto \begin{cases} a_{ij}, \text{ if } j \neq i \\ \alpha, \text{ if } j = i \end{cases}$$

The sequential CRP described in definition 4.2.1 is a specific type of the ddCRP where $a_{ij} = 1$ for $j < i$ and $a_{ij} = 0$ for $j > i$. In this special type of ddCRP, customers are said to be in 'proximity' if the order in which customers i and j are seated can be described as $j < i$. Accordingly, the measure $a_{ij} = 1$ deems customers i and j to be similar when i and j satisfy this notion of 'proximity'. The ddCRP defines a general notion of 'proximity' such that the

order between i and j is no longer relevant. For both $j > i$ or $j < i$, customers i and j are labelled as being in 'proximity'. Customers identified to be similar become friends and friendships connect the relevant customer. Accordingly, connected customers form a cluster.

The ddCRP prior is not conditioned on the seating of other customers; the representation of friendships between customers is used to allocate customers to tables. In other words, two customers that can reach other by traversing a sequence of customer assignments are allocated to the same table. The ddCRP determines a partition composed of connected customer assignments such that customers are connected to other customers. Since the number of occupied tables in a restaurant are random, the number of clusters are determined by the data. In Figure 4.3, friendships between customers 1, 2, and 4 and customers 3 and 5 are formed with probability proportional to $a_{ij} = 1$ and connected customers then sit at the two tables labelled as Table 1 and Table 2. Table 2 is a new cluster formed by customer 3 with probability proportional to α. To summarise, customer 1 is friends with customer 2 and 4, customer 2 is friends with customer 1, and customer 4 is friends with customer 1. Similarly, customer 5 is friends with customer 3 and customer 3 befriends themself.

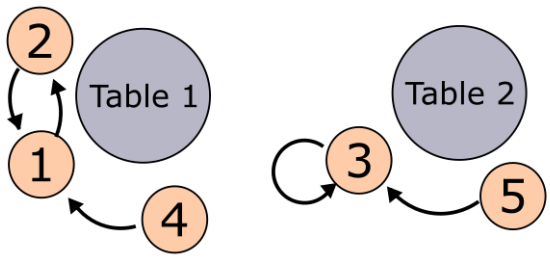


Figure 4.3: Distance dependent Chinese restaurant process (corresponds to a directed graph with out-degree equal to one, $deg^{+}(i) = 1$. This is discussed in Chapter 5).

Connected customers form their own table and new customers that enter the restaurant extend the number of connected customers. In general, multiple configurations of customers assignments might lead to the same table assignment and customer assignments can result in the formation of a cycle. For example, the partition structure $\{\{2, 1, 4\}, \{3, 5\}\}$ can be formed using the configuration displayed in Figure 4.3 as well as using other friendship

combinations between customers. Customer 1 sits with customer 2, customer 2 sits with customer 1 and a cycle is formed without either customers befriending themselves.

The assignments are summarised as $c_1 = 2, c_2 = 1, c_3 = 3, c_4 = 1, c_5 = 3$ and the induced table assignments $z(\mathbf{c})$ are $z(c_1) = 1, z(c_2) = 1, z(c_3) = 2, z(c_4) = 1, z(c_5) = 2$. The partition structure that corresponds to this seating plan is $\{\{1, 2, 4\}, \{3, 5\}\}$. The probability of determining such a configuration is proportional to $1 \cdot 1 \cdot \alpha \cdot 1 \cdot 1 = \alpha$. The determined probability indicates that the α parameter does not control the number of clusters in a ddCRP. The formation of the cycle between customer 1 and customer 2 is not formed with probability proportional to α; the α parameter only controls the formation of self-links and cannot lend effective control to the number of clusters. In contrast to the traditional and sequential CRP, the formation of a new cluster in a ddCRP is not dependent on a customer forming a self-link.

4.4 Binary dependent Chinese restaurant process (binDCRP)

The sequential view of the CRP is a special case of the ddCRP that is defined to accommodate temporal constraints. The generalised view of the ddCRP accommodates data regardless of their order and can also be re-written for spatial data. This process is labelled as the *binary dependent Chinese restaurant process* (binDCRP). The binDCRP is a special case that focusses on the non-sequential view of the ddCRP and customers are assigned to other customers identified as being similar if they are deemed to be in spatial 'proximity'.

Definition 4.4.1 *The binary distance dependent Chinese restaurant process (binDCRP) draws customer assignment c_i such that:*

$$p(c_i = j \mid \alpha) \propto \begin{cases} a_{ij} = 1, \textit{ if } j \sim i \\ a_{ij} = 0, \textit{ if } j \nsim i \\ \alpha, \textit{ if } i = j. \end{cases}$$

In other words, the customer assignment c_i is assigned to j with probability proportional to 1 if j is located in 'proximity' to i ($j \sim i$) and 0 if j is not located in 'proximity' to i

($j \nprec i$). A sense of similarity defined by a_{ij} is dependent on the nature of application and $a_{ij} = 1$ deems customers i and j to be similar if customer j is in 'proximity' to customer i. Similar customers become friends and connected customers belong to a common cluster. In Chapter 5, we describe the binDCRP for spatial data and define similarity between vertices identified to be in 'proximity' within a relevant binDCRP graph.

Chapter 5

Clustering: A nonparametric Bayesian model

5.1 Introduction

Spatial clustering methods seek to identify homogeneous regions in a heterogeneous network using recorded spatial data (e.g., junctions in an urban road network, areal units in a map). In the following sections, a non-parametric spatial clustering algorithm for a graph network is proposed within a Bayesian framework such that the primary concerns are to 1) identify *spatially contiguous homogeneous clusters* using spatio-temporal data, 2) fully account for spatial dependencies within determined clusters, 3) accommodate underlying *temporal patterns*, and 4) determine the number of clusters within the network in a data-driven manner. This holistic approach, implemented within a flexible Bayesian framework, seeks to identify mean-based differences in the temporal pattern and overcome several problems that are associated with the two-stage approach (e.g., functional distributional approach in Chapter 3) to spatial clustering. The functional distributional clustering approach generates a hierarchy of clusters, relies on additional methods to evaluate the number of clusters, is not based on formal statistical principles and is an algorithm motivated by distinct differences in both the mean and the variance over time. The flexible Bayesian model introduced in this Chapter is motivated by spatio-temporal datasets over a road network (e.g., the spread of occupancy in an urban grid style traffic network) and areal unit data (e.g, the change in

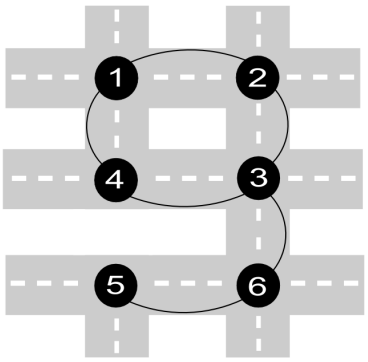
property prices for observations recorded over a map).

This non-parametric spatial clustering model introduced within a Bayesian framework is developed such that it satisfies the conditions listed above. Within this model, a binary dependent Chinese restaurant process (binDCRP) is placed as a prior to accommodate geographical constraints in the network. The binDCRP is developed as a special case of the ddCRP to accommodate the geographical constraints posed by the network. In order to fully account for the spatial correlation within individual suggested clusters, a conditional auto-regressive (CAR) model is utilised to incorporate neighbourhood relationships for vertices within a cluster; the temporal dependencies are accommodated using a first order auto-regressive (AR-1) model. This model assumes that observations follow a Gaussian distribution and adopts several Kronecker product identities to improve the computational efficiency of the sampler. We derive a relevant Metropolis within Gibbs sampler over this network to fully explore all possible clustering structures and infer the relevant parameters defined within the model.

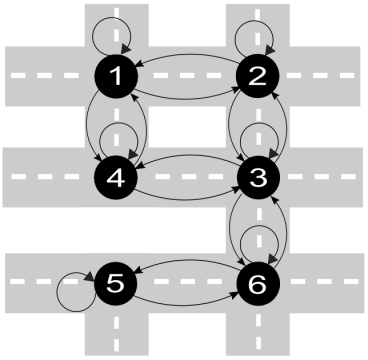
In Chapter 4, we formally introduced the binary dependent Chinese restaurant process (binDCRP) as a special case of the distance dependent Chinese restaurant process to accommodate spatial dependencies. The binDCRP was introduced in Chapter 4 within the framework of a restaurant using culinary references. Section 5.2 introduces spatial structures (both a road network and a network imposed over a map to represent areal unit data) and utilises this spatial framework to describe a binDCRP graph. In an urban road network, junctions represent vertices and road segments represent edges between vertices. A binDCRP graph is said to be connected if there is a path between every pair of vertices in the graph and each maximal connected subgraph of a binDCRP graph is a connected component. Connected components formed within the binDCRP graph provide a partition structure over the graph network. In Figure 5.1, four road networks are displayed to summarise the development of a binDCRP graph from an undirected graph to the eventual formation of a cluster structure in the binDCRP graph. Figure 5.1a displays an urban road network as an undirected graph, with junctions and road segments between the junctions.

This undirected graph has no loops and undirected edges are placed between the vertices in the graph. In Figure 5.1b, an undirected graph with the addition of loops can also be viewed as a directed graph with loops. All vertices have an edge to a neighbouring vertex (if a road is present) and directed edges are placed in both directions. In this graph, a loop is also placed at each vertex. In Figure 5.1c, the binDCRP graph is defined as a subgraph of the undirected graph with loops such that the out-degree of each vertex in the graph is equal to one. A cluster structure in the binDCRP is represented by connected components and the absence of an edge between vertices in the binDCRP graph leads to the formation of a new cluster. Figure 5.1c displays a single connected component within the binDCRP graph. An example of a cluster structure, formed within the binDCRP graph, with more than one connected component is displayed in 5.1d; many other cluster structures using the same set of vertices can also be formed within the framework of a binDCRP graph. In Figure 5.1d, each vertex has an edge to only one other vertex or a loop at a given vertex. Three clusters correspond to three connected components such that vertex 5 has no edges to other vertices in the network, vertex 6 has no edges to other vertices in the network and vertices 1, 2, 3 and 4 have no edges to the remaining vertices in the network.

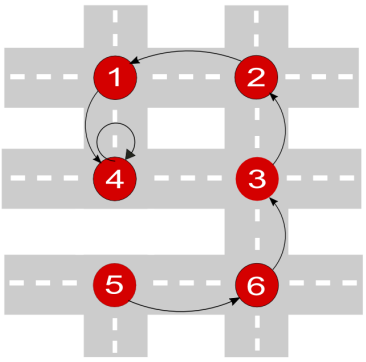
In Section 5.3, we introduce a modification of the binDCRP that allows for the number of clusters to be restricted. For example, a cluster structure formed using the modified binDCRP results in a reduction in the formation of singletons and clusters formed by cycles. The number of clusters formed in the binDCRP framework is equal to the number of singletons plus the number of cycles. In Section 5.4, the likelihood is defined using an assumption of normality over the observations recorded for each vertex and is computed within the defined binDCRP graph framework and the associated canCRP graph. The spatio-temporal precision matrix described within the definition of the likelihood is rewritten using Kronecker product identities to improve the computational efficiency of the model. In Section 5.5, we formally describe and discuss the computational implementation of the elements of the sampler in the framework of the binDCRP graph. The Metropolis within Gibbs sampler enables the sampler to explore all possible segmentations within the binDCRP graph framework and enables the inference of relevant parameters in the spatio-temporal precision



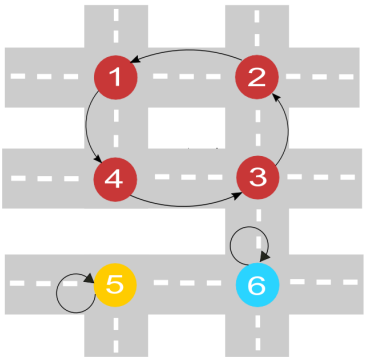
(a) Undirected graph (without loops)



(b) Undirected graph (viewed as directed)



(c) binDCRP graph



(d) Cluster structure in a binDCRP graph

Figure 5.1: Junctions in a road network

matrix.

To summarise, this chapter introduces the formal Bayesian model and describes the associated binDCRP prior, the defined likelihood, and the relevant sampler to explore all possible segmentations over the spatial structure. In Chapter 6, the performance of this model is illustrated by an application to a simulated dataset (with a known true cluster structure). Several applications to real world spatio-temporal datasets (both for an urban road network and areal unit data) are also discussed in Chapter 6.

5.2 Spatial framework

5.2.1 Notation

Let a graph $G = (V, E)$ be composed of a non-empty finite set of vertices $V = \{v_1 \dots v_n\}$ and a set of edges E that connect the vertices. An edge $e \in E$ between a vertex $v_i \in V$ and vertex $v_j \in V$ is denoted as (v_i, v_j) and if $(v_i, v_j) \in E$ then v_i and v_j are said to be *adjacent*. If an edge (v_i, v_j) connects vertex v_i to another vertex v_j, then the vertex v_i is said to be incident to the edge (v_i, v_j). Accordingly, the *neighbourhood* of a vertex v_i is determined by a set of adjacent vertices and the *degree* of a vertex in a graph denoted by $\deg(v_i)$ is defined as the number of edges that are incident to the vertex v_i. For an edge (v_i, v_j) in a graph, the vertex v_i is the *origin* and vertex v_j is labelled as the *terminus*. The *in-degree* of a vertex v_i, denoted as $\deg^-(v_i)$, is the number of edges with v_i as the terminus. The *out-degree* of a vertex v_i, denoted as $\deg^+(v_i)$, is the number of edges with v_i as the origin. A *loop* in a graph G is an edge (v_i, v_i) that has an origin and terminus at v_i and the loop is counted in both the out-degree and the in-degree. Spatial data for points in space arranged as a network can be represented as an undirected graph. We define an undirected graph (without loops) in Definition (5.2.1) and an undirected graph (with loops) in Definition (5.2.2), by using spatial data examples.

5.2.2 Undirected graph (without loops)

Definition 5.2.1 *An undirected graph (without loops) $G = (V, E)$ is defined such that*

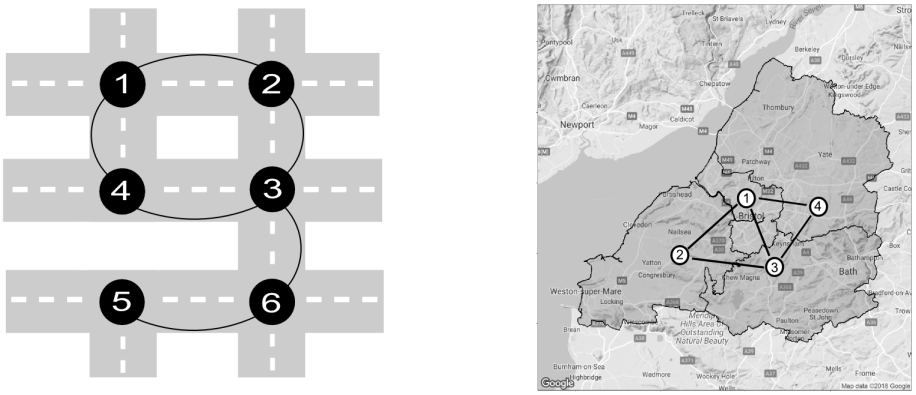
$$\forall\ v_i, v_j \in V, (v_i, v_j) \in E \Rightarrow (v_j, v_i) \in E.$$

In other words, a graph is defined as undirected if the edge relation between vertices v_i and v_j is symmetric and E is considered to be a set of unordered pairs. The directed relationship for an edge (v_i, v_j) is represented diagrammatically by an arrow from v_i to v_j, i.e., $v_i \rightsquigarrow v_j$. The edge (v_i, v_j) in an undirected graph represents edge (v_i, v_j) in direction $v_i \rightsquigarrow v_j$ and edge (v_j, v_i) in direction $v_j \rightsquigarrow v_i$. Since this relationship is symmetric, the edge is considered to be unordered.

For an undirected graph, let a $n \times n$ adjacency matrix $\mathbf{A}$ be defined for an undirected graph G without loops such that

$$A_{v_i v_j} = \begin{cases} 1 = A_{v_j v_i}, \text{ if } (v_i, v_j) \in E \\ 0, \text{ if } (v_i, v_j) \notin E \\ 0, \text{ if } v_i = v_j. \end{cases} \tag{5.1}$$

Figure 5.2a presents an undirected graph G in a road network using junctions and road segments between relevant junctions. In an urban road network, junctions represent vertices and road segments represent edges between vertices.



(a) Road Network (b) Map

Figure 5.2: Undirected graph (without loops)

In Figure 5.2a, the undirected graph G as a road network is composed of six vertices (junctions) $V = \{1, 2, 3, 4, 5, 6\}$ and a set of unordered pairs of vertices stored as edges (road segments) in E. Any edge (e.g., $(1, 2)$) in an undirected graph represents edges in both directions, i.e., $(1, 2)$ and $(2, 1)$. The neighbourhood of vertex 1 is $\{2, 4\}$ and the degree of vertex 1 is 2, i.e., out-degree, $\deg^+(1) = 2$, in-degree, $\deg^-(1) = 2$. The adjacency matrix $\mathbf{A}$ for G is defined as:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Figure 5.2b also displays a map with areal units translated to vertices in a graph network. Neighbouring regions that share a border in a map correspond to an edge between two vertices in an undirected graph.

A directed graph without loops is subject to the constraints of an edge relation, where $(v_i, v_j) \in E'$ is a distinct edge that does not also imply $(v_j, v_i) \in E'$. The edge relation can

be asymmetric and the pair of vertices are ordered. More formally,

$$G' = (V, E') \text{ such that } \forall\ v_i, v_j \in V \text{ and } (v_i, v_j) \in E', E' \subseteq E.$$

For example in the road network in Figure 5.2a, an equivalent directed graph is defined such that the edge (1, 2), vertex 1 $\rightsquigarrow$ vertex 2 and edge (2, 1), vertex 2 $\rightsquigarrow$ vertex 1 are two distinct edges.

5.2.3 Undirected graph (with loops)

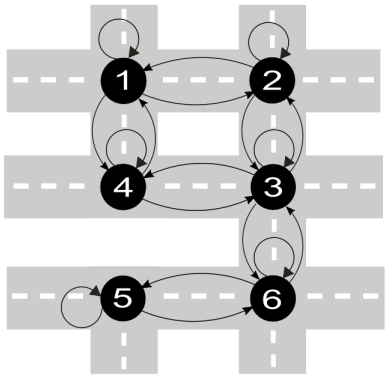
Definition 5.2.2 *For an undirected graph $G = (V, E)$, the set of edges with loops is defined by $\mathcal{L}(E) = E \cup \{(v_i, v_i) : v_i \in V\}$. An undirected graph can be viewed as a directed graph with loops and is defined as $G^* = (V, \mathcal{L}(E))$.*

The degree of loops is counted twice (e.g., $(1, 1) = (1, 1)$) for both the out-degree and the in-degree such that $\deg^+(v_i) = \deg^-(v_i) = 2$. For example, the degree of the loop at vertex 3 is $\deg^+(3) = 2$. In other words, such an undirected graph has an edge between vertices in both directions and a loop at each vertex through the graph.

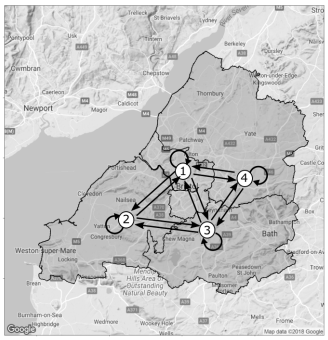
For an undirected graph with loops, let a $n \times n$ adjacency matrix $\mathbf{A}$ be defined such that

$$A_{v_i v_j} = \begin{cases} 1 = A_{v_j v_i}, \text{ if } (v_i, v_j) \in \mathcal{L}(E) \\ 0, \text{ if } (v_i, v_j) \notin \mathcal{L}(E) \\ 1, \text{ if } v_i = v_j. \end{cases} \tag{5.2}$$

Figure 5.3 displays undirected graphs (road network and a map) connected by edges and with loops at vertices.



(a) Road Network



(b) Map

Figure 5.3: Undirected graph from Figure 5.2 viewed as a directed graph with the addition of loops.

In addition to a loop at vertex 1, a vertex 1 also has edges in both directions to vertex 2 and vertex 4. This is represented by $1 \rightsquigarrow 2, 2 \rightsquigarrow 1, 1 \rightsquigarrow 4, 4 \rightsquigarrow 1$, and $1 \rightsquigarrow 1$. Equation (5.2) is used to define the adjacency matrix for the road network (undirected graph with loops) in Figure 5.3a:

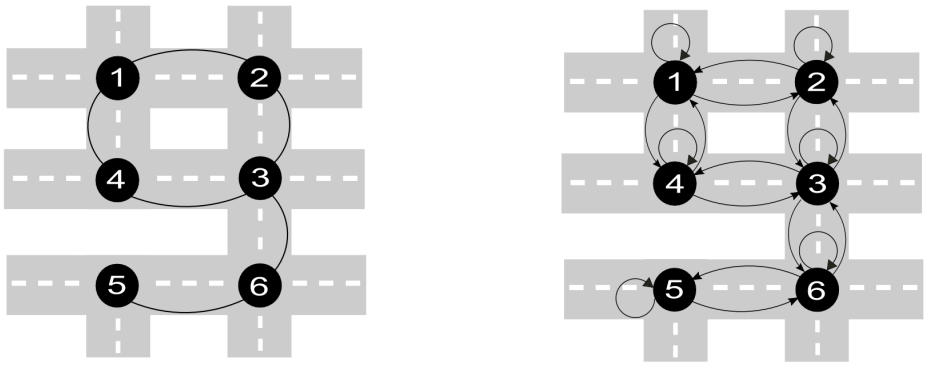
$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

A directed graph with loops need not necessarily have edges in both directions between adjacent vertices and a graph G'' is defined as

$$G'' = (V, E'') \text{ such that } \forall v_i, v_j \in V \text{ and } (v_i, v_j) \in E'',\ E'' \subset \mathcal{L}(E).$$

To summarise, Figure 5.4a displays an undirected graph $G = (V, E)$, such that E is a set of unordered pairs. An undirected graph (with loops) can also be viewed as a directed graph (with loops), such that an edge is present between vertices in both directions and a loop is present at each vertex in the graph. An undirected graph (with loops) is displayed in Figure 5.4b and is defined as $(G^* = V, \mathcal{L}(E))$, where the set of edges with loops is defined

by $\mathcal{L}(E)$. However, a directed graph with loops need not have edges in both directions for each vertex. A graph $G'' = (V, E'')$ is defined such that $E'' \subset \mathcal{L}(E)$.



(a) Undirected graph (without loops) (b) Undirected graph (viewed as directed)

Figure 5.4: Junctions in a road network

The undirected graph (with loops) and associated directed graphs (with loops) provide the framework for introducing the binDCRP graph. In Section 5.2.4, we introduce a special graph, where the edges are removed such that the out-degree of each vertex is one and refer to this special graph as the binDCRP graph.

5.2.4 Binary dependent Chinese restaurant process (binDCRP)

5.2.4.1 binDCRP graph

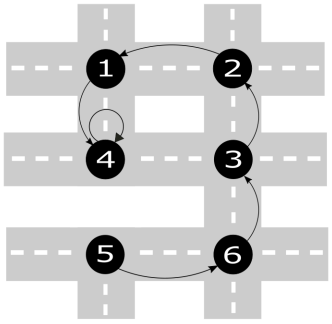
Let S be a graph composed of vertex set V and edge set E_b and the graph S is said to be a *subgraph* of the undirected graph with loops if $E_b \subseteq \mathcal{L}(E)$.

Definition 5.2.3 *The binDCRP graph S for a given problem is defined as:*

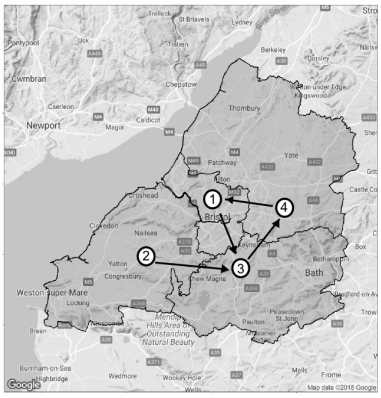
$$S = (V, E_b), E_b \subseteq \mathcal{L}(E) \text{ such that } deg^+(v_i) = 1, \forall v_i \in V.$$

To summarise, the binDCRP graph S is a subgraph of the undirected graph with loops and satisfies a condition that the out-degree of vertex $v_i \in V$ is equal to one, i.e., $deg^+(v_i) = 1$. In other words, each vertex $v_i \in V$ is allowed to have an edge to *only* one other vertex

$v_j \in V$ or to form a loop. Figure 5.5a and Figure 5.5b display examples of a binDCRP graph in a road network and in a map. Given the layout of the binDCRP graph in the road network and the map, a vertex is allowed to have a single edge to another vertex or to itself. The notion of 'proximity' (introduced for a restaurant framework in Chapter 4) in the road network is enforced by a road segment between two junctions in a road network. In other words, $v_i \sim v_j$ indicates that there is a road segment between the two vertices; this allows for an edge (v_i, v_j) to belong to the set of edges E_b. For example, vertex 1 is allowed to have an edge to vertex 4 or vertex 2 or a loop at vertex 1. There cannot be an edge between vertex 1 and vertex 3 due to the lack of a road segment, thus there is no edge in $\mathcal{L}(E)$. Similarly, vertex 5 is allowed to have an edge to vertex 6 or a loop at vertex 5 but there can be no edge to vertex 4. The displayed binDCRP graph is an example of a potential binDCRP graph for a given set of vertices. There can be multiple configurations of edges between vertices, leading to many possible binDCRP graphs.



(a) Road Network



(b) Map

Figure 5.5: An example of a binDCRP graph in a road network and in a map (Each vertex $v_i \in V$ in both the graph has $deg^+(v_i) = 1$)

Figure 5.6 displays an example where the required condition for a binDCRP graph is not satisfied, i.e., the $\deg^+(v_i) \neq 1$, $\forall v_i \in V$. At vertex 4, there is an edge to vertex 1 and an edge to vertex 3. This results in vertex 4 having out-degree greater than one, $\deg^+(v_i) = 2$.

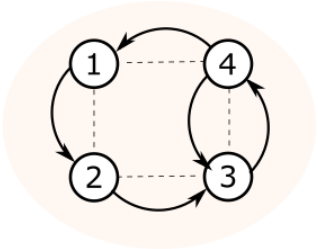


Figure 5.6: Does not qualify as a binDCRP graph

In the binDCRP graph, a *path* from v_1 to v_n is a sequence of adjacent edges $((v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n))$; the length of the path is equal to the number of edges which form the path. A vertex v_j is said to be *reachable* from a vertex v_i, if there is a path from v_i to v_j. In Figure 5.7, the path from vertex 1 to vertex 3 traverses edges $((1,2),(2,3),(3,3))$ and vertex 3 is said to be reachable from vertex 1. The path from vertex 4 to vertex 7 traverses edges $((4,5),(5,6),(6,7),(7,7))$ and vertex 7 is said to be reachable from vertex 4. Two vertices in a binDCRP graph are said to be *connected* if a vertex v_j is reachable from v_i or v_i is reachable from v_j. A binDCRP graph S is said to be connected if there is a path between every pair of vertices in the graph. Each maximal connected subgraph of a binDCRP graph S is a *connected component*. A connected component within a binDCRP graph represents a *cluster* and the binDCRP graph in Figure 5.7 is composed of two clusters.

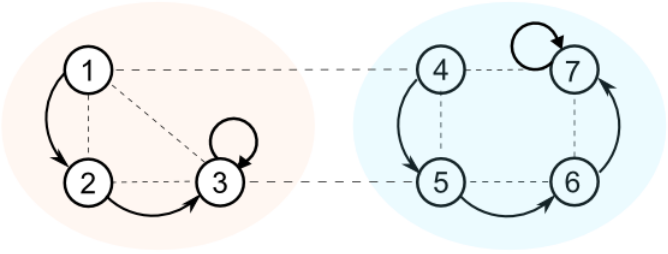


Figure 5.7: Two connected components of a binDCRP graph

In Figure 5.8, instead of a loop at vertex 3 there is an edge between vertex 3 and 5. This edge $(3,5)$ connects the path $((1,2),(2,3),(3,3))$ and $((4,5),(5,6),(6,7),(7,7))$ such that the binDCRP graph is composed of a single larger path between all the vertices in the

graph. In Figure 5.7, vertex 3 is reachable from vertex 1 and vertex 7 is reachable from vertex 4. However, vertex 7 is not reachable from vertex 1. In Figure 5.8, vertex 7 is to be reachable from vertex 1 and the binDCRP graph is composed of a single large cluster.

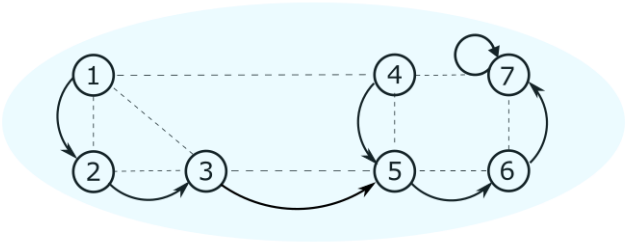


Figure 5.8: Single connected component of a binDCRP graph

5.3 Prior: binary dependent Chinese restaurant process (binDCRP)

The distance dependent Chinese restaurant process (ddCRP) was first introduced by Blei and Frazier (2011) to accommodate non-exchangeable data and Chapter 4 formally describes this process. In Chapter 4, we also introduced the binary dependent Chinese restaurant process (binDCRP) as a special case that focusses on the non-sequential case of the ddCRP. The binDCRP was introduced within a restaurant framework such that connections between customers in the restaurant correspond to the formation of a cluster. In this section, the restaurant based framework for both the ddCRP and the binDCRP is translated and adapted to the binDCRP graph.

The generative process of the ddCRP is described using vertex assignments for vertices in the binDCRP graph. The ddCRP draws a vertex assignment c_{v_i} using a general notion of 'proximity', $a_{v_i v_j}$. The distribution is determined by the probability of drawing a vertex assignment c_{v_i} for a given vertex v_i to connect the vertices (v_i and $c_{v_i} = v_j$) by an edge (v_i, v_j). A partition structure is composed of K clusters within a binDCRP graph and the set of all clusters determined by the binDCRP is denoted as $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$. For example, in Figure 5.7, two connected components correspond to a set of two clusters

$\mathcal{C} = \{C_1, C_2\}$. Let vertex assignments be denoted by $\mathbf{c}$ and the corresponding cluster representation be denoted by $z(\mathbf{c})$. Formally, the ddCRP adapted to the binDCRP graph is described as:

Definition 5.3.1 *The ddCRP draws a vertex assignment* c_{v_i}*:*

$$p(c_{v_i} = v_j) \propto \begin{cases} \alpha, \text{ if } v_i = v_j \\ a_{v_i v_j}, \text{ if } v_i \neq v_j \end{cases} \tag{5.3}$$

However, a special case of the ddCRP is introduced as the binary dependent Chinese restaurant process (binDCRP) to accommodate spatial dependencies. The general notion of proximity is adapted to ensure that only adjacent vertices are allowed to be drawn as a vertex assignment and this enforces a bias towards the formation of spatially contiguous clusters. In other words, edges between adjacent vertices form a connected component that correspond to a spatially contiguous cluster in the network. The binDCRP generates a clustering result represented by vertex assignments over the binDCRP graph and is described in the framework of the binDCRP graph. In Chapter 4, we defined this process looking at an informal description of the neighbourhood. We now restate the definition in a graph-theoretic context.

Definition 5.3.2 *The binDCRP draws a vertex assignment* c_{v_i}*:*

$$p(c_{v_i} = v_j) \propto \begin{cases} \alpha, \text{ if } v_i = v_j \\ a_{v_i v_j} = 1, \text{ if } v_i \sim v_j \\ a_{v_i v_j} = 0, \text{ if } v_i \nsim v_j \end{cases} \tag{5.4}$$

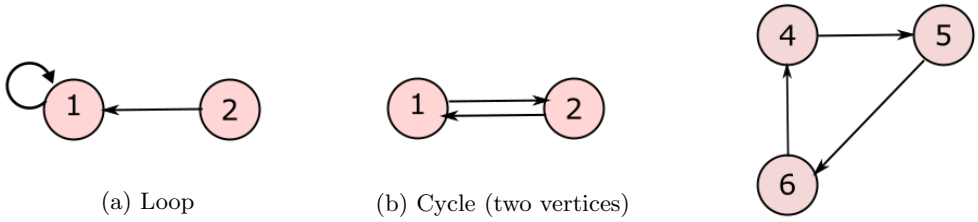
The defined similarity $a_{v_i v_j}$ for the binDCRP is equal to one for a vertex v_i that is allowed to have an edge to vertex v_j and zero when a vertex v_i is not allowed to have an edge to v_j. A condition $v_i \sim v_j$ indicates that a vertex v_i is allowed to be assigned to a vertex v_j such that $(v_i, v_j) \in E_b$; $v_i \nsim v_j$ indicates that this vertex assignment is not allowed. In the binDCRP, new clusters are formed by loops with probability proportional to α (as in definition (4.4.1)) but new clusters can also be formed by cycles between a set of vertices.

Definition 5.3.3 *A cycle in a binDCRP graph is a path* $((v_{i_1}, v_{i_2}), \ldots, (v_{i_n}, v_{i_1}))$ *that starts and ends at the same vertex* v_{i_1}.

A cycle specific to the binDCRP graph does not have any repeated edges but can have repeated vertices. An edge that connects a vertex to itself is called a *loop*. A pair of vertices v_i and v_j can also form a cycle in the binDCRP graph, $((v_i, v_j), (v_j, v_i))$. An edge (v_i, v_j) is said to be redundant given an edge $(v_j, v_i) \in E_b$ and (v_j, v_i) is said to be redundant given an edge $(v_i, v_j) \in E_b$. However, both (v_i, v_j) and (v_j, v_i) are not redundant at the same time.

Definition 5.3.4 *An edge* (v_i, v_j) *in the binDCRP graph* S *is said to be redundant if* $\forall v_i, v_j \in V$, *the removal of the edge* (v_i, v_j) *from a cycle results in a path that is not a cycle.*

Figure 5.9 displays scenarios that represent different configurations of edges between adjacent vertices. In Figure 5.9a, an edge $(1, 1)$ at vertex 1 forms a loop such that the out-degree of vertex 1 is one and a loop is the smallest possible cycle. In Figure 5.9b, edges (1, 2) and (2, 1) form a cycle. Given an edge $(1, 2)$, $1 \rightsquigarrow 2$ with $\deg^+(1) = 1$, the edge $(2, 1)$ is a redundant edge and vice versa. Figure 5.9c displays a cycle composed of three vertices; a cycle can be listed in any order such that $\{(4,5),(5,6),(6,4)\} = \{(5,6),(6,4),(4,5)\} = \{(6,4),(4,5),(5,6)\}$. For a cycle $\{(4,5),(5,6),(6,4)\}$, the removal of an edge $(6, 4)$ leads to a path $\{(4,5),(5,6)\}$ that is not a cycle and $(6, 4)$ is labelled as a redundant edge. This holds true for each edge in the cycle but cannot all be labelled as redundant edges.



(a) Loop (b) Cycle (two vertices) (c) Cycle

Figure 5.9: Vertex assignments and edges in a binDCRP graph

In a binDCRP, each cluster is formed by the presence of a loop at a vertex or by the presence of a cycle. The *number of clusters* is equal to the *number of loops + number of cycles*. In

Figure 5.8, the graph has a loop at vertex 7 which indicates a single cluster. In Figure 5.7, there is a loop at vertex 3 and at vertex 7 which indicates two clusters in the binDCRP graph. However, in Figure 5.10, the binDCRP graph has a loop at vertex 3, a cycle between vertex 4 and vertex 5 and a loop at vertex 7. This indicates the presence of three clusters, each shaded by a distinct colour.

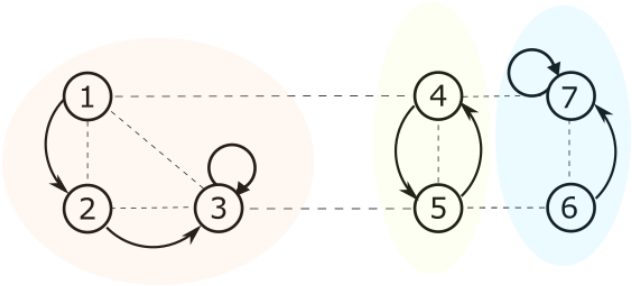


Figure 5.10: Three connected components of a binDCRP graph

In a traditional CRP and a sequential CRP, the α parameter controls the number of clusters by controlling the number of loops. However, as demonstrated in Figure 5.10, new clusters are also formed by cycles (e.g., ((4, 5), (5, 4))). This reduces the ability of a binDCRP to utilise the α parameter to control the number of clusters. The lack of control over the number of clusters is especially problematic in a neighbourhood based model where each vertex has edges to only a few number of vertices, leading to cycles being highly likely. We propose a modification to the binDCRP such that a probability of α is introduced to control both loops and redundant edges that lead to the formation of a cycle. This in turn enables the α parameter to control the number of clusters.

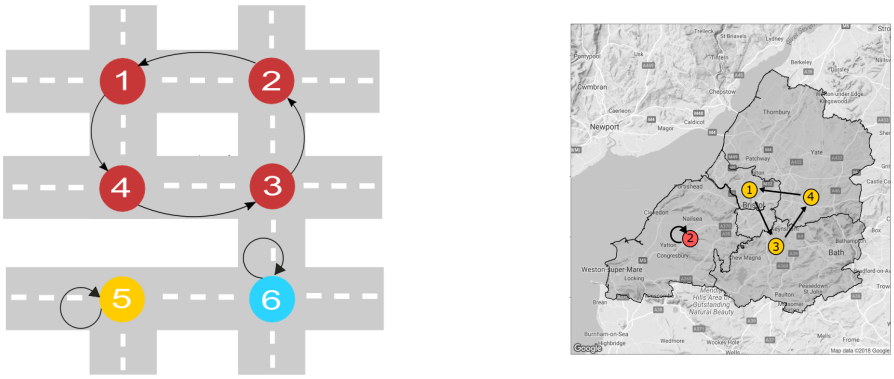
5.3.0.1 Modified binDCRP

Definition 5.3.5 *The modified binDCRP draws a vertex assignment* c_{v_i} *as:*

$$p(c_{v_i} = v_j) \propto \begin{cases} \alpha, \text{ if } v_i = v_j \text{ or if a cycle is formed when } (v_i, v_j) \text{ is added} \\ a_{v_i v_j} = 1, \text{ if } v_i \sim v_j \\ a_{v_i v_j} = 0, \text{ if } v_i \nsim v_j \end{cases}$$

In other words, for the modified binDCRP, the probability of drawing a vertex assignment v_j to connect vertex v_i by edge (v_i, v_j) is proportional to α if a cycle is formed by the addition of this edge to the derived binDCRP graph. A loop at a vertex v_i is also formed with probability proportional to α. A vertex v_i forms an edge to other vertices in 'proximity', with probability proportional to one, if a cycle is not formed by the addition of the relevant edge.

In Figure 5.11a, the road network is composed of three clusters (colored in blue, green and red). The loop at vertex 5 and at vertex 6 is formed with probability proportional to α. In the red cluster, the path from vertex 2 to vertex 3 is written as $((2,1),(1,4),(4,3))$. The addition of the edge $(3,2)$ to this path with probability proportional to α forms a cycle between vertices 1, 4, 3, and 2. The path and the resulting cycle can be formed in multiple ways. The set of vertices and relevant edges in each cluster represent three connected components and vertices shaded in the same color belong to a common cluster. The presence of two loops in the graph at vertex 5 and vertex 6 and a cycle between vertex 1, vertex 4, vertex 3 and vertex 2 suggests three clusters. Given a path composed of three edges, the addition of the fourth edge to the path forms a cycle with probability proportional to α. Accordingly, the probability of the partition structure $\{C_1, C_2, C_3\}$ being generated is proportional to α^3. Similarly, the map in Figure 5.11b is composed of two clusters (coloured in yellow and red).

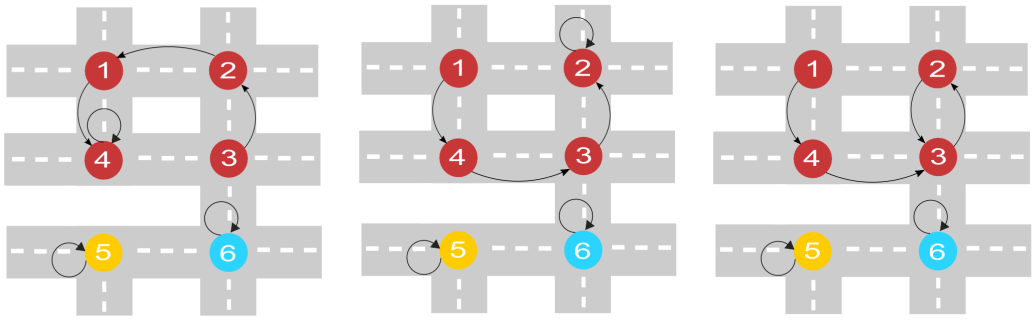


(a) Road Network

(b) Map

Figure 5.11: An example cluster structure in a binDCRP graph where vertex assignments for vertices are drawn by the modified binDCRP.

Figure 5.12 displays multiple scenarios that correspond to the cluster structure in Figure 5.11a. Figure 5.12a, Figure 5.12b and Figure 5.12c present binDCRP graphs that are composed of three clusters, with common vertices but different edges between the vertices. In Figure 5.12a, the path in the red cluster from vertex 3 to vertex 4 is $\{(3,2),(2,1),(1,4)\}$ and connects all the vertices in the binDCRP graph. Figure 5.12b and Figure 5.12c have different paths that connect all the vertices in the binDCRP graph. In Figure 5.12b and Figure 5.12c, the path in the red cluster from vertex 1 to vertex 2 is $\{(1,4),(4,3),(3,2)\}$. In Figure 5.12b, there is also a loop at vertex 2 that is not present at vertex 2 in Figure 5.12c.



(a) (b) (c)

Figure 5.12: Different configurations within a binDCRP graph that lead to the same cluster structure.

In general, a spatial pattern in a graph implies that vertices closer to one another are more similar than vertices located further away. In this section, we introduced a modified binD-CRP that is able to accommodate the geographical constraints imposed by the network and restrict the number of clusters. However, it is also reasonable to assume that vertices are spatially correlated to adjacent vertices within a cluster. We do not expect the modified binDCRP to be fully capable of accommodating spatial correlation within individual clusters. In the following section, additional dependencies posed by the structure of network are described in the context of the model.

5.4 Spatial clustering using spatio-temporal data

5.4.1 Data model

In this section, the likelihood of the observations is defined under the cluster structure suggested by the prior. Let the matrix of observations be denoted by $\mathbf{X}$, θ represent the set of parameters defined within the model and $\mathcal{C}$ be a set of clusters. A cluster structure is derived from the vertex assignments c_{v_i} of each vertex and the resulting connected components. Let a vertex assignment c_{v_i} for vertex v_i belong to vertex assignments denoted by $\mathbf{c}$ and let the clusters that result from the numerous vertex assignments be denoted by $z(\mathbf{c})$. For example, in Figure 5.11, let vertices 1, 2, 3 and 4 belong to cluster C_1. The relevant vertex assignments are $(4, 1, 2, 3)$. Accordingly, the vertex assignments $\mathbf{c}$ for the graph is $(4, 1, 2, 3, 5, 6)$, the induced cluster representations $z(\mathbf{c})$ is $(1, 1, 1, 1, 2, 3)$ and the set of clusters is $\mathcal{C} = \{C_1, C_2, C_3\}$.

The observations recorded over time for each vertex v_i is assumed to follow a Gaussian distribution and the likelihood term is represented by $p(\mathbf{X} \mid \mathcal{C}, \theta)$. The terms in the likelihood are decomposed as the following:

$$p(\mathbf{X} \mid \mathcal{C}, \theta) = \prod_{k=1}^{K} p(\mathbf{X}_{C_k} \mid \mathcal{C}, \theta) \tag{5.5}$$

Let K be the number of distinct clusters generated by the vertex assignments, the kth cluster be denoted by $C_k \in \mathcal{C}$ and let $\mathbf{X}_{C_k}$ be the matrix of observations that are allocated

to cluster C_k. The matrix $\mathbf{X}_{C_k}$ represents observations over time for all the vertices in the cluster C_k. (In the following sections, we do not use the factorised definition to help keep the notation simple.)

The binDCRP for a single observation at each vertex is defined as:

$$p(\mathbf{X} \mid \mathcal{C}, \theta) = \prod_{C \in \mathcal{C}} \int_{\mu} p(\mathbf{X}_C \mid \mu_C, \sigma^2 \mathbf{I}_{|C|}) p(\mu) d\mu \tag{5.6}$$

This is essentially a mixed model with random intercept, where the graph chooses the level each observation is assigned to and is equivalent to

$$p(\mathbf{X} \mid \mathcal{C}, \theta) \sim \mathcal{N}(\mathbf{X} \mid 0, \mathbf{\Omega}(\mathcal{C}, \theta)) \tag{5.7}$$

Within each cluster, it is reasonable to assume that vertices are spatially correlated to adjacent vertices and the spatial covariance matrix $\mathbf{\Omega}^{-1}$ is written as:

$$\mathbf{\Omega}_{ij}^{-1} = \begin{cases} \delta\sigma^2\mathbf{I}, \text{ if } v_i \text{ and } v_j \text{ are in the same cluster} \\ 0, \text{ otherwise} \end{cases} \tag{5.8}$$

This is an equi-correlation model that is unrealistic for spatial data. The next step is to assume a conditional auto-regressive (CAR) model in each cluster. This is the same as constructing a CAR model for the original graph network with edge set E and removing all edges to create a cluster boundary. The associated graph is referred to as the *conditional auto-regressive network for Chinese restaurant process* (canCRP) graph. In Figure 5.13, the formation of the canCRP graph is developed from an initial undirected graph without loops. In Figure 5.13a, the vertices are each connected to other adjacent vertices to represent a network. The binDCRP graph is defined within this context, such that the out-degree of each vertex is equal to one. In Figure 5.13b, two clusters are formed within this binDCRP graph such that the number of clusters is equal to the number of loops at vertices. The CAR model for the graph network in Figure 5.13a is developed by removing all edges to create a cluster boundary. In Figure 5.13c, the edges are removed from the network to create the clusters formed by the binDCRP graph framework. The Figure 5.13d

then displays the associated canCRP graph, where a CAR model is assumed in each cluster and the two clusters are shaded in different colours.

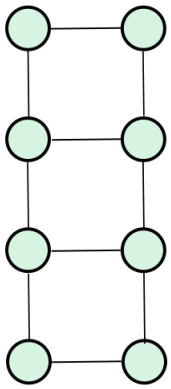
To summarise, the binDCRP is placed as a prior over the graph network and suggests the partition structure $\mathcal{C}$ composed of K clusters. We do not expect the binDCRP to fully accommodate the spatial dependencies within the network. More specifically, it is expected that the vertices within a cluster are spatially correlated with neighbouring vertices. In order to account for this 'within-cluster' correlation, we introduce a type of conditional auto-regressive (CAR) model within the conditional auto-regressive network for Chinese restaurant process (canCRP) graph. The canCRP is defined within the likelihood, unlike the binDCRP that is associated with the prior.

5.4.2 Conditional auto-regressive network for Chinese restaurant process (canCRP)

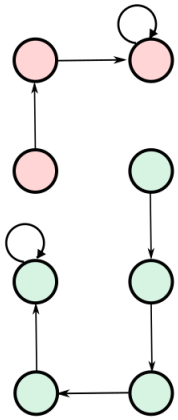
5.4.2.1 canCRP graph

Definition 5.4.1 *The canCRP graph denoted by S_c is defined within an identified cluster in the binDCRP graph S_b. Let $S_b = (V, E_b)$ be a binDCRP obtained for data from a network (V, E). Then the canCRP graph is $S_c = (V, E_c)$ with $(v_i, v_j) \in E_c$, if and only if $(v_i, v_j) \in E$ and there is a path from v_i to v_j or v_j to v_i in E_b.*

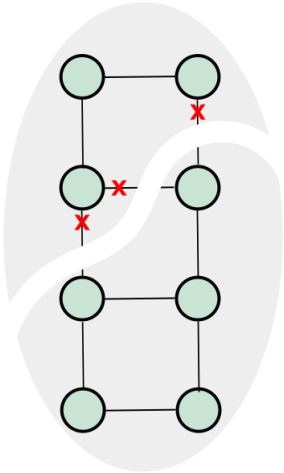
In other words, the adjacency matrix is restricted to edges that only connect vertices within the identified cluster. The adjacency matrix $\mathbf{A}_{S_c}$ for the canCRP graph is defined to include vertices and edges present within a given cluster and not the entire binDCRP graph. For example, in Figure 5.11a, a subgraph corresponding to the cluster shaded in red is composed of vertices $V_{S_c} = \{1, 2, 3, 4\}$ and edges $E_{S_c} = \{(1, 4), (4, 3), (3, 2), (2, 1)\}$ and the cluster



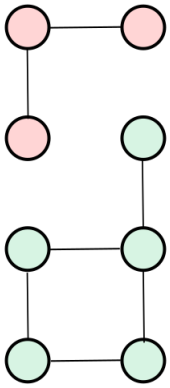
(a) Network



(b) binDCRP graph



(c) Split the network



(d) Two canCRP graphs

Figure 5.13: Conditional auto-regressive model in each cluster and associated graph

represents a canCRP graph. The adjacency matrix $\mathbf{A}_{S_c}$ of this canCRP graph is:

$$\mathbf{A}_{S_c} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The adjacency matrix and the relevant definition of the canCRP graph is utilised to define a conditional auto-regressive (CAR) model.

5.4.2.2 Conditional auto-regressive (CAR) model

Spatial dependence indicates that observations are more similar to neighbouring geographical units of the data than to those units that are further away. Numerous approaches have been adopted for modelling such spatial correlation, including simultaneous autoregressive models and conditional auto-regressive models. Conditional auto-regressive (CAR) models include intrinsic and convolution models (Besag et al., 1991), a model proposed by Cressie (1993) and a model proposed by Leroux et al. (2000). For a formal comparison of CAR models that describe these conditional auto-regressive models, see Lee (2011). In this chapter, we introduce a conditional auto-regressive (CAR) model that enables the model to incorporate information about neighbourhood relationships for vertices within suggested clusters. More specifically, we use a type of conditional auto-regressive model introduced by Leroux et al. (2000) to define the spatial precision matrix $\mathbf{\Omega}_S$ as

$$\mathbf{\Omega}_S = \rho(\text{diag}(\mathbf{A}_{S_{c++}}) - \mathbf{A}_{S_c}) + (1-\rho)\mathbf{I}. \tag{5.9}$$

The spatial precision matrix $\mathbf{\Omega}_S$ is defined over the adjacency matrix $\mathbf{A}_{S_c}$ in Equation (5.9) for the relevant canCRP graph S_c. Due to the nature of a grid style undirected graph, there are limited number of adjacent vertices (for example, each junction has upto four road segments to other junctions). In the context of such graphs, the precision matrices exhibit sparsity.

In the above definition of the spatial precision matrix $\mathbf{\Omega}_S$, the parameter ρ controls the correlation between adjacent junctions, diag($\mathbf{A}_{S_{c++}}$) is a diagonal matrix with elements equivalent to the row sums of the adjacency matrix $\mathbf{A}_{S_c}$ and $\mathbf{I}$ is an identity matrix.

5.4.3 First order auto-regressive model (AR-1)

The spatio-temporal dataset is composed of measurements for each vertex over time. An auto-regressive model is defined such that a value is regressed on previous values from the same series of measurements. In the first order auto-regressive (AR-1) model, exactly the first of the preceeding values in the series is used to predict the value at the present time. The first order auto-regressive model at X_{v_it} for vertex v_i at time t is given by

$$\mathbf{\Omega}_T = \rho X_{v_it-1} \tag{5.10}$$

The temporal dependencies can be easily accommodated in other ways such as the Matern covariance and the choice is not limited to a first order auto-regressive (AR-1) model.

To summarise, assume that an observation recorded for vertex v_i at time j, $X_{v_ij} = \mu_{v_ij} + \epsilon_{v_ij}$, such that $\mu_{v_ij} \sim$ CAR model and $\epsilon_{v_ij} \sim$ AR(1) model. Accordingly, we define a new spatio-temporal precision matrix $\mathbf{\Omega} = \mathbf{\Omega}_S \otimes \mathbf{\Omega}_T$ that accounts for both spatial and temporal dependencies within the suggested clusters.

5.4.4 Likelihood: Data model

In this section, the likelihood is defined to account for observations recorded over both space and time. The observations are assumed to follow a Guassian distribution such that $p(\mathbf{X} \mid \mathcal{C}, \theta) \sim N(0, \mathbf{\Omega}(\mathcal{C}, \theta))$ and the log-likelihood is defined as:

$$\mathcal{L} = \ln(p(\mathbf{X} \mid \mathcal{C}, \theta)) = -\frac{nN}{2}\ln(2\pi) - 0.5\ln|\sigma^2\mathbf{I} + \tau^2\mathbf{\Omega}_S \otimes \mathbf{\Omega}_T| - 0.5\text{vec}(\mathbf{X})^T[\sigma^2\mathbf{I} + \tau^2\mathbf{\Omega}_S \otimes \mathbf{\Omega}_T]^{-1}\text{vec}(\mathbf{X}), \tag{5.11}$$

The parameter θ represents λ (the ratio of parameters $\frac{\tau^2}{\sigma^2}$), ϕ (from the temporal precision matrix) and ρ (spatial precision matrix). The definition of the likelihood utilises

the suggested cluster structure $\mathcal{C}$ and includes the defined spatio-temporal precision matrix $\mathbf{\Omega} = \mathbf{\Omega}_S \otimes \mathbf{\Omega}_T$. A spatio-temporal precision matrix $\mathbf{\Omega}$ defined over larger graphs and greater time periods can result in matrices that are difficult to invert. We exploit multiple identities that enable us to rewrite different terms in the likelihood and improve the computational efficiency of the clustering model. The presence of a unique observation for every space and time combination allows for these identities to be exploited.

5.4.4.1 Kronecker product identities

Identity I: A matrix product with a Kronecker product is written in terms of ordinary matrix products.

$$(\mathbf{\Omega}_S^T \otimes \mathbf{\Omega}_T)\text{vec}(\mathbf{X}) = \text{vec}(\mathbf{\Omega}_T \mathbf{X} \mathbf{\Omega}_S) \tag{5.12}$$

Identity II: Kronecker product plus a diagonal term is expressed using the eigenvalue decomposition for the precision matrices such that $\mathbf{\Omega}_S = \mathbf{\Gamma}_S \mathbf{\Lambda}_S \mathbf{\Gamma}_S^T$ and $\mathbf{\Omega}_T = \mathbf{\Gamma}_T \mathbf{\Lambda}_T \mathbf{\Gamma}_T^T$. In this identity, $\mathbf{\Gamma}_T$ represents a matrix of the eigenvectors of $\mathbf{\Omega}_T$ and $\mathbf{\Lambda}_T$ represents a diagonal matrix of the eigenvalues of $\mathbf{\Omega}_T$.

$$(\sigma^2 \mathbf{I} + \tau^2 \mathbf{\Omega}_S \otimes \mathbf{\Omega}_T) = (\mathbf{\Gamma}_S \otimes \mathbf{\Gamma}_T)(\sigma^2 \mathbf{I} + \tau^2 \mathbf{\Lambda}_S \otimes \mathbf{\Lambda}_T)(\mathbf{\Gamma}_S^T \otimes \mathbf{\Gamma}_T^T) \tag{5.13}$$

The identities introduced above are utilised to rewrite the log of the defined likelihood:

$$\begin{aligned}\mathcal{L} = -\frac{n \cdot N}{2} \ln(2\pi) - 0.5 \sum_{i=1}^{n} \sum_{j=1}^{N} \ln |\sigma^2 \mathbf{I} + \tau^2 \mathbf{\Lambda}_T[n,n] \cdot \mathbf{\Lambda}_S[N,N]| \\ - \frac{1}{2} \text{vec}(\mathbf{\Gamma}_T^T \mathbf{X} \mathbf{\Gamma}_S)^T (\sigma^2 \mathbf{I} + \tau^2 \mathbf{\Lambda}_S \otimes \mathbf{\Lambda}_T)^{-1} \text{vec}(\mathbf{\Gamma}_T^T \mathbf{X} \mathbf{\Gamma}_S)\end{aligned} \tag{5.14}$$

The terms in Equation (5.11) can be evaluated using these Kronecker product identities for an efficient solution. Accordingly, the complexity of computing the terms in the likelihood is reduced from $O(n^3 N^3)$ to $O(n^2 N + N^2 n + n^3 + N^3)$.

For example, let a spatio-temporal dataset represent 360 observations over time and 158 vertices arranged as a graph network. More specifically, $\dim(\mathbf{\Omega}_S) = 158 \times 158$, $\dim(\mathbf{\Omega}_T) =$

360×360 and this leads to $\dim(\mathbf{\Omega}_S \otimes \mathbf{\Omega}_T) = 56880 \times 56880$. The spatio-temporal precision matrix defined as a Kronecker product results in a very large matrix and is difficult to invert. The dimension of the matrix $\mathbf{X}$ is $\dim(\mathbf{X}) = 360 \times 158$ and the number of observations in $\text{vec}(\mathbf{X})$ is 56880. The Kronecker product identity I helps to avoid computing the Kronecker product $\mathbf{\Omega}_S \otimes \mathbf{\Omega}_T$, a matrix of dimension 56880×56880. Instead, $\text{vec}(\mathbf{\Omega}_T \mathbf{X} \mathbf{\Omega}_S)$ results in a vector composed of 56880 observations.

Using the defined likelihood in Equation (5.14) and the binDCRP prior (Equation (5.3.5)) placed over the spatio-temporal dataset $\mathbf{X}$, the posterior is defined as

$$p(c_{v_1:v_N} \mid \mathbf{X}, \theta) \propto \prod p(c_{v_i}) p(\mathbf{X} \mid \mathcal{C}, \theta), \tag{5.15}$$

where the cluster representation is derived from the assignments of vertices to other vertices. The prior term uses the assignments between vertices to ensure that the adjacency and proximity structure of individual vertices are accounted for. The likelihood uses the eventual allocation of clusters to the vertices in the graph. $p(\mathbf{X} \mid \mathcal{C}, \theta)$ represents the likelihood that is conditional on the allocation of clusters and the log over the defined likelihood is utilised for computational purposes.

5.5 Posterior inference

The posterior generates partitions for the graph using the relevant data model and the binDCRP prior. However, the binDCRP places a prior over the combinatorial number of all possible configurations of vertices and their relevant assignments. This results in the posterior being intractable (i.e., difficult to directly evaluate). Instead, the algorithm employs a Markov chain Monte Carlo (MCMC) inference method, a Metropolis within Gibbs sampler to evaluate the posterior. The parameters defined within the spatio-temporal precision matrix are learnt by proposing Metropolis-Hastings updates. For generating the clustering distribution, the model employs a Gibbs sampling scheme over a graph and possible edges between vertices are explored by replacing an edge at random at each step. The Markov chain is defined by iteratively sampling each vertex assignment c_{v_i} for vertex

v_i, conditioned on the remaining vertex assignments $\mathbf{c}_{-v_i}$ and data $\mathbf{X}$.

$$p(c_{v_i} \mid \mathbf{c}_{-v_i}, \mathbf{X}, \theta) \propto p(c_{v_i}) p(\mathbf{X} \mid \mathcal{C}, \theta) \tag{5.16}$$

The prior $p(c_{v_i})$ is defined by Definition (5.3.5) and the likelihood is defined for spatio-temporal data (Section 5.4). Let $z(\mathbf{c}_{-v_i})$ denote the cluster representation when the vertex assignment for vertex v_i is removed and $z(\mathbf{c})$ denote the cluster representation when all vertices have a relevant vertex assignment (as specified in the binDCRP graph). The sampling from Equation (5.16) is a two-stage process. The sampler first removes a vertex assignment from the existing structure and then considers the probability of new vertex assignments when replaced and its effects on the likelihood term. More specifically, in the first stage, the sampler removes the vertex assignment c_{v_i} from the current configuration of vertices and edges between vertices. In the next stage, the prior probability of each possible value of c_{v_i} is determined and its effect on the likelihood term is examined. This is denoted as moving from $p(\mathbf{X} \mid z(\mathbf{c}_{-v_i}))$ to $p(\mathbf{X} \mid z(\mathbf{c}))$. In the first stage, the removal of the vertex assignment c_{v_i} either retains the cluster structure, i.e., $z(\mathbf{c}^{old}) = z(\mathbf{c}_{-v_i})$ or splits the cluster associated with v_i into two new clusters. After the vertex assignment has been removed at v_i, the second stage is concerned with the reassignment of the vertex assignment. The random vertex assignment in the graph either leaves the cluster structure intact, i.e., $z(\mathbf{c}_{-v_i}) = z(\mathbf{c})$ or joins the cluster of vertex v_i to a vertex in a different cluster. This enables the sampler to explore the space of all possible cluster structures within the graph.

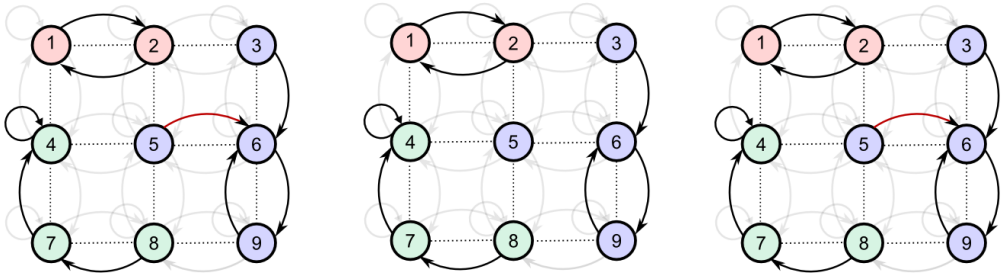
Let two clusters C_1 and C_2 represent the connected components formed by vertex assignments. This representation is used to describe the change in cluster configuration and the resulting change in the associated likelihood definition when a vertex from cluster C_1 is assigned to cluster C_2. The indices C_1 and C_2 are joined to form cluster C_3. Formally, to represent the sampler, we first remove a vertex assignment c_{v_i} of vertex v_i that can potentially split the cluster. The likelihood remains the same for scenarios where the vertex reassignment does not lead to a new cluster. Equation (5.17) describes a scenario where the vertex reassignment leads to the formation of a new cluster structure such that $z(\mathbf{c}_{-v_i}) \neq z(\mathbf{c})$.

$$p(c_{v_i} \mid \mathbf{c}_{-v_i}, \mathbf{X}, \theta) \propto \begin{cases} p(c_{v_i})\gamma(\mathbf{X}, \mathcal{C}, \theta) \text{ if } c_{v_i} \text{ joins } C_1 \text{ and } C_2 \text{ to form } C_3 \\ p(c_{v_i}) \text{ otherwise,} \end{cases} \tag{5.17}$$

$$\text{where } \gamma(\mathbf{X}, \mathcal{C}, \theta) = \frac{p(\mathbf{X}_{C_3} \mid \theta)}{p(\mathbf{X}_{C_1} \mid \theta)p(\mathbf{X}_{C_2} \mid \theta)}$$

5.5.1 Example scenario

In order to demonstrate the implementation of the sampler, we construct an example cluster structure within a binDCRP graph and utilise these structures to construct two different scenarios. In the first scenario, the initial cluster structure is displayed in Figure 5.14a and the edge from vertex 5 to vertex 6 is sampled. The edge $(5, 6)$ is removed from the set of edges such that the sampler searches for a new vertex assignment. The edge is then replaced from vertex 5 such that the partition structure is returned to the original allocation of clusters. In this scenario, v_i is 5, c_{v_i} is 6, $\mathbf{c}_{-v_i}$ represents all vertex assignments except $\mathbf{c}_{-v_i} = 6$ and $z(\mathbf{c}^{old}) = z(\mathbf{c}_{-v_i}) = z(\mathbf{c})$. In Figure 5.14a, the initial partition structure and structure after replacement is $\{\{1, 2\}, \{3, 5, 6, 9\}, \{4, 7, 8\}\}$ (displayed in Figure 5.14a and Figure 5.14c). This is an example of a scenario that does not lead to a change in the partition structure.

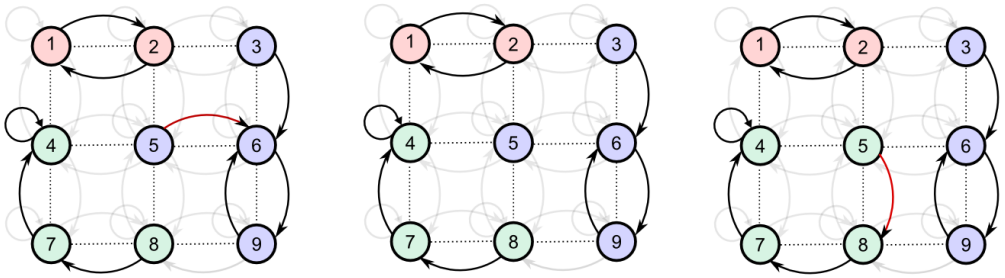


(a) Sample an edge (b) Remove the sampled edge (c) Replace an edge

Figure 5.14: First scenario: No change in cluster structure

In the second scenario, the initial partition structure $\mathcal{C}$ is composed of $\{C_1, C_2, C_3\} = \{\{1, 2\}, \{3, 5, 6, 9\}, \{4, 7, 8\}\}$. The edge (5, 6) is sampled and removed. The replacement of

the edge (5, 6) by edge (5, 8) leads to vertex 5 joining a new cluster. In this scenario, v_i is 5, c_{v_i} is 6, $\mathbf{c}_{-v_i}$ represents all vertex assignments except $\mathbf{c}_{v_i} = 6$ and $z(\mathbf{c}^{old}) = z(\mathbf{c}_{-v_i}) \neq z(\mathbf{c})$. Here, the likelihood is computed using Equation (5.17). In Figure 5.15, the partition structure after replacement is $\{\{1, 2\}, \{3, 6, 9\}, \{5, 8, 7, 4\}\}$. This is one example of a scenario where the reassignment leads to a change in the partition structure.



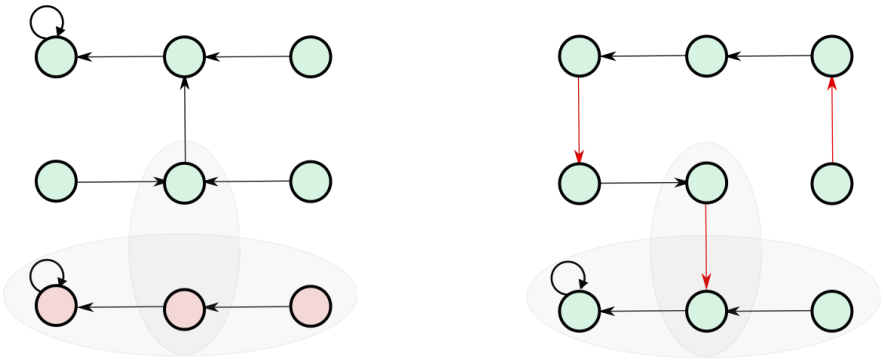
(a) (b) (c)

Figure 5.15: Second scenario: Change in cluster structure

This sampler is described using two scenarios, where the partition structure remains constant after sampling and where the partition structure is modified after sampling. In the implementation of this sampler, unless the vertex assignments result in a change in the cluster structure, cached computations of previous iterations are utilised.

5.5.2 Implementation

The sampler seeks to explore all possible cluster structures within a given graph structure and infer the relevant parameters (ρ, ϕ, λ) within the model. A change in cluster structure would require the adjacency matrix and the neighbourhood structure to be updated to reflect the new cluster structure. For example, in Figure 5.16a, the graph is composed of two connected components; each connected component has one loop. For this graph to become a single cluster multiple changes are required. In Figure 5.16b, a graph composed of a single connected components over the same vertices is displayed. The rewiring of edges in the graph changes the cluster structure and the edges that are changed to create this new structure are highlighted in red. Each of these updates need to be recorded in an efficient manner.



(a) Two clusters

(b) One cluster

Figure 5.16: Rewiring to form a new cluster structure in a binDCRP graph framework

The sampler is dependent on the ability to efficiently traverse the graph structure; for this purpose a *flood fill* search algorithm is implemented. A breadth first search is utilised to reach adjacent vertices and other connected vertices. The nature of the graph with a limited number of adjacent vertices makes a breadth first search more appropriate than a depth first search and demonstrates better performance. Let the set of neighbours for a vertex v_i be denoted by $\mathbf{g} = \{v_1, v_2, \ldots v_g\}$. For example, in Figure 5.17, vertex assignments $\mathbf{c}$ is $(7, 4, 4, 5, 4, 5, 6, 5, 8, 9, 11)$ and the set of neighbours for vertex v_i, say at vertex 4 is $\{2, 3, 5, 11\}$.

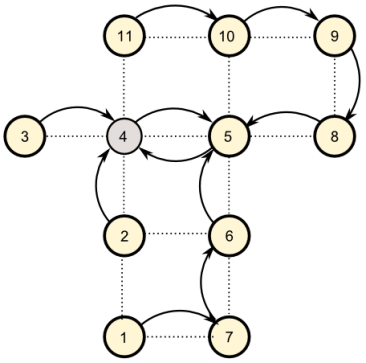


Figure 5.17: Graph traversal

More specifically, in the implementation of the sampler, a search algorithm is utilised to identify connected components (within one level) for each vertex in the graph. The algo-

rithm starts at a vertex v_i, visits the neighbours identified by $\mathbf{g}$ and searches for a vertex within the connected component. This breadth first search is adapted within the framework of the binDCRP graph. This search algorithm marks all reachable vertices and determines a path within the connected component. In other words, this search algorithm enables the identification of the adjacent vertices and all edges connecting the neighbouring vertices to other vertices.

In Figure 5.18, another traversal through the graph is displayed. The parent vertex is shaded in grey and each vertex has a vertex assignment to an adjacent vertex. The result for the vertex coloured in grey in Figure 5.17 is such that all the vertices are marked with a value of 1 and coloured in yellow. In Figure 5.18, the initial level of connected components are colored in yellow and all vertices visited as the second level of connected components are shaded in green. This is unlike Figure 5.17, where a single level of connected components exist and all the vertices are shaded in yellow.

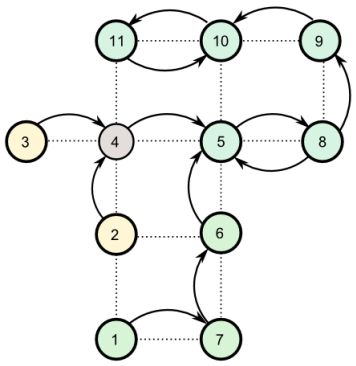


Figure 5.18: Example of a flood fill search

For example, in Figure 5.18, for the vertex 4, the flood fill traverses the graph using a breadth first search to identify adjacent vertices. This determines a result such that vertices 3, 4 and 2 are marked with a value of 1 and 5, 8, 9, 10, 11, 1, 6 and 7 are marked with a value of 2. This search algorithm helps the sampler to monitor changes in the cluster structure, associated edges for a vertex v_i and changes in the vertex assignments. The following two algorithms enable the updates of vertex assignments, clusters and adjacency matrices of the network to be carried out within the sampler. In Algorithm 4, clusters $\mathcal{C}$, vertex assignments

c and the adjacency matrix **A** for the graph are updated within the binDCRP framework. More specifically, the adjacency matrix is updated for the vertex v_i and the corresponding neighbours **g**. A vertex v_i is first sampled at random from the graph and a set of neighbours **g** is defined for vertex v_i. In order to be able to identify the existing connected structure, a breadth-first search is utilised and the clusters are updated. The spatial precision matrix $\mathbf{\Omega}$ is computed over the updated cluster structure and the log-likelihood is defined using Equation (5.11) over the new spatio-temporal precision matrix. A vertex assignment is sampled using the probability computed for the binDCRP as in Definition (5.3.5). The adjacency matrix with the resampled edge and the vertex assignments are updated to store the new relevant structure.

Algorithm 4: Updates edges with possible cluster change, $U_{\mathcal{C}}$

Input : A single vertex v_i

Output: Clustering assignments $z(\mathbf{c})$, vertex assignments **c**, Adjacency matrix **A**

1 Identify the set of neighbours **g** for vertex v_i.

2 Include vertex v_i in the set of neighbours **g**

3 Traverse the network using a breadth first search and update the clustering assignments $z(\mathbf{c})$

4 Compute the probability of sampling a vertex assignment, c_{v_i}, using Definition (5.3.5)

5 Compute $\mathbf{\Omega}_S$ for the updated cluster and log-likelihood $\mathcal{L}$ using Equation (5.11).

6 Sample the vertex assignment using the probability $p(c_{v_i})$

7 Update clustering assignments $z(\mathbf{c})$, vertex assignments **c** and adjacency matrix **A**

To improve the efficiency of the graph traversal that explores connected components, Algorithm 5 focusses on the rewiring of edges between vertices within a cluster in the binDCRP graph. In Algorithm 5, the updates are described for a specific cluster. To begin with, for a single vertex v_i, a set of neighbours **g** is identified for the vertex. The set of neighbours is modified to also include v_i and restricted to include only vertices that belong to the same cluster as the vertex v_i. The Algorithm searches through the connected component using the breadth first search and then samples a vertex. The vertex assignments **c** and the adjacency matrix **A** are updated to store this new structure.

Algorithm 5: Update edges subject to no change in cluster structure - 'Rewiring' within clusters; U_{fc}

Input : A single vertex v_i

Output: Adjacency matrix **A** and vertex assignments **c**

1 Identify the set of neighbours **g** for vertex v_i

2 Include vertex v_i in the set of neighbours **g**

3 Identify cluster C that vertex v_i belongs to.

4 Update vector of neighbours **g** to only include vertices within the relevant cluster.

5 Traverse the network using a breadth first search

6 Sample a vertex assignment

7 Update vertex assignments **c** and adjacency matrix **A**

The Metropolis-Hastings updates for the parameters in the spatio-temporal precision matrix $\mathbf{\Omega} = \mathbf{\Omega}_S \otimes \mathbf{\Omega}_T$ is introduced in Algorithm 6. The algorithm is introduced for an example parameter θ that accounts for the range of θ and is defined within the temporal precision matrix $\mathbf{\Omega}_T$. The algorithm can be also be applied for the parameter ϕ defined in $\mathbf{\Omega}_T$. This algorithm can also be generalised to infer other parameters within the spatial precision matrix $\mathbf{\Omega}_S$ such as λ and ρ. An initial value is first specified and a candidate parameter is generated from a Gaussian distribution. The likelihood $\mathcal{L}_{new}$ is computed with the new parameter and the acceptance ratio a is then determined. The chain is started from an arbitrary initial value and the set of accepted values represent a sample from the distribution of the parameter θ.

Algorithm 6: Metropolis-Hastings updates for a parameter θ of the temporal precision matrix $\mathbf{\Omega}_T$

Assign an initial value of θ and set the current likelihood as $\mathcal{L}_{old}$.
Generate θ^* using θ and from the distribution $\mathcal{N}(0, 1)$.
if $\theta^* > 0$ **then**
    Compute $\mathbf{\Omega_T}$ and $\mathbf{\Lambda}_T$ using θ^*
    Compute the likelihood $\mathcal{L}_{new}$ with the updated $\mathbf{\Omega_T}$
    Determine the acceptance ratio $a = \frac{\mathcal{L}_{new}}{\mathcal{L}_{old}} \frac{\theta^*}{\theta}$
    if $a > \mathcal{U}(1)$ **then**
        Update $\mathbf{\Omega}_T$
        Update the likelihood
        Accept θ^* and assign θ equal to θ^*
    else
        Reject the candidate and θ remains the same
    end
Return the values of the parameter θ

The Metropolis within Gibbs sampler is composed of Gibbs sampling steps that are utilised to explore partitioning structures within the network and Metropolis-Hastings updates that infer the ρ, λ and ϕ parameters in the spatio-temporal precision matrix. The updates described in Algorithm 4, 5 and 6 are all utilised within the sampler described in Algorithm 7. The sampler is run over a defined number of iterations and is initialised with a random clustering of the vertices in the graph network. A sample of vertices from the binDCRP graph are drawn at random and the re-wiring of the edges within a cluster associated with each vertex is explored. This process of re-wiring is performed using Algorithm 5. The relevant changes caused by the re-wiring results in updates in the adjacency matrix and vertex assignments. A single vertex v_i is also sampled and changes in cluster assignments are explored for the relevant vertex. The probability framework described over the binDCRP graph is utilised to sample a vertex assignment. Accordingly, the resulting change in the adjacency matrix, vertex assignments and cluster assignments are stored. These updates and set of iterations to enable the rewiring within a cluster are repeated for n_1 iterations.

Splitting the process of exploring all possible cluster structures allows for a more efficient implementation.

Algorithm 7: Inference for the binDCRP based model

Input : Adjacency matrix $\mathbf{A}$ and initial values of hyperparameters ϕ, λ and ρ

Output: Clustering assignments $z(\mathbf{c})$, vertex assignments $\mathbf{c}$ and hyperparameters

```
Initialise random clustering of vertices
Set number of iterations
Set number of steps as n_1
for iter in iterations do
    for steps in 1:n_1 do
        Set n_2 as a subset of vertices; vertices sampled at random from the
         binDCRP graph.
        for v_i in n_2 do
            Implement Algorithm 5 for v_i.
            Update adjacency matrix A and vertex assignments c
        end
        Sample a single vertex v_i from the binDCRP graph.
        Implement Algorithm 4
    end
    Implement Algorithm 6 to infer φ, λ and ρ
end
```

Within the same iteration of the sampler, the cluster structure is utilised to compute the spatial precision matrix $\mathbf{\Omega}_S$ and define the likelihood as in Equation 5.11. The sampler is run over many such iterations to explore the cluster structure in a comprehensive manner.

5.6 Discussion

This chapter introduces a formal Bayesian clustering method that seeks to determine spatially contiguous clusters over a spatio-temporal dataset. This holistic approach to clustering utilises a non-parametric framework to accommodate the geographical constraints

imposed by the network and determines the number of clusters in a data-driven manner. The development of this spatial clustering model for spatio-temporal datasets is motivated by observations recorded over time for vertices arranged as a graph. In the context of this model, a spatial structure includes vertices as junctions in an urban road network or as regions in a map forming areal unit data. The determined spatially contiguous clusters represent distinct temporal patterns and the model distinguishes between differences in the mean. This is unlike the functional distributional clustering model, introduced in Chapter 3, that is motivated by datasets with multi-modal distributions and with varying mean and variance.

In Chapter 4, we introduced the binary distance dependent Chinese restaurant process as a special case of the distance dependent Chinese restaurant process. The binDCRP is utilised to accommodate the geographical constraints imposed by the nature of the graph network. This formal Bayesian model places the binDCRP as a prior; the binDCRP graph is constructed such that each vertex has an edge to only one other vertex or a loop. In this chapter, we modified the binDCRP to restrict the number of clusters by defining a parameter α over the introduction of a loop and a redundant link. The number of clusters is equal to the number of cycles plus the number of loops. It is reasonable to expect that a vertex is spatially correlated to adjacent vertices and the binDCRP does not fully incorporate the within cluster spatial dependencies. A type of conditional auto-regressive (CAR) model is introduced to account for the spatial correlation within a cluster. The CAR model seeks to model the spatial correlation between adjacent vertices using neighbourhood relationships within a cluster. A first-order auto regressive (AR-1) model is also introduced to account for the temporal dependencies. The spatial and temporal precision matrix is utilised to define the spatio-temporal precision matrix $\mathbf{\Omega}$. The model assumes that observations follow a Gaussian distribution and that there are no missing observations associated with each vertex in the dataset. This allows for the defined likelihood to be rewritten using Kronecker product tricks over the spatio-temporal precision matrix and enables the model to be implemented in a computationally efficient manner. The Metropolis within Gibbs sampler is described over the binDCRP graph framework and explores all possible cluster structures.

Chapter 6

Application

Traffic congestion in an urban road network

Traffic occupancy is a process that varies over time and needs to be studied over both space and time. The formal Bayesian model (as described in Chapter 5) seeks to identify connected components that correspond to distinct temporal patterns within the network. In traffic modelling, this is applied to identify distinct temporal patterns of occupancies over time. Several methods exist within the transportation modelling literature that seek to determine spatially contiguous clusters to minimise heterogeneity (Saeedmanesh and Geroliminis, 2016, 2017). However, constraints posed by associated observations recorded over time pose multiple and often unique challenges. The formal Bayesian approach introduced in Chapter 5 provides a framework that is constructed to fully incorporate spatial, temporal and network dependencies and seeks to identify the number of clusters in a data-driven manner. In this chapter, we first examine several features of the binDCRP based model using an application to a simulated spatio-temporal dataset over an urban road network. This study includes the utilisation of varying α parameter values to demonstrate its ability to restrict the number of clusters and also examines the improvement in the computational efficiency of the sampler by the adaptation of Kronecker product rules. This method accommodates geographical constraints imposed by the structure of the urban road network and accounts for the spatial correlation between adjacent junctions within a cluster. We then apply the method to the dataset generated by the AIMSUN simulator (introduced

in Chapter 3) to illustrate well-defined cluster structures and evaluate the mixing of the sampler. This chapter also includes an application to areal unit data for property prices that are recorded from 1995 to 2016 in Avon county, England.

6.1 Simulated data

In this section, data is simulated over the San Francisco network to reflect three distinct clusters. The occupancy data is simulated over a 2.5 square miles network (network is as specified in Chapter 3) in downtown San Francisco. This urban road network is composed of 158 junctions and 316 links. The urban road network is again composed of junctions that each has a maximum of four adjacent junctions; this limited number of road segments between junctions in the network translates to a sparse spatial precision matrix. The spatial correlation is modelled by a type of conditional auto-regressive (CAR) model introduced by Leroux et al. (2000) and the temporal precision structure follows a first order auto-regressive (AR-1) model. Occupancy observations are generated for each junction in the network over six hours with a sampling rate of 60 seconds. Correlated data is generated for each junction over time and is defined by a spatio-temporal precision matrix that is utilised to model the spatial and temporal dependencies. The spatio-temporal precision matrix is utilised to define the three distinct clusters $\mathcal{C} = \{C_1, C_2, C_3\}$, such that within each cluster, a state space model generates zero and one values corresponding to defined occupancy levels.

Figure 6.1 displays occupancy observations that are simulated over a period of six hours for each cluster (represented by blue, green and red). Each cluster represents a temporal pattern with different mean values but with common variances. Occupancy observations represented by a temporal pattern in blue (40 - 80 %) have a higher mean value than the temporal pattern in red (30 - 70 %). These observations both have higher mean values than the observations presented by the pattern in green (15 - 55 %). These generated values differ from the data simulated in Chapter 3. The spatio-temporal dataset simulated in Chapter 3 is composed of clusters that represent differences in both the mean and the variance and the functional distributional clustering algorithm seeks to accommodate multi-modal distri-

butions. The mean-based Bayesian clustering model assumes that the observations follow a Gaussian distribution. In this section, we first apply the developed flexible binDCRP-based Bayesian model to examine its performance over the simulated spatio-temporal dataset with a known true cluster structure.

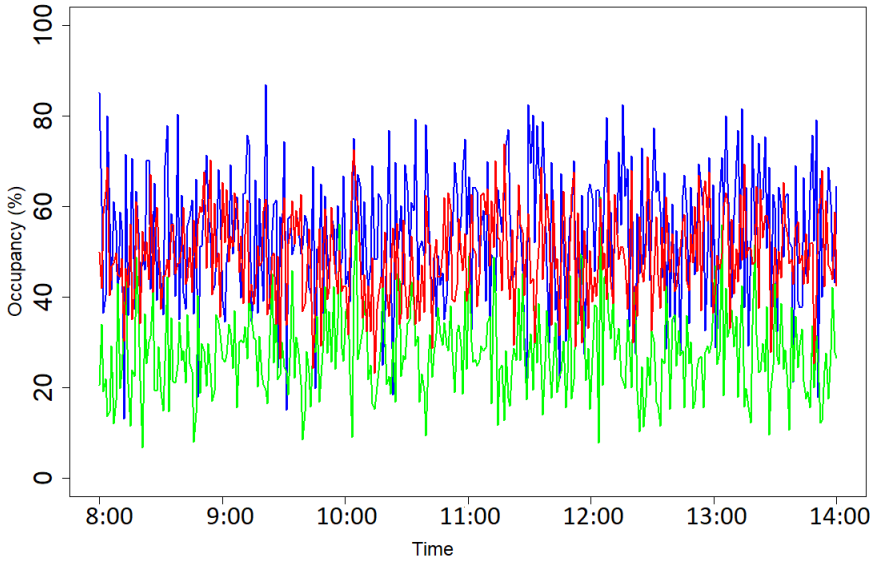
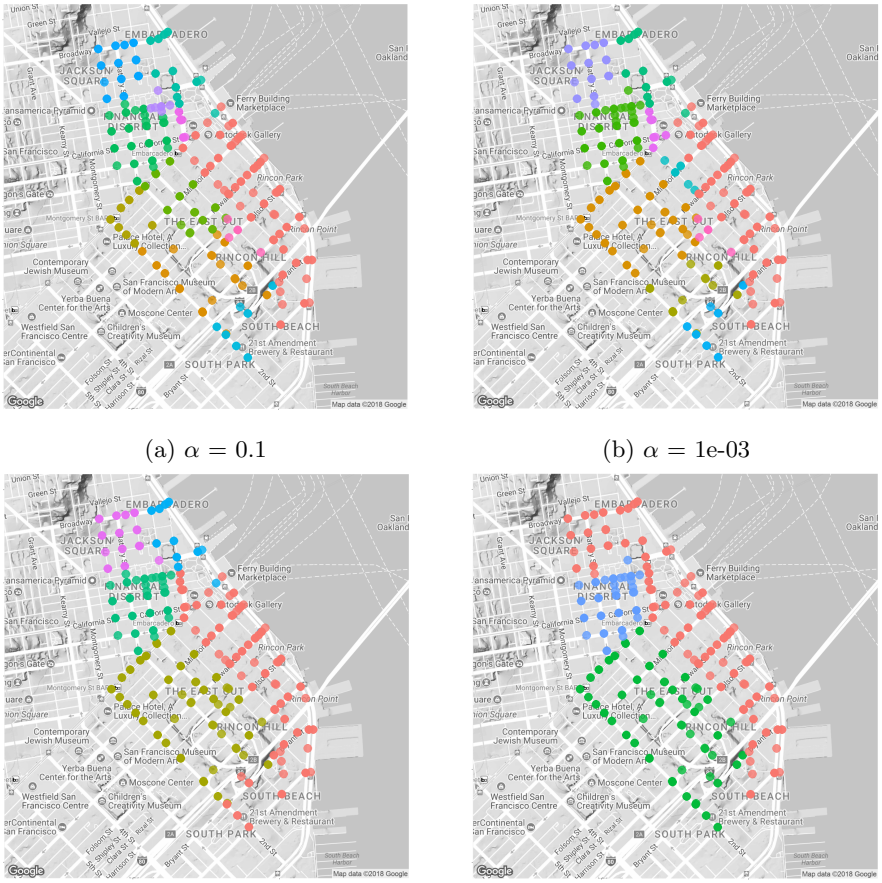


Figure 6.1: Simulated data for three clusters in the network

6.1.1 Number of clusters

In the regular ddCRP (described in Chapter 4), the formation of a new cluster is not limited to the introduction of a loop. The binDCRP, as a special case of the ddCRP applied to spatial datasets, retains this framework and new clusters are formed by both cycles as well as loops. However, this results in poorly defined cluster structures and a large number of singletons. We modified the binDCRP (introduced in Chapter 5) to enable the model to exert control over the number of clusters using the parameter α. This parameter α restricts new clusters that are formed by loops and cycles. In the binDCRP graph, the addition of a redundant edge to a path results in a cycle. We utilise the above simulated spatio-temporal dataset over an urban traffic network to demonstrate the model's ability to control the number of clusters in the model.

Figure 6.2 displays clustered networks at different values of the parameter α. The parameter determines two clusters at α = 1e-07, three clusters at α = 1e-04 and steadily more clusters at higher levels of α. At $\alpha = 1e-04$, the true cluster structure composed of three distinct clusters is determined.



(a) α = 0.1

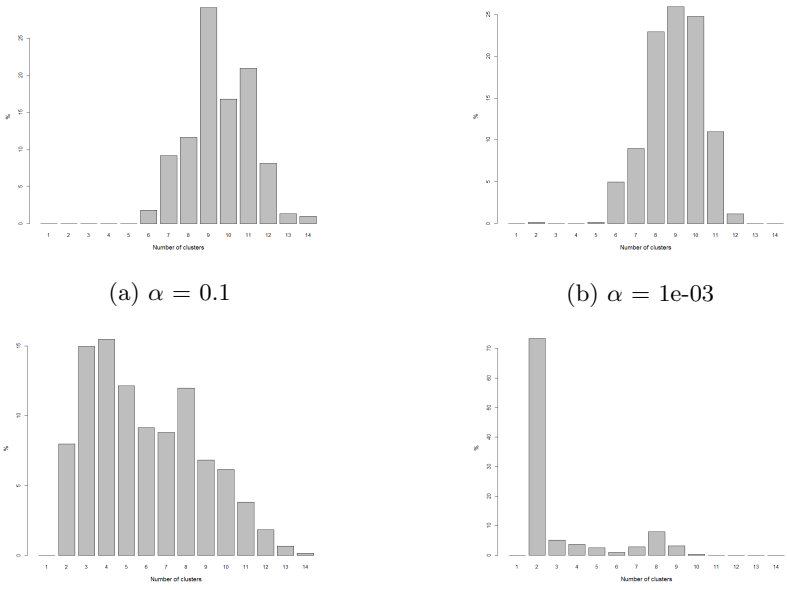
(b) α = 1e-03

(c) α = 1e-04

(d) α = 1e-07

Figure 6.2: Clustered networks with varying number of clusters at different values of α

In Figure 6.3, the number of clusters reduce with a decrease in the value of the parameter α. Each sub-plot displays a distribution of the number of clusters that are determined from the sampler (described in Algorithm 7, Chapter 5).



(a) $\alpha = 0.1$ (b) $\alpha =$ 1e-03

(c) $\alpha =$ 1e-04 (d) $\alpha =$ 1e-07

Figure 6.3: Distribution of number of clusters at different values of α

6.1.2 Spatio-temporal precision matrix

The spatio-temporal dataset simulated over the San Francisco network is composed of 158 junctions and 360 observations are recorded for each junction over a period of six hours. The relevant spatio-temporal precision matrix is denoted by $\mathbf{\Omega}$ and is written as $\mathbf{\Omega} = \mathbf{\Omega}_S \otimes \mathbf{\Omega}_T$. The dimensions of the spatial precision matrix, the temporal precision matrix and the spatio-temporal precision matrix are listed as follows:

$$\dim(\mathbf{\Omega}_S) = 158 \times 158$$

$$\dim(\mathbf{\Omega}_T) = 360 \times 360$$

$$\dim(\mathbf{\Omega}) = \dim(\mathbf{\Omega}_S \otimes \mathbf{\Omega}_T) = 56880 \times 56880$$

As computed above, the Kronecker product of the spatial and temporal precision matrix results in a very large matrix. It would be ideal to avoid the computation of this large matrix and this is possible by the utilisation of relevant Kronecker product identities. By utilising the Kronecker product identities that are introduced in Chapter 5, the computational time

is reduced significantly. The implementation of Kronecker product identities assumes the presence of a unique observation for every space and time combination and this hold for the simulated spatio-temporal dataset. For example, using Identity I, the computation time to evaluate $\mathbf{\Omega}_S^T \otimes \mathbf{\Omega}_T \text{vec}(\mathbf{X})$ is significantly reduced for a single iteration within the sampler. In the simulated data over the San Francisco network, $\text{vec}(\mathbf{\Omega}_T \mathbf{X} \mathbf{\Omega}_S)$ is computed in a tenth of a second, as compared to $(\mathbf{\Omega}_S^T \otimes \mathbf{\Omega}_T)\text{vec}(\mathbf{X})$ which takes six seconds. The difference in computational time for a single iteration leads to a significant reduction time for the overall sampler. The computation within the likelihood uses the Identity I and Identity II and simplifies the term $\text{vec}(\mathbf{X})^T(\sigma^2\mathbf{I} + \tau^2\mathbf{\Omega}_S \otimes \mathbf{\Omega}_T)^{-1}\text{vec}(\mathbf{X})$ using eigenvalues and eigenvectors. This term, when rewritten as $\text{vec}(\mathbf{X})^T(\mathbf{\Gamma}_S \otimes \mathbf{\Gamma}_T)(\sigma^2\mathbf{I} + \tau^2\mathbf{\Lambda}_S \otimes \mathbf{\Lambda}_T)^{-1}(\mathbf{\Gamma}_S^T \otimes \mathbf{\Gamma}_T^T)\text{vec}(\mathbf{X})$, can be computed very efficiently.

6.2 AIMSUN simulator

In this section, the Bayesian method is applied to a spatio-temporal dataset that is generated over the 2.5 square miles network area in downtown San Francisco, CA. The same dataset is introduced in Chapter 3, but this chapter utilises the data that is recorded over the entire six hours. In general, higher resolution spatio-temporal data for urban road networks is not necessarily available in open data sources and the AIMSUN simulator serves to replicate multiple scenarios. More specifically, the AIMSUN microscopic traffic simulator is utilised to mimic origin destination traffic demand scenarios over the network (Barceló and Casas, 2005, Barceló et al., 2010, Casas et al., 2010). Transportation researchers have utilised data generated from an AIMSUN simulator for simulation experiments. We generate data that is able to replicate realistic urban traffic network scenarios in a manner that also broadly reflects three distinct patterns within the network. The AIMSUN simulated data suggests an approved way to model the network in simulation experiments and has been widely utilised in transportation research (e.g., Geroliminis et al. (2014), Saeedmanesh and Geroliminis (2016)). The nature of spread of traffic congestion, presence of spatial correlation (both across the network and within clusters) and the need to necessarily model both spatial and temporal dimensions pose multiple challenges.

Occupancy observations, for each junction in the downtown San Francisco network, are simulated over six hours from 8 am in the morning to 2 pm in the afternoon. In Figure 6.4, the downtown San Francisco network and the relevant region of interest is highlighted. The region within the brown border translates to the region of interest and is composed of 158 junctions. The darker brown line along Market Street highlights a divide that indicates differences in the range of occupancy values for the region. The simulated data reflects the expectation that the network has lower occupancy earlier in the day and higher occupancy levels during the middle of the day. The higher levels of occupancy represent an increase in vehicular traffic that is caused by lunch and associated travel within the urban network.

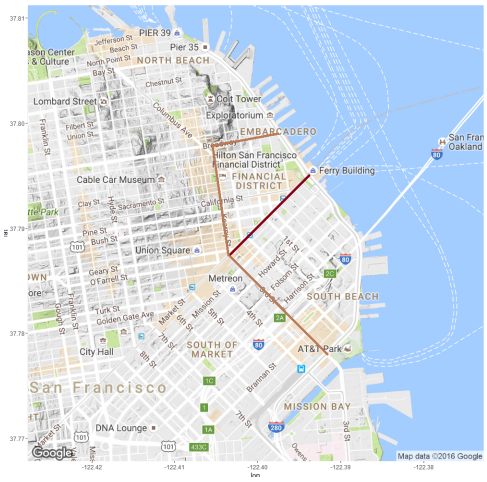
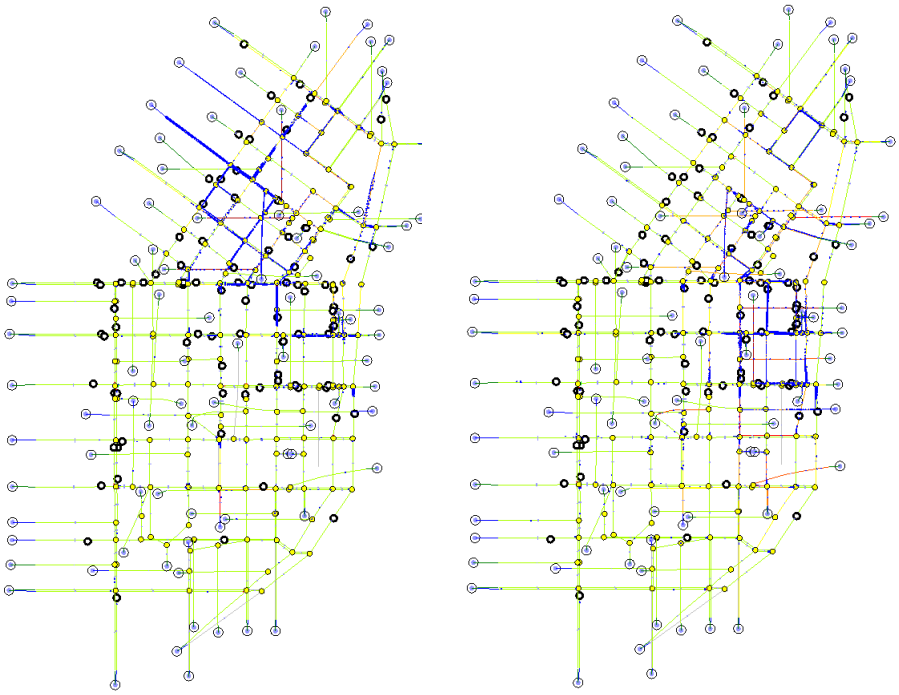


Figure 6.4: Downtown San Francisco

The scenarios are generated such that there are multiple sources that introduce traffic into the urban road network. Figure 6.5 displays urban road networks with traffic congestion levels at 9 am and 1 pm. Figure 6.5a displays the traffic congestion at 9 am, which is concentrated in the top part of the network, above Market Street. Congestion is concentrated to the right, towards the later part of the time period (six hours), as displayed in Figure 6.5b. An initial source of occupancy is at the left of the network, this spreads through the network and then concentrates towards the right. This reflects the differences in the traffic demand scenarios corresponding to three distinct clusters in the network. In this section, the Bayesian model seeks to identify a cluster structure that reflects these described

differences in congestion through the day.



(a) 9 am (b) 1 pm

Figure 6.5: Traffic congestion in the network as generated from AIMSUN simulator

In Figure 6.6, the San Francisco network is displayed and the sources that generate vehicular movement within the network are circled in red, green and blue. We limit the dataset to only include the circled junctions positioned within the network. The dataset does not include circled junctions that are positioned outside the network; these sources are circled in purple, brown and blue and are placed outside the urban road network. The vehicular traffic initially builds from the purple and red sources and is concentrated around the neighbouring junctions. This corresponds to the movement of traffic from the region to the left of the Financial district towards the Financial district. A major proportion of the traffic in the network is introduced from the sources circled in green and brown. The vehicular occupancies then concentrate towards the right middle of the network and this is also displayed in Figure 6.5b. This translates to the movement of traffic from the South of Market Street and towards the East Cut and the Embarcadero. These transitions in the

movement of traffic are caused by the expected change around lunchtime. The spread in traffic occupancy across the network mimics a continuous process and its dynamic nature poses multiple challenges to the development of clustering algorithms.

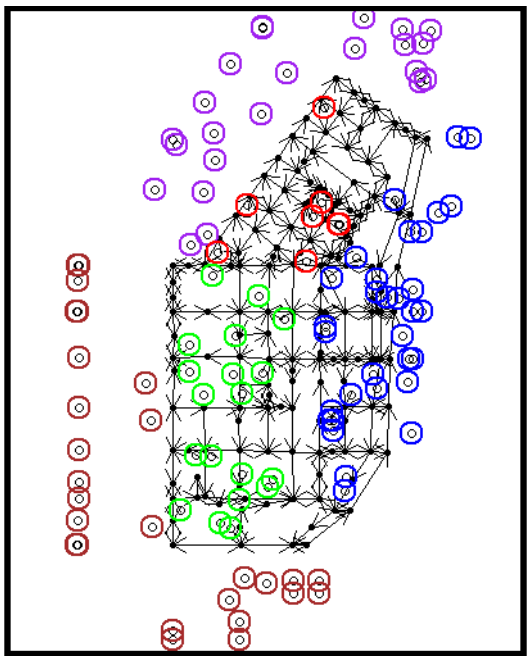
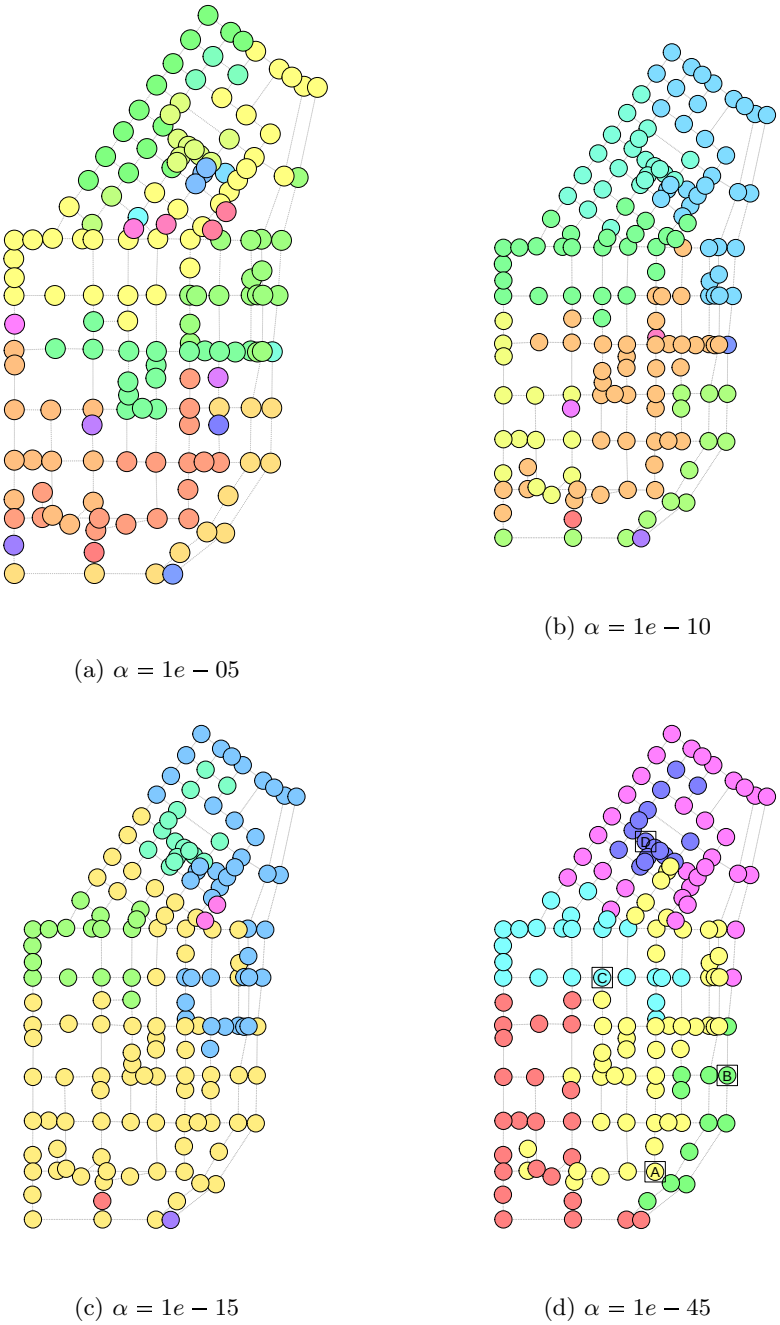


Figure 6.6: San Francisco network composed of 158 junctions. Individual junctions that serve as sources of vehicular traffic in the network are circled and indicate differences in the occupancy observations. These differences translate to unique temporal patterns and distinct clusters.

The Bayesian model is applied to this described data and the different clustering results are presented in Figure 6.7. The model is initialised with $\phi = 0.5, \lambda = 50$ and $\rho = 0.75$ and 1500 iterations of this sampler are run over the dataset. The defined spatial precision matrix is over 158 junctions and the temporal precision matrix is defined over 120 observations such that $\dim(\mathbf{\Omega}_S) = 158 \times 158$ and $\dim(\mathbf{\Omega}_T) = 120 \times 120$. The chosen clustering results are determined as corresponding to the highest probability among other clusterings. In Figure 6.7d, the clustering result is chosen corresponding to the highest posterior mode at 7.244. The cluster structure is composed of six spatially contiguous clusters and there is an expected division along Market street between the clusters in purple and pink as compared to the clusters in green, red, blue and yellow. The numerous singletons that are formed at higher values of α are less likely to be formed at lower values.



(a) $\alpha = 1e-05$

(b) $\alpha = 1e-10$

(c) $\alpha = 1e-15$

(d) $\alpha = 1e-45$

Figure 6.7: Clustering results over the network at different levels of α

In Figure 6.7d, we highlight several junctions in the clustering ouptut and compare the cluster structure across multiple chains. In each chain, the posterior mode is utilised to select the cluster structure. For junction A, it belongs to the same cluster in nine out of

ten such chains. Similarly, the junction labelled B belongs to the same cluster in seven out of ten, the junction labelled C belongs to the same cluster in six out of ten and the junction labelled D also belongs to the same cluster in six out of ten. The clustering output displayed in Figure 6.7d is composed of six clusters and the temporal pattern for each of the six clusters is presented in Figure 6.8.

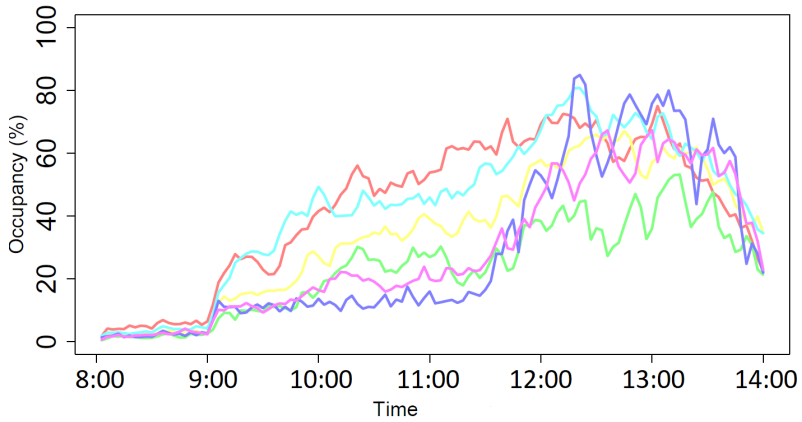
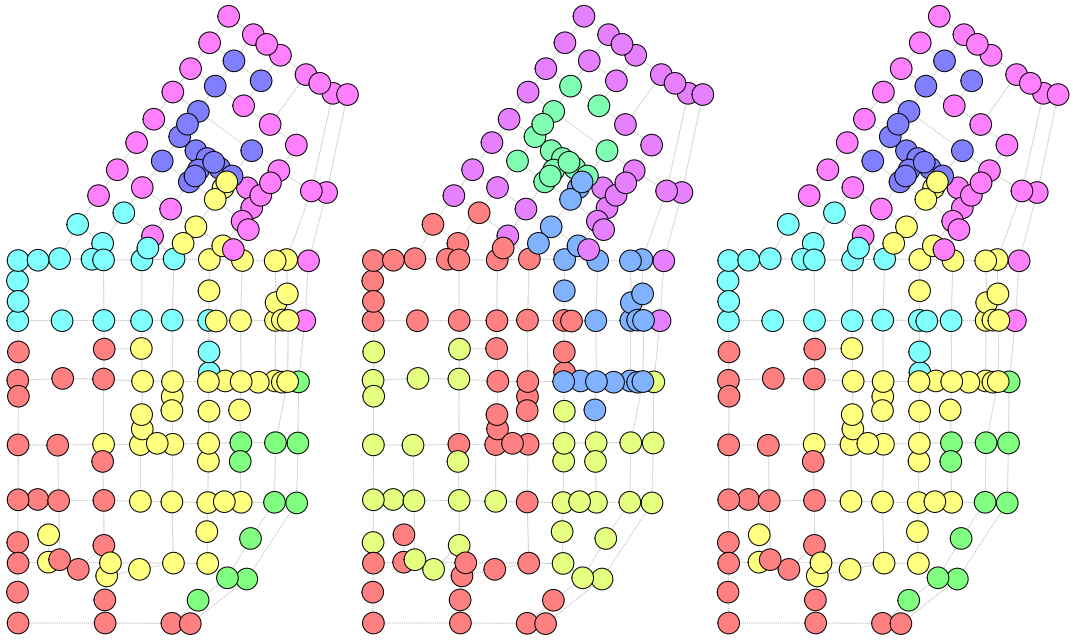


Figure 6.8: Temporal pattern corresponding to the determined clusters in Figure 6.7d

The temporal pattern is composed of higher occupancies for the red and blue clusters (particularly between the second and upto the fourth hour). With the division of clusters along Market Street, this also corresponds to a temporal pattern composed of lower occupancies (earlier in the day) for the pink and purple cluster. These patterns of traffic reflect the nature of the traffic scenarios simulated by the AIMSUN simulator. The generated sources of traffic are concentrated on the left portion of the network and includes the red, blue, purple and pink clusters. Accordingly, the increase in traffic is associated with lunchtime and the patterns presented in Figure 6.8 reflect this increase in traffic. In addition, the higher values of occupancy (earlier in the day), for the red and the blue clusters, suggest that the traffic spreads from the South of Market street area towards the Financial district area, the Embarcadero and the East Cut. The traffic peaks at noon and then steadily starts to diminish across the network over the remaining two hours of the evaluated time period.

This is also reflected in the patterns generated in Figure 6.9, where multiple clustering results within the same chain are displayed. The results do not correspond to the highest

posterior mode (as displayed in Figure 6.7d) but provide other potential cluster structures. The posterior mode for the structure in Figure 6.9a is 3.164, in Figure 6.9b is 2.748 and in Figure 6.9c is 1.332.



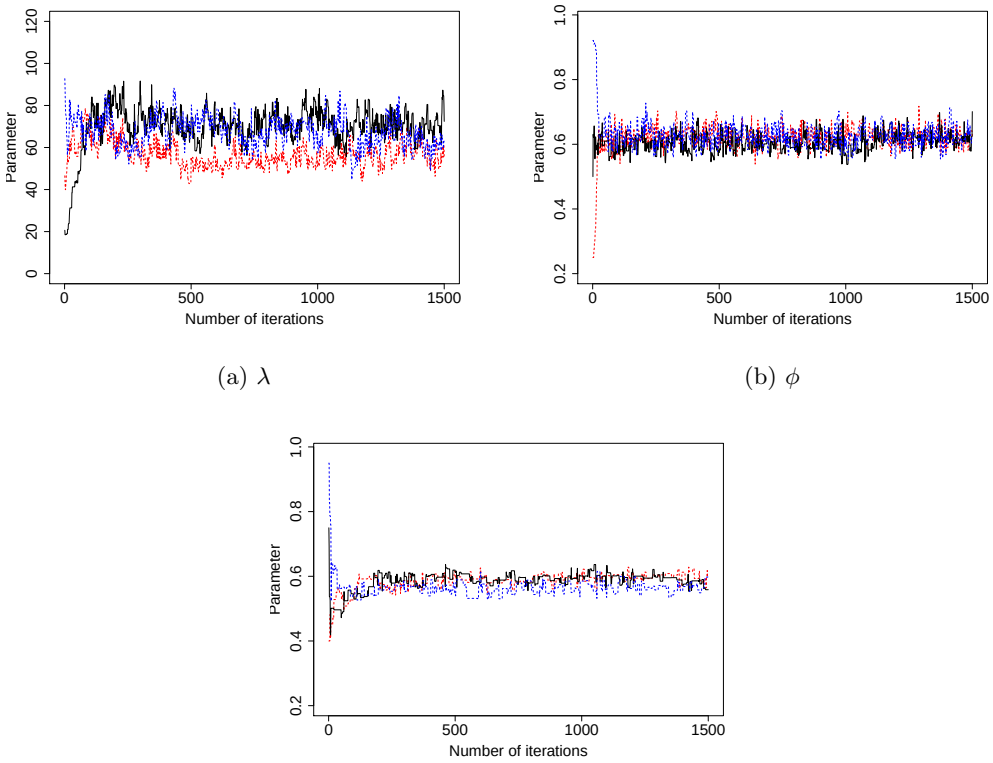
(a) (b) (c)

Figure 6.9: Cluster structures at lower posterior modes, when $\alpha = 1e - 45$

6.2.1 Diagnostics

The performance of the sampler is evaluated for the model when the α parameter is set at 1e-45. We examine multiple aspects of the MCMC including mixing, burn-in and the run length. The sampler should explore the entire parameter space efficiently such that it does not reject or accept too many proposals. Trace plot is one such important tool that seeks to assess the mixing of the chain. Figure 6.10 displays the trace plots for the parameters inferred by Metropolis-Hastings updates. Ideally, a trace plot should not be composed of a steadily increasing or decreasing pattern. The burn-in is assessed by a glance at the trace plots for the three parameters, as displayed in Figure 6.10. Figure 6.10a and Figure 6.10b display trace plots for the λ and ϕ parameter. Removing the first 300 observations, the trace

plot hovers between 0.55 to 0.7 for ϕ and 0.5 to 0.65 for ρ. We also started the sampler at higher and lower initial values of λ, ϕ and ρ to ensure that the sampler continues to converge to the same range of estimated values. This allows the reliability of the output to be assessed since a sampler can be stuck in a local maximum. Different initial values for the parameter ϕ and ρ results in chains that converge to the same range of values. In general, it is rather difficult to tell how long the chain should be run but the trace plot typically serves as one such indicator of the efficiency of the MCMC sampler. The trace plot for the parameter ϕ indicates very good mixing of the chain, which indicates that the relevant parameter space is explored efficiently. The trace plot for the λ parameter also indicates reasonable mixing. However, the trace plot for the ρ parameter shows poor mixing; a well-mixing chain would move freely without getting stuck in regions of the parameter space. This could suggest the need for the MCMC sampler to be run over more number of iterations as well as the need for other modifications to be investigated. In Figure 6.10, the trace plots above are displayed for the MCMC sampler when the model uses an α value of 1e-45. Similar plots for other values of α (α = 1e-05, 1e-10, 1e-15 and 1e-80) are provided in the Appendix.



(a) λ (b) ϕ

(c) ρ

Figure 6.10: Trace plots for the parameters λ, ϕ, ρ

6.3 Property prices

6.3.1 Data

This section utilises data for associated geographies, with a focus on middle layer super output areas (MSOA), over the Avon county in England. The Avon county is composed of four local authority areas, 'North Somerset', 'Bath and North East Somerset', 'Bristol, City of' and 'South Gloucestershire'. Residential property transactions from 1995 to 2016 are available from the Office of National Statistics (ONS) for MSOA units across the Avon county. Additional details of the housing price statistics for small areas (HPSSA) in England can be found at https://www.ons.gov.uk/peoplepopulationandcommunity/housing/bulletins/housepricestatisticsforsmallareas. Similarities and differences

in the changes of property values over time, for areal units of interest in the Avon county, can be identified using the flexible Bayesian model introduced in Chapter 5. Observations over time (from 1995 to 2016) for MSOA units in Avon county lead to a dataset composed of 140 units that are each adjacent to a limited number of other units. This defines a sparse adjacency matrix and corresponding precision matrix. In this spatio-temporal dataset, a unique observation is present for every combination of space and time.

The Bayesian approach introduced for spatio-temporal data in Chapter 5 is applied to a map with areal unit associated data. The binDCRP graph framework introduced in Section 5.2 can also be adapted to this map based spatial structure. The spatio-temporal matrix is defined over 140 regions in the graph and 22 recorded observations such that $\dim(\mathbf{\Omega}_S) = 140 \times 140$ and $\dim(\mathbf{\Omega}_T) = 22 \times 22$.

6.3.2 Results

Figure 6.11 displays a cluster structure over four local authority areas 'Bristol, City of', 'Bath and North East Somerset', 'North Somerset', and 'South Gloucestershire'. The clusters are determined using the relevant MSOA units for the local authority areas. The Avon County is collectively composed of these four local authority areas and the application of the Bayesian method results in eight clusters. Figure 6.12 displays a plot that describes the temporal pattern associated with each cluster. The observations for the MSOA units within the cluster are aggregated to determine a single trajectory over time for each cluster. Cluster 2 represents a distinct temporal pattern compared to clusters 1, 3, and 4.

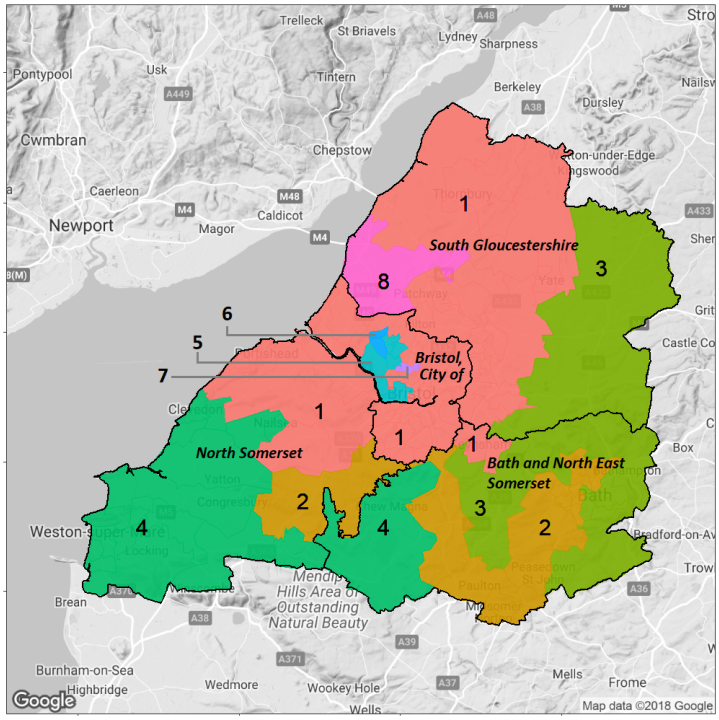


Figure 6.11: Cluster structure of the Avon county composed of four local authority areas. This is determined using housing prices data recorded from 1995 to 2016 for MSOAs.

In Figure 6.13, the temporal patterns for the eight determined clusters are displayed as separate plots; this enables individual differences to be studied in an easier manner. The observations over time for MSOA units in cluster 2 represent higher property prices than for cluster 3. Aggregating over these observations result in two distinct temporal patterns as displayed in Figure 6.13b. Cluster 4 is composed of observations over a greater range of property prices such that there are higher prices as well as lower prices. These are aggregated to form a temporal pattern above cluster 1 as displayed in 6.13a. In general, the property prices values for observations associated with cluster 3 are lower than the observations for MSOA units in cluster 1, 2 and 4. The range of property prices is also narrower for cluster 3 compared to the range of property prices for observations over MSOA units in cluster 1, 2 and 4. In Figure 6.13c, the clusters correspond to trajectories that each have a distinct temporal pattern. The clusters are also determined to ensure that they satisfy constraints that result in the formation of spatially contiguous clusters. For example, cluster 7 and cluster 8 cannot belong to the same cluster unless cluster 1 is broken up; this would lead to

the formation of a different partitioning.

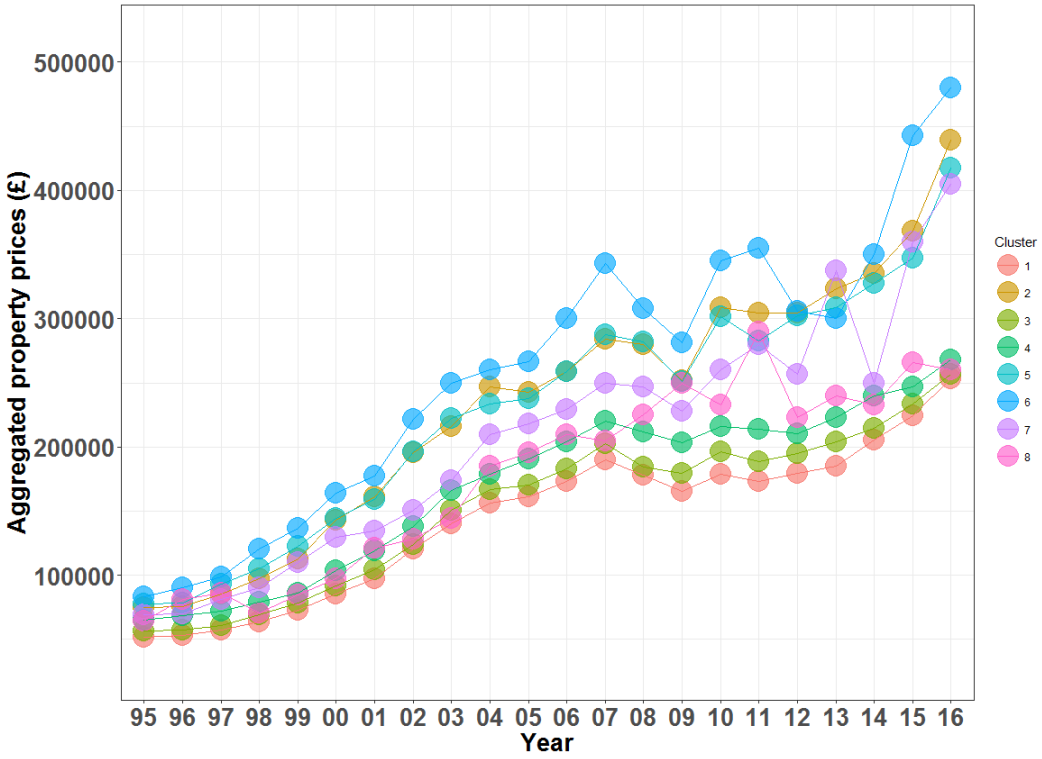
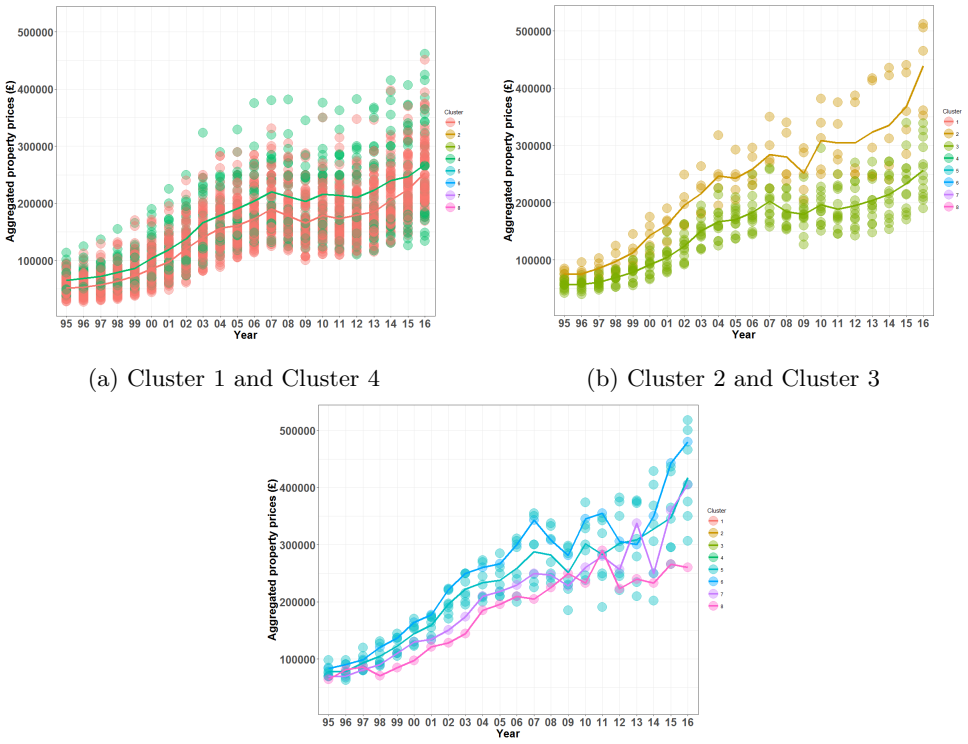


Figure 6.12: Temporal patterns for clusters using median house price data (from 1995 to 2016) recorded for Middle Layer Super Output areas (MSOA).



(a) Cluster 1 and Cluster 4

(b) Cluster 2 and Cluster 3

(c) Cluster 5, 6, 7 and 8

Figure 6.13: Observations over time for the clusters displayed in Figure 6.11

6.4 Discussion

This chapter primarily focusses on illustrating the applications of the binDCRP based Bayesian clustering model (introduced in Chapter 5). This model is applied to multiple spatio-temporal datasets and the examples utilised in this chapter are motivated by observations recorded for junctions in an urban road traffic network as well as observations associated with a map-based structure. We simulated a spatio-temporal dataset over an urban road network such that occupancy observations over time are associated with junctions over the network. In this spatio-temporal dataset, occupancy observations are recorded over a period of time for each junction in the urban road network. The distinct clusters over three non-overlapping regions in the network represent occupancy data with different mean values but with common variance. This simulated spatio-temporal dataset differs from the dataset simulated in Chapter 3; the simulated dataset in Chapter 3 represents three distinct

clusters with different distributions (both mean and variance).

In Chapter 5, we modified the binDCRP to control both the number of singletons and the number of redundant links that lead to cycles. This restricts the number of clusters that are formed over the spatial structure using the parameter α. In this Chapter, we utilise the spatio-temporal simulated data to evaluate the ability of the binDCRP based model to restrict the number of clusters within the urban road network. Different values of the α parameter are chosen and the associated cluster structure demonstrates the number of clusters at different values of α. The binDCRP is able to exert reasonable control over the number of clusters and the model generates fewer number of clusters at lower levels of α. Within the simulated spatio-temporal dataset, there is a unique observation for every space and time combination and this allows for the utilisation of Kronecker product identities introduced in Chapter 5. In this chapter, we also demonstrate the ability to improve the computational efficency of the model by utilising Kronecker product identities.

This chapter also illustrates the performance of the binDCRP based model by an application to the real-world AIMSUN traffic simulator dataset. This dataset was first introduced in Chapter 3, but we describe this spatio-temporal dataset and the associated scenarios in greater detail. The dataset generated by the AIMSUN simulator is composed of three different demand scenarios and the spread of congestion is simulated as a continuous process over the network. Unlike the spatio-temporal dataset that is simulated to generate three distinct clusters, the three different demand scenarios lead to clusters that have considerable overlap in associated temporal patterns. This dataset does benefit from the Bayesian model's ability to accommodate geographical constraints over the network and incorporate spatial correlation within a cluster. It is reasonable to assume that a junction within a cluster is spatially correlated to adjacent junctions and the model utilises a conditional auto-regressive (CAR) model to account for this level of spatial dependency. The binDCRP-based model is run at different values of the α parameter and we select a cluster structure that corresponds to six distinct spatially contiguous clusters. The Bayesian framework of the binDCRP-based model also enables us to generate several clusterings associated with the posterior mode.

We examine the mixing of the sampler and present relevant diagnostic plots. The ability of the binDCRP-based model to be applied to an areal unit dataset is also demonstrated by an application to property prices recorded over twenty two years for the Avon county in England, United Kingdom. The Avon county is composed of four local authority areas and observations are recorded for middle layer super output areas (MSOAs). Eight clusters are identified over the Avon county and the mean-based differences between these clusters are described in this chapter.

Chapter 7

Conclusions

Spatial clustering algorithms seek to adequately accommodate the geographical constraints posed by the network and generate meaningful clusters. This thesis focusses on the development of clustering algorithms that identify spatially contiguous clusters by accounting for the spatial, temporal and network dependencies within the spatio-temporal data. The spatial clustering methods introduced in this thesis are nonparametric and are also motivated by challenges posed by different temporal pattern scenarios. The developed methods are directed towards identifying clusters that represent mean-based differences and distribution-based differences (including both mean and variance). The development of new spatial clustering methods for spatio-temporal datasets is motivated by the need to identify meaningful clusters in a computationally efficient manner. In this thesis, the clustering methods are described for a grid-style graph network; a graph network is composed of vertices and edges between the vertices and each vertex is assumed to have a limited number of adjacent vertices. Spatial structures in this thesis are represented as a road network with junctions and road segments between junctions as well as a map composed of areal units. Both these structures can be translated to a graph composed of vertices and edges between vertices. The examples in this thesis are primarily from traffic modelling, where the goal is to identify distinct patterns of traffic congestion within an urban road network. Occupancy observations are recorded over time for each junction and a congested network corresponds to higher levels of occupancy in the network. In addition, the formal Bayesian approach to clustering is also described in the context of map based spatial structures and its application

is illustrated using property prices recorded over a period of time for associated areal units.

The first method introduced in this thesis is the functional distributional clustering algorithm and this approach to clustering utilises a measure of distance that is both functional and distributional. This ad-hoc approach is implemented within an agglomerative hierarchical clustering framework and the method generates a hierarchy of clusters. In order to choose the optimal number of clusters from this hierarchy of clusters, we introduce a modified clustering balance criterion. The clusters are distinguished by differences in the densities over time and are able to accommodate multi-modal distributions. Unlike mean-based clustering methods, clusters represent differences in both the mean and the variance over the temporal pattern. In addition, observations within the method are not assumed to follow a Gaussian distribution assumption. A visualisation composed of three-dimensional plots for each cluster is also introduced within the framework of this method. Each three-dimensional plot describes the change in densities over time and is utilised to effectively represent the differences between the clusters. The performance of this method is demonstrated by its ability to detect the underlying true cluster structure within a comprehensive simulation study. The simulation study highlights the superior performance of a functional distributional clustering approach compared to functional clustering, distributional clustering and functional data analysis (FDA) based clustering approaches.

The second method introduced in this thesis is a flexible Bayesian approach to clustering that is able to determine the number of clusters in a data-driven manner. This is a mean-based approach that is implemented within a formal statistical framework and the model assumes that occupancy observations follow a Gaussian distribution. We first introduce a special case of the distance dependent Chinese restaurant process that is adapted for spatial data and define this as the binary dependent Chinese restaurant process (binDCRP). The model utilises the binDCRP as a prior to accommodate the geographical constraints imposed by the structure of the network. In this model, the binDCRP is extended to restrict the number of clusters and the number of clusters corresponds to the number of loops and number of cycles. The model also assumes that observations recorded for a vertex in the

graph are spatially correlated to adjacent vertices. In order to fully incorporate the spatial dependencies within a cluster, a conditional auto-regressive model is utilised. Ideally, a mixture of CAR models would adequately incorporate the differences in correlation within the network. However, the binDCRP based model depends on an assumption of conjugacy and this would no longer hold if a mixture of CAR models is introduced over individually identified clusters. A first order auto-regressive (AR-1) model is also utilised to accommodate the temporal dependencies. The binDCRP based approach is implemented with the assumption that there is a unique observation for every space and time combination. This allows Kronecker product identities to be applied to the spatio-temporal precision matrix (defined within the likelihood), which significantly improve the computational efficiency of the sampler. We utilise a Metropolis within Gibbs sampler to infer relevant parameters defined in the model and explore all potential partition structures within the graph network. The ability to search through the network and reach all vertices within a connected component is aided by a breadth first search. As future work, we seek to develop a comprehensive simulation study that compares the Bayesian clustering approach to other existing nonparametric spatial clustering methods.

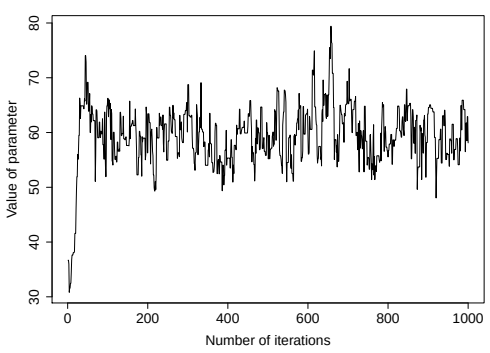
The binDCRP based Bayesian clustering method is applied to simulated data over the urban road network, data generated by a real-world AIMSUN traffic simulator using well defined origin-destination demand scenarios and to areal unit data. The simulated spatio-temporal data represents three distinct spatially contiguous clusters, where each cluster is composed of junctions that are spatially correlated to adjacent junctions and the simulated dataset is utilised to demonstrate the performance of the model. The AIMSUN traffic simulator generates data over the same urban traffic network in downtown San Francisco and seeks to mimic the nature of traffic congestion that evolves over a period of time. We utilised the spatio-temporal dataset to examine the mixing of the sampler and present relevant diagnostic plots. This method is also illustrated by an application to observations associated with an areal unit dataset; property prices are recorded over a period of twenty years for areal units in the Avon County in England, United Kingdom.

In this thesis, both methods were developed to determine spatially contiguous clusters that represent distinct temporal patterns. However, the determined clusters, generated by the functional distributional clustering algorithm and the Bayesian approach to non-parametric spatial clustering, are static in nature. In future work, we seek to extend these developed methods in a computationally efficient manner to be able to generate dynamic clusters that also change in shape over time. This would lead to a significant modification in the framework of the Bayesian method, since Kronecker product identities would no longer be applicable.

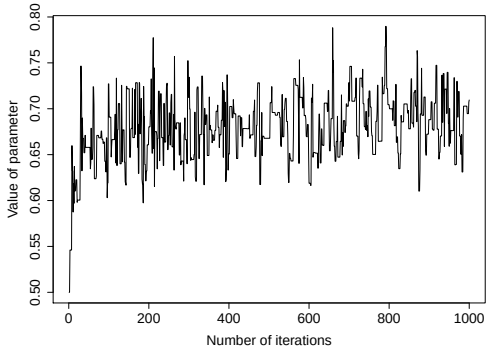
Appendix A

Additional Details

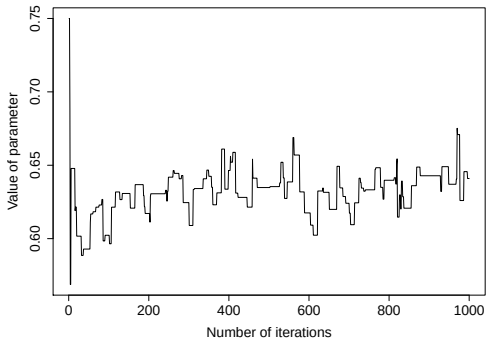
Figure A.1, Figure A.2, Figure A.3 and Figure A.4 display trace plots associated with different values of α parameter. The alpha parameter is utilised within the Bayesian model to restrict the number of clusters and these results are from the AIMSUN simulator example in Section 6.2. The trace plots at each value of α are displayed for ρ, ϕ and λ and the model utilises alpha values of 1e-5, 1e-10, 1e-15 and 1e-80. The results described in the AIMSUN simulator example in Chapter 6 utilises α = 1e-45; the following trace plots are for all other values of α.



(a) λ

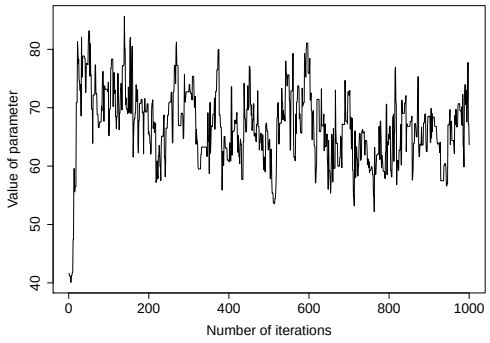


(b) ϕ

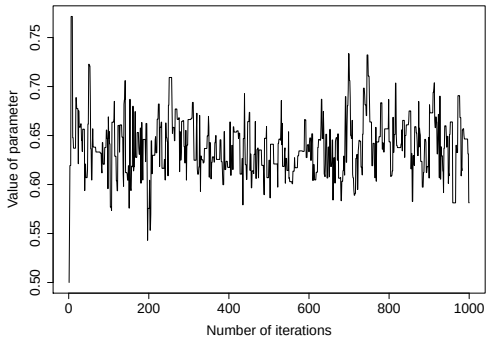


(c) ρ

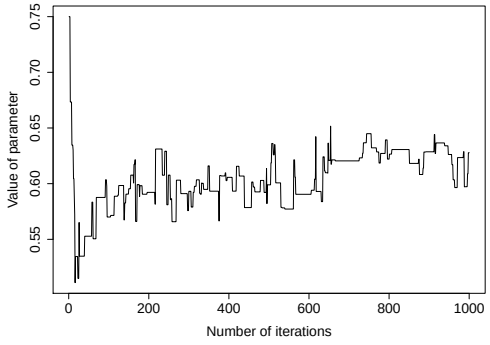
Figure A.1: Trace plots for the parameters λ, ϕ, ρ when the model utilises $\alpha = 1e - 05$



(a) λ

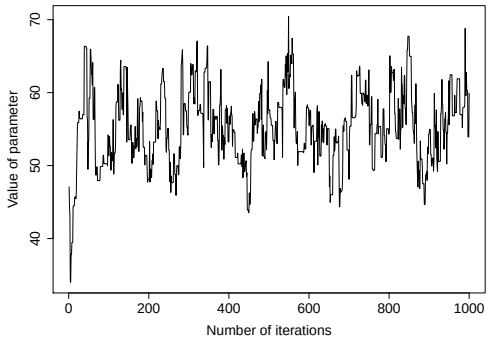


(b) ϕ

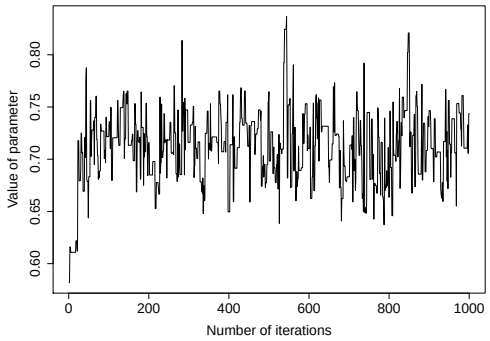


(c) ρ

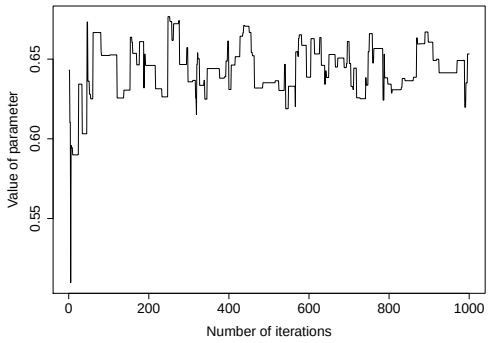
Figure A.2: Trace plots for the parameters λ, ϕ, ρ when the model utilises $\alpha = 1e - 10$



(a) λ

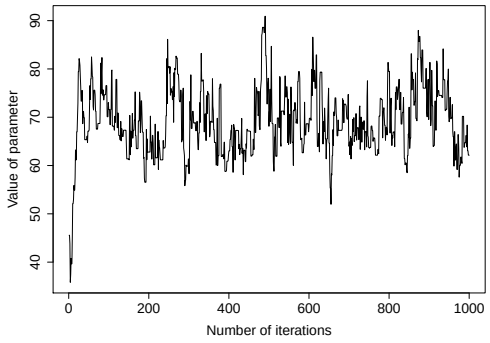


(b) ϕ

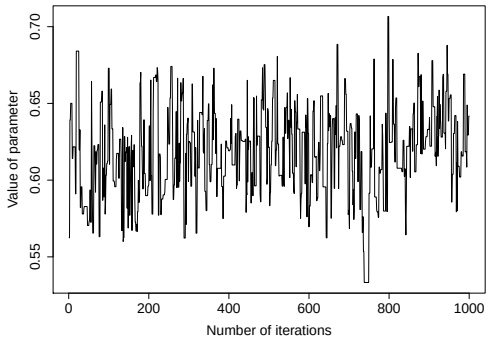


(c) ρ

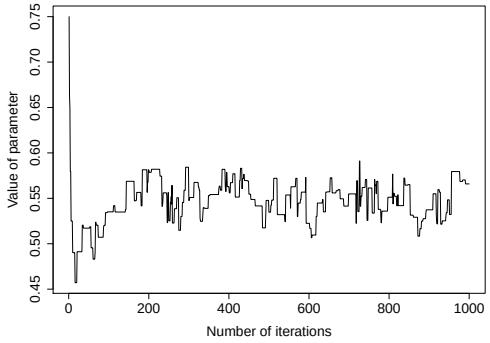
Figure A.3: Trace plots for the parameters λ, ϕ, ρ when the model utilises $\alpha = 1e - 15$



(a) λ



(b) ϕ



(c) ρ

Figure A.4: Trace plots for the parameters λ, ϕ, ρ when the model utilises $\alpha = 1e - 80$

References

Konstantinos Aboudolas and Nikolas Geroliminis. Perimeter and boundary flow control in multi-reservoir heterogeneous networks. *Transportation Research Part B: Methodological*, 55:265–281, 2013.

Konstantinos Aboudolas, Markos Papageorgiou, and E Kosmatopoulos. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 17(2):163–174, 2009.

Marcel R Ackermann, Johannes Blömer, and Christian Sohler. Clustering for metric and nonmetric distance measures. *ACM Transactions on Algorithms (TALG)*, 6(4):59, 2010.

Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2009.

Craig Anderson, Duncan Lee, and Nema Dean. Identifying clusters in bayesian disease mapping. *Biostatistics*, 15(3):457–469, 2014.

Zahid Ansari, MF Azeem, Waseem Ahmed, and A Vinaya Babu. Quantitative evaluation of performance and validity indices for clustering the web navigational sessions. *arXiv preprint arXiv:1507.03340*, 2015.

Francis R Bach and Michael I Jordan. Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, 7(Oct):1963–2001, 2006.

Jaime Barceló and Jordi Casas. Dynamic network simulation with AIMSUN. *Simulation approaches in transportation analysis*, pages 57–98, 2005.

Jaume Barceló et al. *Fundamentals of traffic simulation*, volume 145. Springer, 2010.

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

David M Blei and Peter I Frazier. Distance dependent chinese restaurant processes. *Journal of Machine Learning Research*, 12(Aug):2461–2488, 2011.

Konstantinos Blekas, Aristidis Likas, Nikolas P Galatsanos, and Isaac E Lagaris. Mixture model based image segmentation with spatial constraints. In *Signal Processing Conference, 2004 12th European*, pages 2119–2122. IEEE, 2004.

Adrian W Bowman. An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71(2):353–360, 1984.

Stephen P Brooks and Andrew Gelman. General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics*, 7(4):434–455, 1998.

Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

Jordi Casas, Jaime L Ferrer, David Garcia, Josep Perarnau, and Alex Torday. Traffic simulation with AIMSUN. In *Fundamentals of traffic simulation*, pages 173–232. Springer, 2010.

Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

Marie Chavent, Vanessa Kuentz-Simonet, Amaury Labenne, and Jérôme Saracco. Clustgeo: an R package for hierarchical clustering with spatial constraints. *Computational Statistics*, pages 1–24, 2017.

Huaihou Chen, Philip T Reiss, and Thaddeus Tarpey. Optimally weighted L2 distance for functional data. *Biometrics*, 70(3):516–525, 2014.

Mary Kathryn Cowles and Bradley P Carlin. Markov chain monte carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434): 883–904, 1996.

NAC Cressie. Statistics for spatial data (revised ed.) wiley. *New York*, 1993.

Gabriela B Cybis, Janet S Sinsheimer, Trevor Bedford, Andrew Rambaut, Philippe Lemey, and Marc A Suchard. Bayesian nonparametric clustering in phylogenetics: modeling antigenic evolution in influenza. *Statistics in medicine*, 37(2):195–206, 2018.

Carlos F Daganzo. Urban gridlock: Macroscopic modeling and mitigation approaches. *Transportation Research Part B: Methodological*, 41(1):49–62, 2007.

David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

Pedro Delicado, Ramón Giraldo, C Comas, and Jorge Mateu. Statistics for spatial functional data: some recent contributions. *Environmetrics*, 21(3-4):224–239, 2010.

M Downton. Comparing classifications: an evaluation of several coefficient of partition agreement. In *Proceedings of the meeting of the classification society, Boulder, CO, 1980*, 1980.

M Dumont, PA Reninger, A Pryet, G Martelet, B Aunay, and JL Join. Agglomerative hierarchical clustering of airborne electromagnetic data for multi-scale geological studies. *Journal of Applied Geophysics*, 157:1–9, 2018.

Joseph C Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

Jianqing Fan and Tsz Ho Yim. A crossvalidation method for estimating conditional densities. *Biometrika*, 91(4):819–834, 2004.

David Freedman and Persi Diaconis. On the histogram as a density estimator: L 2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57(4):453–476, 1981.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

Dani Gamerman and Hedibert F Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. Chapman and Hall/CRC, 2006.

Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

Andrew Gelman, Donald B Rubin, et al. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

Nikolas Geroliminis, Nan Zheng, and Konstantinos Ampountolas. A three-dimensional macroscopic fundamental diagram for mixed bi-modal urban networks. *Transportation Research Part C: Emerging Technologies*, 42:168–181, 2014.

John Geweke, Richard Meese, and Warren Dent. Comparing alternative tests of causality in temporal systems: Analytic results and experimental evidence. *Journal of Econometrics*, 21(2):161–194, 1983.

Soumya Ghosh, Andrei B Ungureanu, Erik B Sudderth, and David M Blei. Spatial distance dependent chinese restaurant processes for image segmentation. In *Advances in Neural Information Processing Systems*, pages 1476–1484, 2011.

Soumya Ghosh, Michalis Raptis, Leonid Sigal, and Erik B Sudderth. Nonparametric clustering with distance dependent hierarchies. In *UAI*, pages 260–269, 2014.

Ramón Giraldo, Pedro Delicado, and Jorge Mateu. Hierarchical clustering of spatially correlated functional data. *Statistica Neerlandica*, 66(4):403–421, 2012.

RA Haggarty, CA Miller, and EM Scott. Spatially weighted functional clustering of river network data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 64 (3):491–506, 2015.

Peter Hall, Jeff Racine, and Qi Li. Cross-validation and the estimation of conditional probability densities. *Journal of the American Statistical Association*, 99(468):1015–1026, 2004.

John A Hartigan. *Clustering algorithms*. Wiley, 1975.

John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 28(1):100–108, 1979.

Diman Hassan, Uwe Aickelin, and Christian Wagner. Comparison of distance metrics for hierarchical data in medical databases. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 3636–3643. IEEE, 2014.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

Desmond J Higham, Gabriela Kalna, and Milla Kibble. Spectral clustering and its use in bioinformatics. *Journal of computational and applied mathematics*, 204(1):25–37, 2007.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1): 193–218, 1985.

Dana E Ilea and Paul F Whelan. Color image segmentation using a spatial k-means clustering algorithm. 2006.

Antonio Irpino and Rosanna Verde. A new wasserstein based distance for the hierarchical clustering of histogram symbolic data. In *Data science and classification*, pages 185–192. Springer, 2006.

Julien Jacques and Cristian Preda. Functional data clustering: a survey. *Advances in Data Analysis and Classification*, 8(3):231–255, 2014.

Ronald J Janssen, Pasi Jylänki, and Marcel AJ van Gerven. Let's not waste time: Using temporal information in clustered activity estimation with spatial adjacency restrictions (caesar) for parcellating fmri data. *PloS one*, 11(12):e0164703, 2016.

Pablo A Jaskowiak, Ricardo JGB Campello, and Ivan G Costa. On the selection of appropriate distances for gene expression data clustering. In *BMC bioinformatics*, volume 15, page S2. BioMed Central, 2014.

Yuxuan Ji and Nikolas Geroliminis. On the spatial partitioning of urban transportation networks. *Transportation Research Part B: Methodological*, 46(10):1639–1656, 2012.

Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

Yunjae Jung, Haesun Park, Ding-Zhu Du, and Barry L Drake. A decision criterion for the optimal number of clusters in hierarchical clustering. *Journal of Global Optimization*, 25 (1):91–111, 2003.

Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

Jaejik Kim and L Billard. Dissimilarity measures for histogram-valued observations. *Communications in Statistics-Theory and Methods*, 42(2):283–303, 2013.

Ferenc Kovács, Csaba Legány, and Attila Babos. Cluster validity measurement techniques. In *6th International symposium of hungarian researchers on computational intelligence*. Citeseer, 2005.

Duncan Lee. A comparison of conditional autoregressive models used in bayesian disease mapping. *Spatial and spatio-temporal epidemiology*, 2(2):79–89, 2011.

Brian G Leroux, Xingye Lei, and Norman Breslow. Estimation of disease rates in small areas: a new mixed model for spatial dependence. In *Statistical models in epidemiology, the environment, and clinical trials*, pages 179–191. Springer, 2000.

Sheen S Levine and Robert Kurzban. Explaining clustering in social networks: Towards an evolutionary theory of cascading benefits. *Managerial and Decision Economics*, 27(2-3): 173–187, 2006.

Qi Li and Jeffrey S Racine. Nonparametric estimation of conditional cdf and quantile functions with mixed categorical and continuous data. *Journal of Business & Economic Statistics*, 26(4):423–434, 2008.

Kar Wai Lim, Wray Buntine, Changyou Chen, and Lan Du. Nonparametric bayesian topic modelling with the hierarchical pitman–yor processes. *International Journal of Approximate Reasoning*, 78:172–191, 2016.

Darlene Lu, Yorghos Tripodis, Louis Gerstenfeld, Serkalem Demissie, and Jonathan Wren. Clustering of temporal gene expression data with mixtures of mixed effects models with a penalized likelihood. *Bioinformatics*, 1:9, 2018.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

Ian C McDowell, Dinesh Manandhar, Christopher M Vockley, Amy K Schmid, Timothy E Reddy, and Barbara E Engelhardt. Clustering gene expression time series data using an infinite gaussian process mixture model. *PLoS computational biology*, 14(1):e1005896, 2018.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Max Mignotte. A de-texturing and spatially constrained k-means approach for image segmentation. *Pattern Recognition Letters*, 32(2):359–367, 2011.

Glenn W Milligan and Martha C Cooper. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 21(4):441–458, 1986.

Glenn W Milligan and Martha C Cooper. Methodology review: Clustering methods. *Applied psychological measurement*, 11(4):329–354, 1987.

Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

Peter Orbanz and Joachim M Buhmann. Nonparametric bayesian image segmentation. *International Journal of Computer Vision*, 77(1-3):25–45, 2008.

Slobodan Petrovic. A comparison between the silhouette index and the davies-bouldin index in labelling IDS clusters. In *Proceedings of the 11th Nordic Workshop of Secure IT Systems*, pages 53–64, 2006.

Jim Pitman et al. Combinatorial stochastic processes. Technical report, Technical Report 621, Dept. Statistics, UC Berkeley, 2002. Lecture notes for St. Flour course, 2002.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL https://www.R-project.org.

Adrian E Raftery and Steven Lewis. How many iterations in the gibbs sampler? Technical report, WASHINGTON UNIV SEATTLE DEPT OF STATISTICS, 1991.

James O Ramsay and Bernard W Silverman. *Applied functional data analysis: methods and case studies*. Springer, 2007.

William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

Carl Edward Rasmussen. The infinite gaussian mixture model. In *Advances in neural information processing systems*, pages 554–560, 2000.

Sylvia Richardson and Peter J Green. 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, 1997.

Christian P Robert, George Casella, and George Casella. *Introducing monte carlo methods with r*, volume 18. Springer, 2010.

Simon Rogers and Mark Girolami. *A first course in machine learning*. CRC Press, 2016.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

Mats Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, pages 65–78, 1982.

Mohammadreza Saeedmanesh and Nikolas Geroliminis. Clustering of heterogeneous networks with directional flows based on "snake" similarities. *Transportation Research Part B: Methodological*, 91:250–269, 2016.

Mohammadreza Saeedmanesh and Nikolas Geroliminis. Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks. *Transportation research part B: methodological*, 105:193–211, 2017.

Piercesare Secchi, Simone Vantini, and Valeria Vitelli. Spatial clustering of functional data. In *Recent Advances in Functional Data Analysis and Related Topics*, pages 283–289. Springer, 2011.

Ali Seyed Shirkhorshidi, Saeed Aghabozorgi, and Teh Ying Wah. A comparison study on similarity and dissimilarity measures in clustering continuous data. *PloS one*, 10(12): e0144059, 2015.

Richard Socher, Andrew Maas, and Christopher Manning. Spectral chinese restaurant processes: Nonparametric clustering based on similarities. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 698–706, 2011.

Martin A Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540, 1987.

Thaddeus Tarpey and Kimberly KJ Kinateder. Clustering functional data. *Journal of classification*, 20(1):093–114, 2003.

Yee Whye Teh. Dirichlet process. In *Encyclopedia of machine learning*, pages 280–287. Springer, 2011.

Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

Du-Ming Tsai and Ying-Hsiung Chen. A fast histogram-clustering approach for multi-level thresholding. *Pattern Recognition Letters*, 13(4):245–252, 1992.

ShengLi Tzeng, Christian Hennig, Yu-Fen Li, and Chien-Ju Lin. Distance for functional data clustering based on smoothing parameter commutation. *arXiv preprint arXiv:1604.02668*, 2016.

Jan-Willem van Dam and Michel Van De Velden. Online profiling and clustering of facebook users. *Decision Support Systems*, 70:60–72, 2015.

Deepak Verma and Marina Meila. Comparison of spectral clustering methods. *Advances in neural information processing systems*, 15:38, 2003.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4): 395–416, 2007.

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

Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

Ke Xie, Jin Wu, Wankou Yang, and Changyin Sun. K-means clustering based on density for scene image classification. In *Proceedings of the 2015 Chinese Intelligent Automation Conference*, pages 379–386. Springer, 2015.

Shan Zeng, Rui Huang, Zhen Kang, and Nong Sang. Image segmentation using spectral clustering of gaussian mixture models. *Neurocomputing*, 144:346–356, 2014.

Wei Zhang, Xiangzhong Fang, Xiaokang Yang, and QM Jonathan Wu. Spatiotemporal gaussian mixture model to detect moving objects in dynamic scenes. *Journal of Electronic Imaging*, 16(2):023013, 2007.

Zhongheng Zhang, Fionn Murtagh, Sven Van Poucke, Su Lin, and Peng Lan. Hierarchical cluster analysis in clinical research with heterogeneous study population: highlighting its visualization with r. *Annals of translational medicine*, 5(4), 2017.

Xi Zhu and Diansheng Guo. Mapping large spatial flow data with hierarchical clustering. *Transactions in GIS*, 18(3):421–435, 2014.