UNIVERSITY OF GLASGOW

Spatial Clustering Algorithms for Areal Data

by

Cunyi Wang

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
School of Mathematics and Statistics
Supervised by Nema Dean

December 19, 2018

Declaration of Authorship

I, Cunyi Wang, declare that this thesis titled, 'Spatial Clustering Algorithms for Areal 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 this University.
- 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: _____

"It always seems impossible until it is done"

Nelson Mandela

Abstract

The main aim of this thesis is to develop new spatial clustering approaches which can simultaneously identify different areal clusters and guarantee their geographical contiguity. The second aim is to adjust the finite mixture model in order to cope with the issues caused by outliers or singletons (clusters with only one object). In addition, the thesis also aims to extend the applications of these newly proposed spatial clustering techniques from univariate to multivariate space.

In Chapter 1, I will review some available clustering techniques in grouping spatial data and will also introduce different types of clustering data and the Glasgow housing market data which will be used in the thesis's application. At the end of this chapter, I will outline the structure of this thesis. In Chapter 2, I will give the general statistical theory and inference methodologies used across this thesis, including frequentist and Bayesian statistical inferences, multidimensional scaling and the Procrustes transformation. In Chapter 3, I will introduce techniques that could be used in transforming between two types of clustering data introduced in Chapter 1. Chapter 4 will define some cluster and graph terminology and will also introduce different clustering techniques, such as hierarchical clustering, Chameleon hierarchical clustering and model-based clustering. In this chapter, I will also cover some techniques used in cluster comparisons, methods for number of clusters decisions and number of dimensions decisions. Chapter 6 will introduce more detail about spatial hierarchical clustering. The simulation results from spatial hierarchical clustering will be used as the reference results for comparison with the results from the proposed novel spatial clustering techniques in later chapters.

The newly proposed clustering techniques, Chameleon spatial hierarchical clustering, spatially constrained finite mixture model with noise component or with priors and spatially constrained Bayesian model-based clustering with dissimilarities, in clustering areal data will be introduced in Chapters 7, 8 and 9 respectively. Also, the simulations and the application in Glasgow housing market will be given at the end of each of these three chapters. Chameleon spatial hierarchical clustering combined the spatial contiguity with Chameleon hierarchical clustering, so areas grouped together are spatially contiguous. Spatially constrained finite mixture models incorporate the spatial prior distribution into the classical finite mixture model to deal with the spatial contiguity issue. Also, I will make the spatially constrained finite mixture model more robust by incorporating a uniform distribution to model the noise points or adding prior distributions to the model. In Chapter 9, I will add a spatial prior to the model-based clustering with

dissimilarities model and then will use a Bayesian approach to obtain a spatial contiguous clustering. Chapter 10 will be conclusions and discussion about the newly proposed clustering methods.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Dr Nema Dean for the continuous support of my Ph.D study and related research, for her patience, motivation and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

In addition, I would like to express my gratitude to Prof Gwilym Pryce for the data measurement he proposed and also Dr Craig Anderson for his motivation in spatial clustering and the other favors provided by the other staff in the Maths and Statistics department.

I would like to thank my friends for accepting nothing less than excellence from me. Last but not the least, I would like to thank my family: my parents and those for supporting me spiritually throughout writing this thesis and my life in general.

Contents

List of Figures

List of Tables

Abbreviations

AIC	**A**kaike **I**nformation **C**riterion
ARI	**A**djusted **R**and **I**ndex
ASW	**A**verage **S**ihouette **W**idth
BCSS	**B**etween **C**luster **S**um of **S**quares
BFS	**B**readth **F**irst **S**earch
BIC	**B**ayesian **I**nformation **C**riterion
BMBCD	**B**ayesian **M**odel- **B**ased **C**lustering with **D**issimilarities
BMDS	**B**ayesian **M**ulti**d**imensional **S**caling
CH	**C**alinski and **H**arabasz
CMDS	**C**lassical **M**ulti**d**imensional **S**caling
CPED	**C**ross **P**rice **E**lasticity of **D**emand
CPEP	**C**ross **P**rice **E**lasticity of **P**rice
CSHC	**C**hameleon **S**patial **H**ierarchical **C**lustering
EM	**E**xpectation **M**aximization
GEM	**G**eneralized **E**xpectation **M**aximization
HCM	**H**eavy **C**lique **M**atching
HEM	**H**eavy **E**dge **M**atching
HER	**H**yper **E**dge **R**efinement
K-NN	**K**-**N**earest **N**eighbours
LEM	**L**ight **E**dge **M**atching
McMC	**M**arkov **c**hain **M**onte **C**arlo
MDS	**M**ulti**d**imensional **S**caling
PH	**P**earson version of **H**ubert's Γ
RM	**R**andom **M**atching
SC-MDS	**S**plit and **C**ombine **C**lassical **M**ulti**d**imensional **S**caling

SHC	**S**patial **H**ierarchical **C**lustering
SVD	**S**ingular **V**alue **D**ecomposition
TDR	**T**rue **D**iscovery **R**ate
TPR	**T**rue **P**ositive **R**ate
WCSS	**W**ithin **C**luster **S**um of **S**quares

Chapter 1

Introduction

Cluster analysis is a technique to assign objects into groups, where objects put into the same cluster share similar characteristics or are closer to each other than objects assigned to different clusters. So the main aim of clustering is to identify the underlying group structure of the data. There are many different clustering techniques that can be used to group different types of data.

Agglomerative hierarchical clustering is a traditional and well-developed algorithm in grouping data [103]. It groups the objects by constructing a hierarchy of clusters based on the dissimilarities between objects. In the beginning stage, all the objects are their own clusters, then at each iteration, the most similar two clusters are merged into one new cluster and the cluster distances between pairs of clusters are recalculated. At the end, all the objects are in one cluster. A single result from hierarchical clustering is usually achieved by cutting the hierarchy, then the objects coming from the same branch will be grouped into one cluster.

In Karypis et al. [59], the authors proposed a method called Chameleon hierarchical clustering for grouping large-scaled datasets with diverse cluster shapes by using the internal characteristics of clusters. Chameleon hierarchical clustering mainly consists of three stages: K-nearest neighbours (K-NN) graph stage, graph partitioning stage and merging stage. In the beginning, we will construct a K-nearest neighbour graph by identifying and connecting the most similar K neighbours to each other. At the second stage, the finest similarity graph will be coarsened into a small-sized graph (with less number of objects, also called the coarser graph), this procedure will be repeated several times and we call the graph formed at the last step as the coarsest graph. The initial partitioning is formed based on this coarsest graph. The next step is to project the coarsest graph back to the finest graph and a refinement technique will be applied

to potentially change the cluster membership of the bordering objects to modify the partitioning at the same time. At the last stage, merging the most similar partitions at each step into one and give a hierarchy of possible clusterings.

Clustering can also be done by taking advantage of the finite mixture model framework [79]. If the coordinate (observations $\times$ variables) data are available, the finite mixture model assumes that the data are from a set of J component densities with some unknown parameters $\boldsymbol{\theta}$ and some set of mixing proportions. The posterior probabilities of belonging to different components can then be used to group the data.

A different type of mixture clustering model for dissimilarity data is model-based clustering with dissimilarities [34]. It combines two ideas. Firstly, it uses dissimilarities to estimate the latent positions of objects in a Euclidean space. Secondly, it assumes the latent positions are generated from a finite mixture of multivariate Gaussian distributions and each Gaussian distribution corresponds to one cluster.

One of the applications I will examine is grouping areal units in the housing market to identify potential submarkets. Housing market data are different from other types of data. It is spatial data (non-spatial attributes combining with spatial attributes), which means that it is fundamentally geographically connected in nature. The traditional clustering techniques can only group data based on their characteristics usually without taking spatial information into account. So one of the aims in this thesis is to develop different cluster techniques for grouping spatial data, where the areal units grouped together should not only share similar characteristics, but be geographically contiguous as well.

Based on this idea, the spatial hierarchical clustering algorithm was proposed by Anderson et al. [13]. Spatial hierarchical clustering combines the areal geographical information with hierarchical clustering. In spatial hierarchical clustering, two clusters with the minimum distance or dissimilarity may not be merged unless they are also geographically connected [13]. The novelty of this method over the traditional hierarchical clustering is that two clusters can only be joined together if they share at least one common border or their membership groups share some common borders. Suppose we have two clusters both with more than one object in each cluster. If these two clusters are the clusters with the minimum dissimilarity in the current iteration, and one of the objects in one cluster is geographically connected to at least one of the objects from the other cluster, then these two clusters will be merged, otherwise they will not be. So, based on the novelty of spatial hierarchical clustering, I will extend some other clustering techniques in a similar manner.

1.1 Clustering Data

There are two common types of data which are used in cluster analysis. One is coordinate data. The other type is the relationship between pairs of objects, resulting in dissimilarity or similarity data. Hierarchical clustering and model-based clustering with dissimilarities use dissimilarity data, while Chameleon hierarchical clustering uses similarity data. We can transform the dissimilarity into similarity data easily and vice versa. If coordinate data are available, we can also create a distance matrix by transforming the coordinate data into dissimilarity data.

1.1.1 Dissimilarity or Similarity Data

In this thesis, the dissimilarity data used in the simulations are calculated by using distances between pairs of objects. Examples of common types of distances will be introduced in Section 4.2. In terms of similarity data in Chameleon spatial hierarchical simulations, they are transformed from dissimilarity data. We can apply any monotonically decreasing function (e.g. reciprocal) to dissimilarity data to achieve this. In the main application, we will introduce a new type of similarity data, Cross Price Elasticity of Price (CPEP) data, which can be applied to substitutable items. In CPEP data, all values are ranging from 0 to 1. 1 indicates items are completely substitutable, 0 means they are not substitutable at all. The details for CPEP will be covered in Section 1.3.1.

1.1.2 Coordinate Data

Coordinate data are data with information on different dimensions in a space, which can form the object configuration in the space. However, this type of data is not always available. For some scenarios, the only accessible data are dissimilarity or similarity data. In order to cope with this issue, we can apply multidimensional scaling algorithms to dissimilarity data in order to estimate coordinate data. In Chapter 3, I will introduce three types of multidimensional scaling techniques, classical multidimensional scaling, Bayesian multidimensional scaling and split-and-combine classical multidimensional scaling.

1.2 Spatial Data

Spatial data are data with additional geographical attributes [101]. There are three main types of spatial data: geostatistical data, areal data and point data [101]. The basic form of geostatistical data is in the form of

$$(\mathbf{z}_i, \mathbf{y}_i) \text{ for } i = 1 \cdots, N,$$

$\mathbf{z}_i$ is used to identify a spatial location, which could for example be longitude and latitude in degrees or Easting and Northing in metres [32]. $\mathbf{y}_i$ is a characteristic measure at that location. N is the number of observations. Theoretically speaking, data could be observed at infinite number of locations, but in practice, the data are collected at a finite number of user specified locations. For example, if we want to measure the air pollution concentrations in a certain region, we usually select positions for pollution monitors sparsely with longer distances between pairs of positions, because two closely located pollution monitors will get similar results.

Areal data are data which are recorded on an overall region which is further partitioned into a finite number of non-overlapping areal units with well-defined boundaries. For example, if a region $\boldsymbol{\mathcal{A}}$ can be fully divided into N non-overlapping areal units, then the areas in the region $\boldsymbol{\mathcal{A}}$ can be expressed as $\boldsymbol{\mathcal{A}} = \{\mathcal{A}_1, \cdots, \mathcal{A}_N\}$. Data is then recorded on these areas, e.g. k^{th} covariate $\mathbf{x}_k = (x_{1k}, \cdots, x_{Nk})$.

Point data is similar to geostatistical data, but the difference between these two types of data is whether the locations are user-specified or part of the random process. An example of point data is the positions of trees in a forest.

In this thesis, I will use areal data. The spatial information for areal data is stored in a neighbourhood matrix $\mathbf{W}$. The neighbourhood matrix $\mathbf{W}$ is an $N \times N$ symmetric binary matrix, whose entries show the connection between pairs of areas. If two areas share a boundary, then the corresponding entry in $\mathbf{W}$ will be 1. Otherwise, if two areas have no geographical connections, then the corresponding entry will be 0.

1.3 The Glasgow Housing Market

The housing market is a complex and constantly evolving structure. It is hard to accurately capture its changing patterns, because the house trade is different from other differentiated goods in that it is fundamentally spatially contiguous in nature [28]. Moreover, the housing market is not a global or national market, it is highly localized, with different areas following different patterns and affected by different factors [28]. In addition to this, the housing market has a large effect on the economy and the residents. As housing and properties constitute a large proportion of the world wealth, it represents an important part of the macroeconomy [48]. So all of these desirable attributes mean that the analysis of the housing market may be challenging but ultimately necessary. More details about Glasgow housing market and its data will be introduced in Chapter 5.

1.3.1 Cross Elasticity of Price (CPEP)

In this thesis, I will explore the substitutability of Glasgow housing market areas. However, the measurement of substituability in the housing market is a challenge in grouping the areal units. House prices in equilibrium are set by the balance of supply and demand [78]. When the supply of properties increases, house prices will drop, then the demand for properties increases greatly. Otherwise, if the supply of house properties shrinks, house prices will increase, then the demand for properties decreases. However, if supply persistently fails to respond to higher prices or the house prices are driven by some non-fundamental factors of housing, such as property investment or trend-chasing, then finding a quantified measurement of the substitutability in the housing market will be a challenging task.

In economics, cross price of elasticity of demand (CPED) is used to measure the substitutability among goods. CPED measures the responsiveness of demand of for one good following a change in the price of another related good [19]. If CPED is a negative value, then these two goods are complementary, otherwise, the two goods are substitutable.Unrelated goods will always have a zero CEPD. However, CPED analysis requires us to estimate how the demand for one good is affected by the selling price of another. In order to avoid the estimation of demand, Gwilym Pryce [94] proposed a new measurement, Cross Price Elasticity of Price (CPEP), as a measurement of substitutability. The closer to 1

the CPEP is, the more substitutable the two areas will be, e.g. for a pair of areas i and j implies that if an increase in price of area i will lead to an increase in the demand for area j, if the supply of area j is not sufficient, then this will cause a rise in the price of area j. So the rise in price for one area leads to the rise in price for the other substitutable areas. Therefore if CPEP is approaching 1, then it means that area i and area j are very substitutable based on the pairs of areal units' time series trends, which means the house prices of these pairs of areal units are varying in a similar pattern. If CPEP is close to zero, then area i and area j are not substitutable based on the pairs of areal units' time series trends, which means the time series trends of the pairs of areal units are very different. So in this project, CPEP will be used as the numeric measurement of substitutability of the time series trends between pairs of areal units.

The definition of CPEP is as follows. If $\mathbf{y}_i = \{y_{i1} \cdots, y_{iT}\}$ are the time series data in administrative unit i and $\mathbf{y}_j = \{y_{j1} \cdots, y_{jT}\}$ are the time series data over the same time periods in administrative unit j, the log regression models on proportional change between the two areal units are $\log(y_{it}/y_{i1}) = \alpha_{ij} + \beta_{ij} \log(y_{jt}/y_{j1}) + \varepsilon_{ijt}$ and $\log(y_{jt}/y_{j1}) = \alpha_{ji} + \beta_{ji} \log(y_{it}/y_{i1}) + \varepsilon_{jit}$ separately. The expression of CPEP between this pair of areal units is defined using the following algorithm:

1. If $\beta_{ij} \leq 0$, then $\text{RCPEP}_{ij} = 0$, similarly for β_{ji} and RCPEP_{ji};
2. If $0 < \beta_{ij} < 1$, then $\text{RCPEP}_{ij} = \beta_{ij}$, similarly for β_{ji} and RCPEP_{ji};
3. If $\beta_{ij} > 1$, then $\text{RCPEP}_{ij} = \frac{1}{\beta_{ij}}$, similarly for β_{ji} and RCPEP_{ji};

$$\text{CPEP}_{ij} = \text{CPEP}_{ji} = \max\left(\text{RCPEP}_{ij}, \text{RCPEP}_{ji}\right)$$

where RCPEP_{ij} is the Regression of Cross Price Elasticity of Price. CPEP estimated by regressing log two areas' time series data. The estimated value of CPEP is not used for predicting housing prices, but is used as a substitutable measurement. So the assumptions of the regression are not necessary to meet. More details of CPEP in the Glasgow housing market will be introduced in Chapter 5.

1.4 Aims and Structures of the Thesis

The main aim of this thesis is to develop new spatial clustering approaches which can simultaneously identify different areal clusters and guarantee their geographical

contiguity. The second aim is to cope with the issues caused by outliers or singletons (partitions with only one object). In addition, the thesis also aims to extend the applications of these newly proposed spatial clustering techniques from univariate to multivariate space.

In Chapter 2, I will outline the general statistical theory and inference methodologies used across this thesis, including frequentist and Bayesian statistical inferences, multidimensional scaling and the Procrustes transformation. As mentioned previously, in cluster analysis, there are two main types of data. One type of the data is the between objects relationship data, such as dissimilarity or similarity matrices. The other type of the data is coordinate data. So in Chapter 3, I will introduce techniques that could be used in transforming between these two types of data. Chapter 4 will define some cluster and graph terminology and will also introduce different clustering techniques, such as hierarchical clustering, model-based clustering and Chameleon hierarchical clustering. One of the challenges faced in clustering real datasets is how to decide the true number of clusters and the number of dimensions where necessary, as these are usually unknown information to the researchers. So Chapter 4 will also introduce some techniques used in cluster comparisons, methods for number of clusters decisions and number of dimensions decisions. Chapter 6 will introduce more detail about spatial hierarchical clustering. The simulation results from spatial hierarchical clustering will be used as the reference results for comparison with the results from the proposed novel spatial clustering techniques in later chapters.

The newly proposed spatial clustering techniques will be introduced in Chapters 7, 8 and 9. Techniques proposed in Chapters 7 and 9 will use the first type of data (similarity or dissimilarity matrix) when clustering objects, while the technique introduced in Chapter 8 will use coordinate data.

The Chameleon spatial hierarchical clustering proposed in Chapter 7 is the most similar spatial clustering technique to spatial hierarchical clustering, which takes advantage of the graph characteristics. It is composed of three stages: K-NN graph stage, graph partitioning stage and merging stage. All three stages will be illustrated step by step by using an illustrative example. There are several parameters that need to be set in Chameleon spatial hierarchical clustering. These will be chosen from a sequence of values by comparing clustering performance across different simulation scenarios in order to give guidelines about these in general.

The second newly proposed spatial clustering technique is spatial finite mixture models which will cope with the problems caused by noise points, either including prior terms to

regularize covariance matrices or modeling noise points in a separate model component [15].

The last spatial clustering technique introduced in Chapter 9 is evolved from the spatial finite mixture model with prior terms in Chapter 8, but in a fully Bayesian approach. The Generalized Expectation Maximization (GEM) estimation technique used in Chapter 8 can only give point estimates of the parameters of interest and does not quantify the full posterior distribution. So in Chapter 9, a fully Bayesian approach will be used to estimate the parameters in the model. Although the number of clusters will be set before clustering, the number of clusters may be reduced and unnecessary clusters will be removed at the end.

The application of Glasgow housing market will be used as a running example at the end of Chapters 6, 7, 8 and 9. As mentioned previously, the newly proposed clustering techniques will not be limited to univariate space, so I will incorporate another dataset, the median gross household income of Glasgow intermediate zones in 2010, as another application in order to measure the performance of spatial finite mixture clustering techniques in multivariate space.

The final part of the thesis will be the conclusion chapter, Chapter 10. All newly proposed spatial clustering techniques in the earlier chapters will be compared based on the simulation framework (patterns, number of outliers or levels of variances and means). The advantages and disadvantages of these techniques will be summarized and future work and caveats will also be discussed.

Chapter 2

Statistical Background

This chapter will outline the general statistical theory and inference methodologies used across this thesis. Section 2.1 will define two types of statistical inference: frequentist and Bayesian inference. Sections 2.2 to 2.4 are related to the applications of these inferences on finite mixture models. Section 2.5 will briefly introduce multidimensional scaling, which is a technique to produce coordinate data based on dissimilarities. Finally, in Section 2.6, I will discuss the Procrustes transformation, which is used to deal with the issue caused by lack of uniqueness in multidimensional scaling solutions.

2.1 Statistical Inference

Statistical inference is a set of procedures for using sample data to infer population parameters. In real data analysis, as it is usually hard to collect the full population data, it is impossible to exactly calculate the population parameters. In order to deal with this issue, statisticians use different sampling methods, for example, stratified sampling or cluster sampling, to collect representative samples from the population, then use this sample data to estimate values of the population parameters.

Statistical inference can be mainly divided into two categories, one is frequentist inference, the other is Bayesian inference. Frequentist inference uses the idea of properties of repeated sampling from the population to estimate the population parameters. The expectation maximization (EM) algorithm [80] is usually used in a frequentist inference manner, and will be introduced in Section 2.3. However, the EM algorithm can only give point or interval estimates of the parameters. It fails to fully investigate the parameter

distributions. In order to overcome this, Bayesian inference will be explored in detail in Section 2.4.

2.1.1 Frequentist Inference

In frequentist inference, it assumes parameters are unknowable, but can be estimated from a repeatable process [82]. For example, assuming the probability of getting a head in an experiment of tossing an uneven coin is unknown, but this probability can be estimated by conducting the same experiment many times, then the frequency of getting heads over the total number of experiments will be used to calculated it. Specifically, when the number of experiments is large enough, then the estimated value will be approximately equal to the actual probability of getting a head.

A crucial function in statistical inference which is used in estimating parameters is the likelihood function. It is a function of model parameters given observed data [82] and is usually denoted as $\mathcal{L}(\boldsymbol{\theta}; \mathbf{X})$. In frequentist inference, it can be explained as the plausibility/ likelihood of a parameter value after samples are observed. The optimal parameter value will be the one found that locally or globally maximizes this likelihood function [57]. This procedure might involve finding the derivative function of $\mathcal{L}(\boldsymbol{\theta}; \mathbf{X})$ when it is available. Generally speaking, it can usually be relatively easier to get the derivative of the log likelihood function compared to the likelihood function itself [54]. Since the logarithm is a monotonically increasing function, the value which maximizes the log likelihood function will also be the one maximizes its likelihood function. So it is common to maximize the log likelihood function instead of likelihood function itself.

In frequentist inference, there are two main types of estimations, one is a point estimate, the other is an interval estimate. Point estimates use only one single value to describe the population parameter, while interval estimates use a range of values to describe the population parameter by limiting those values between two boundaries. This type of estimation can also give the confidence level of the interval estimates containing the true parameter in a repeatable process.

2.1.2 Bayesian Inference

The main alternative inference to frequentist is Bayesian inference, which involves updating beliefs based on observations given prior information. This process follows Bayes' Theorem [16]. In Bayes' Theorem, for two dependent events A and B in a sample space, the probability of getting event A given event B has occurred is expressed as:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}.$$

$P(A)$ is the prior probability of event A happening without other information. $P(B \mid A)$ is the probability of observing event B given event A has occurred. $P(B)$ is the probability of observing event B without other information. $P(A \mid B)$ is the posterior probability, given event B has occurred that A happens.

For continuous random variables X and Y, Bayes' Theorem can be re-expressed as:

$$f_{X|Y}(x \mid y) = \frac{f_{Y|X}(y \mid x) \cdot f_X(x)}{f_Y(y)}.$$

$f_X(x)$ is the prior distribution of event X happening without other information. $f_{Y|X}(y \mid x)$ is the conditional distribution of observing event Y given event X has occurred. $f_Y(y)$ is the prior distribution of observing event Y without other information. $f_{X|Y}(x \mid y)$ is the posterior distribution, given event Y has occurred that X happens.

The crucial idea of Bayesian inference is to update beliefs by combining new evidence with prior information. This process can be repeated iteratively, after observing some evidence, the resulting belief can then become new prior information, and this new belief will be updated when further new evidence is collected.

In Bayesian inference, the prior distribution $f(\boldsymbol{\theta})$ is the distribution of $\boldsymbol{\theta}$ without other information. The likelihood function $\mathcal{L}(\boldsymbol{\theta}; \mathbf{X})$ is the likelihood of the parameters $\boldsymbol{\theta}$ given data $\mathbf{X}$. The posterior distribution $f(\boldsymbol{\theta} \mid \mathbf{X})$ is the parameter distribution given the data and it can be expressed as:

$$f(\boldsymbol{\theta} \mid \mathbf{X}) = \frac{\mathcal{L}(\boldsymbol{\theta}; \mathbf{X}) \cdot f(\boldsymbol{\theta})}{f(\mathbf{X})}. \tag{2.1}$$

We can see that $f(\mathbf{X})$ is not a function of $\boldsymbol{\theta}$, so (2.1) can be re-written as:

$$f(\boldsymbol{\theta} \mid \mathbf{X}) \propto \mathcal{L}(\boldsymbol{\theta}; \mathbf{X}) \cdot f(\boldsymbol{\theta}). \tag{2.2}$$

2.1.2.1 Prior Distributions

Prior distributions are unconditional distributions of the data, indicating the probability distribution of the parameters without other information [72]. Prior distributions usually rely on other parameters called hyperparameters [72]. There are three types of prior distributions: informative priors, weakly informative priors and uninformative priors [72].

Informative priors usually express specific and definite information about parameters [33]. For example, if we are curious about the next hour's stock price, it is reasonable to use the current price as the mean for the future price, then the current stock price will be the informative prior for its future price.

Uninformative priors use more general or vague information, such as uninformative priors can express "objective" information such as "the variable is positive" or "the variable is less than some limit". The simplest and oldest rule for determining a non-informative prior is the principle of indifference, which assigns equal probabilities to all possibilities [6]. One type of commonly used uninformative priors are called Jefferys priors [56].

The Jefferys prior [56] is proportional to the square root of the Fisher information determinant of the likelihood, which is given by

$$f(\boldsymbol{\theta}) \propto \sqrt{\det I(\boldsymbol{\theta})},$$

where $I(\boldsymbol{\theta})$ is the Fisher information which is in the form of

$$I(\boldsymbol{\theta}) = \mathrm{E}\left[\left(\frac{\partial}{\partial \boldsymbol{\theta}} \log \mathcal{L}(\boldsymbol{\theta}; \mathbf{X})\right)^2\right].$$

For example, the Jefferys prior of the mean μ in a Gaussian distribution,

$$f(X \mid \mu) = \frac{\exp\left(-(X-\mu)^2/2\sigma^2\right)}{\sqrt{2\pi\sigma^2}}$$

is given by

$$\begin{aligned} f(\mu) \propto \sqrt{\mathrm{I}(\mu)} &= \sqrt{\mathrm{E}\left[\left(\frac{\partial}{\partial \mu} \log \mathcal{L}(\mu; X)\right)^2\right]} \\ &= \sqrt{\mathrm{E}\left[\left(\frac{X-\mu}{\sigma^2}\right)^2\right]} \\ &= \sqrt{\int_{-\infty}^{+\infty} f(X \mid \mu) \left(\frac{X-\mu}{\sigma^2}\right)^2 dX} \\ &= \sqrt{\frac{1}{\sigma^2}} \\ &\propto 1 \end{aligned}$$

Conjugate priors were proposed by Howard Raiffa and Robert Schlaifer [98]. If the posterior distribution and the prior distribution are in the same distribution family, then the prior distribution is called a conjugate prior of this likelihood function. All the distributions in the exponential family have conjugate priors. This means that if a likelihood function can be expressed in the form of

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{X}) = h(\mathbf{X}) \exp\left(\eta(\boldsymbol{\theta}) \cdot T(\mathbf{X}) - A(\boldsymbol{\theta})\right),$$

where $h(\mathbf{X})$ and $T(\mathbf{X})$ are the functions of $\mathbf{X}$, $\eta(\boldsymbol{\theta})$ and $A(\boldsymbol{\theta})$ are the functions of $\boldsymbol{\theta}$, then its conjugate prior exists. For example, assuming $X \sim$Binomial(n, p), if X is the number of successes, n is the total number of trials, p is the probability of successes, then its likelihood function can be expressed as:

$$\begin{aligned} \mathcal{L}(p; n, X) &= \binom{n}{X} p^X (1-p)^{n-X} \\ &= \binom{n}{X} \exp\left(X \log p + (n-X) \log(1-p)\right), \end{aligned}$$

which follows the form of exponential family, $h(X) = \binom{n}{X}$, $\eta(p) = \log\left(\frac{p}{1-p}\right)$, $T(X) = X$ and $A(p) = n \log(1-p)$. If we choose Beta(α, β) as the prior distribution of p,

$$f(p) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha, \beta)}$$

and infer its posterior distribution by applying Bayes' Theorem
$f(p \mid X) \propto f(p) \cdot \mathcal{L}(p; n, X)$, then the posterior distribution of p can be expressed as:

$$\begin{aligned} f(p \mid X) &\propto \frac{p^{\alpha-1}(1-p)^{\beta-1}\binom{n}{X}p^X(1-p)^{n-X}}{B(\alpha,\beta)} \\ &\propto \frac{p^{\alpha-1+X}(1-p)^{\beta-1+n-X}}{B(\alpha,\beta)}, \end{aligned}$$

which also turns out to be a Beta distribution, but with different parameters, Beta$(\alpha + X, \beta + n - X)$. So the Beta prior distribution of a Binomial likelihood is a conjugate prior. Conjugate priors are commonly used in Gibbs sampling which will be introduced in Section 2.4.2.

2.2 Finite Mixture Models

The idea of finite mixture models was proposed by the biometrician Karl Pearson [91] and is used to solve problems when data are generated from a distribution containing different component densities with some set of mixing proportions. Given $\mathbf{X}_i = (X_{i1}, \cdots, X_{iP})$ as the i^{th} data configuration in a P dimensional space and its density function $f(\mathbf{X}_i \mid \boldsymbol{\theta})$ on $\mathcal{R}^P$ [79], the finite mixture model for this data can be expressed as:

$$f(\mathbf{X}_i \mid \boldsymbol{\theta}) = \sum_{j}^{J} p^j f_j(\mathbf{X}_i \mid \boldsymbol{\theta}_j), \tag{2.3}$$

where $f_j(\mathbf{X}_i \mid \boldsymbol{\theta}_j)$ is the density function of the j^{th} component and p^j is the prior mixture probability of belonging to the j^{th} component,

$$0 < p^j \leq 1 \; j = 1 \cdots, J$$

and

$$\sum_{j=1}^{J} p^j = 1.$$

The interpretation of the finite mixture model can be thought as the probability $\mathbf{X}_i$ is generated from one of J component distributions. If Z_i is a categorical random variable modeled by a multinomial distribution, which takes values from 1 to J with probabilities p^1 to p^J, then $f_j(\mathbf{X}_i \mid \boldsymbol{\theta}_j)$ can be re-expressed as $f(\mathbf{X}_i \mid Z_i = j, \boldsymbol{\theta}_j)$.

2.3 Expectation Maximization Algorithm

The expectation maximization (EM) algorithm is usually used to estimate parameters when directly locating the maximum in a likelihood function is difficult due to intractability or missing data [80]. In finite mixture models, we have observed data $\mathbf{X} = (\mathbf{X}_1, \cdots, \mathbf{X}_N)$ and model-related unknown parameters $\boldsymbol{\theta} = \{\mathbf{p}, \sum, \boldsymbol{\mu}\}$, where $\mathbf{p} = \{p_1, \cdots, p_J\}$, $\sum = \{\sum_1, \cdots, \sum_J\}$ and $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_J\}$. The difficulty in maximizing the log likelihood function directly is we do not know which component generated each data point $\mathbf{X}_i$. If we know the latent membership variables, we can easily group data by components, then estimate the component parameters separately [105]. We can use the EM algorithm to estimate the parameters in these statistical models by including latent variables $\mathbf{Z} = (\mathbf{Z}_1, \cdots, \mathbf{Z}_N)$ to indicate which component generated which data points [80].

Given the observed data, unknown model related parameters, the latent variables and a complete data likelihood function $\mathcal{L}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})$, the maximum likelihood estimate (MLE) of the unknown parameters $\boldsymbol{\theta}$ is determined by the marginal likelihood of the observed data [80], which is expressed as:

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{X}) = \int \mathcal{L}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) d\mathbf{Z}.$$

However, if the latent variables $\mathbf{Z}$ can take an extremely large number of possible combinations, then this calculation might not be possible, so the EM algorithm seeks the MLE of $\mathcal{L}(\boldsymbol{\theta}; \mathbf{X})$ by iteratively applying the following steps [80]:

For a given initial value $\boldsymbol{\theta}^{(0)}$. We start with $t = 1$.
The estimation at the t^{th} iteration is as follows:

1. E Step: If the complete data is not fully observed (e.g. $\mathbf{Z}$'s in the finite mixture models are unknown information), then the complete data log likelihood function will be replaced by its conditional expectation given the observed data $Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)}\right)$, we calculate $Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)}\right) = \mathrm{E}\left(\log \mathcal{L}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) \mid \boldsymbol{\theta}^{(t-1)}, \mathbf{Z}\right)$ at the t^{th} iteration to estimate $\mathbf{Z}$.

2. M Step: Choose $\boldsymbol{\theta}^{(t)}$ which can maximize $Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)}\right)$,

$$Q\left(\boldsymbol{\theta}^{(t)}; \boldsymbol{\theta}^{(t-1)}\right) > Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)}\right).$$

3. Increase t by 1.

4. Repeat steps 1 to 3 many times until all the parameters converge. More detail about convergence is given in Section 2.3.1.

2.3.1 Convergence of EM Algorithm

Convergence is often detected by comparing the log likelihood or parameter estimates between two consecutive steps and halting when the change between two iterations is slight or tolerated. This type of convergence is called the lack of progress criteria.

One of the this type of criteria is to detect the change of log likelihood between two consecutive iterations, which is expressed as follows:

$$\left|\log \mathcal{L}\left(\boldsymbol{\theta}^{(t)}; \mathbf{X}\right) - \log \mathcal{L}\left(\boldsymbol{\theta}^{(t-1)}; \mathbf{X}\right)\right| < tol.$$

tol is an user-specified tolerance, which is usually a very small number, e.g. 10^{-6}. Or we can also detect the changed ratio of the difference between two consecutive steps log likelihood over the previous step log likelihood,

$$\frac{\left|\log \mathcal{L}\left(\boldsymbol{\theta}^{(t)}; \mathbf{X}\right) - \log \mathcal{L}\left(\boldsymbol{\theta}^{(t-1)}; \mathbf{X}\right)\right|}{\log \mathcal{L}\left(\boldsymbol{\theta}^{(t-1)}; \mathbf{X}\right)} < tol.$$

In addition, we can also compare parameter estimates between two consecutive steps,

$$\left|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t-1)}\right| < tol.$$

Another convergence criteria is Aitken's convergence criteria. Compared with the two criteria introduced above, Aitken's criterion computes an estimate of the final converged value of the log likelihood of the solution based on the previous three successive estimates iteratively computed at each new stage.

$$\left|\log \mathcal{L}\left(\boldsymbol{\theta}^{(t)}; \mathbf{X}\right)_A - \log \mathcal{L}\left(\boldsymbol{\theta}^{(t-1)}; \mathbf{X}\right)_A\right| < tol,$$

where

$$\log \mathcal{L}\left(\boldsymbol{\theta}^{(t)}; \mathbf{X}\right)_A = \log \mathcal{L}\left(\boldsymbol{\theta}^{(t-1)}; \mathbf{X}\right) + \frac{1}{1 - a^{(t-1)}}\left(\log \mathcal{L}\left(\boldsymbol{\theta}^{(t)}; \mathbf{X}\right) - \log \mathcal{L}\left(\boldsymbol{\theta}^{(t-1)}; \mathbf{X}\right)\right)$$

and

$$a^{(t-1)} = \frac{\left(\log \mathcal{L}\left(\boldsymbol{\theta}^{(t)}; \mathbf{X}\right) - \log \mathcal{L}\left(\boldsymbol{\theta}^{(t-1)}; \mathbf{X}\right)\right)}{\left(\log \mathcal{L}\left(\boldsymbol{\theta}^{(t-1)}; \mathbf{X}\right) - \log \mathcal{L}\left(\boldsymbol{\theta}^{(t-2)}; \mathbf{X}\right)\right)}.$$

When the difference of $\log \mathcal{L}(\boldsymbol{\theta}; \mathbf{X})_A$, an estimate of the converged value of the log likelihood, between two consecutive steps is small enough, then we say the algorithm has converged. Compared with the other convergence criteria, which measure how close the current iteration to the previous iteration, while Aitken's convergence measures how close the current estimate of the converged value is to the previous iteration's estimate of the converged value. So in this thesis, I will use the Aitken's criterion as the convergence criterion [79].

2.4 Markov chain Monte Carlo

Markov chain Monte Carlo (McMC) algorithms are a class of algorithms for sampling from a probability distribution by using a Markov chain [43] and it is particularly useful in Bayesian inference because it emphasizes drawing samples from conditional posterior distributions. A Markov chain is a sequence of random variables with memory-less property. In the Bayesian inference of this thesis, I will mainly focus on the first order Markov chain, which is described as the probability of moving to a future state being independent from the previous states given the current state, $P\left(\boldsymbol{\theta}^{(t+1)} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}, \cdots, \boldsymbol{\theta}^{(1)}\right) = P\left(\boldsymbol{\theta}^{(t+1)} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}\right)$ [75]. So the goal of the McMC process is to construct a Markov chain, which converges to the target distribution by drawing samples progressively from either the conditional posterior distributions or the proposal distributions given the current values. However, the full posterior distributions are usually too complex to draw samples directly from. Thus samples are usually drawn from their conditional posterior distributions given the other unknowns.

The Metropolis algorithm was first proposed by Nicholas Metropolis and Arianna W. Rosenbluth [81] and was then generalized by Wilfred K. Hastings in [51]. The Metropolis Hastings algorithm is a random walk to converge to the target distribution, which will be illustrated in detail in Section 2.4.1. Another commonly used McMC algorithm is Gibbs sampling (a special case of the Metropolis Hastings) [109], which can directly sample values from the conditional posterior distributions without needing any proposal distributions. It will be illustrated in detail in Section 2.4.2

Usually, without any extra information, the starting values may not be close to the central or the main part of the parameter distributions. If the starting values are far away from the central or the main part of the parameter distributions, it will take a longer time to converge, otherwise it will take less time. So the initial period before

parameters getting close to the central or the main part of the parameter distributions should be discarded and this period is called the "burn-in" period. The estimated value will then be calculated from the stable or converged period (period after "burn-in" period). The "burn-in" period can be visually identified from a trace plot. An example of a trace plot is shown in Figure 2.1.

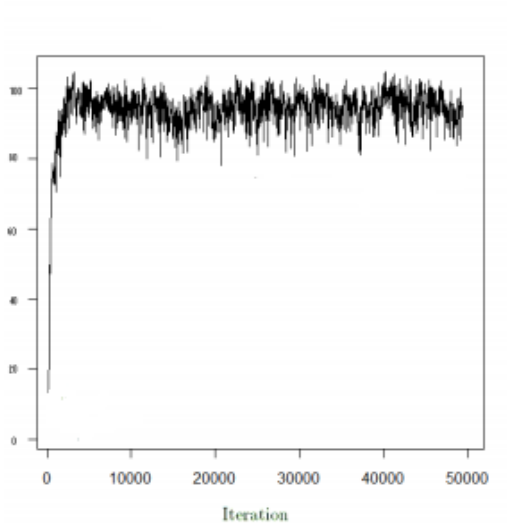


FIGURE 2.1: Trace Plot

From Figure 2.1, we can tell that the "burn-in" period is the period before the $5,000^{th}$ iteration, so this period will be discarded. The point estimate of the parameter θ is usually taken to be a central value, such as the mean or median of the converged period. If $T = 50,000$ is the total number of iterations and the length of the "burn-in" period is $B = 5,000$, then θ can be estimated by the mean of the converged period, e.g. $\hat{\theta} = \frac{\sum_{t=B+1}^{T} \theta_t^{(t)}}{T-B}$ and the 95% credible interval is $(q_{0.025,\theta}, q_{0.975,\theta})$, where $q_{0.025,\theta}$ is the 2.5% quantile of $\theta^{(t)}$ (t=5,000 to 50,000) and $q_{0.975,\theta}$ is the 97.5% quantile of $\theta^{(t)}$, where $\theta^{(t)}$ is the t^{th} sample from the McMC chain. The point and interval estimates of θ will only depend on the sample values after the "burn-in" period. "

2.4.1 McMC: Metropolis Hastings

A general technique in sampling unknown population parameters is the Metropolis Hastings method. In the Metropolis Hastings method, the newly proposed value at

the t^{th} iteration will be generated from a proposal distribution and compared it with its current value, then accepted with a certain probability.

The process of the Metropolis Hastings is as follows:

1. Choose a starting value $\boldsymbol{\theta}^{(0)}$ for the concerned parameter $\boldsymbol{\theta}$, we start with $t = 0$.
2. At each iteration t, increase t by 1.We generate a new value of $\boldsymbol{\theta}$ from the proposal distribution $Q\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t-1)}\right)$ and name it as $\boldsymbol{\theta}^*$.
3. Compute the acceptance ratio

$$r = \frac{Q\left(\boldsymbol{\theta}^{(t-1)} \mid \boldsymbol{\theta}^*\right) P\left(\boldsymbol{\theta}^* \mid \mathbf{X}\right)}{Q\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(t-1)}\right) P\left(\boldsymbol{\theta}^{(t-1)} \mid \mathbf{X}\right)}. \tag{2.4}$$

4. Accept $\boldsymbol{\theta}^*$ as $\boldsymbol{\theta}^{(t)}$ with the probability of $\min(r, 1)$, otherwise, $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$.
5. Repeat steps 2 to 4 many times.

The final point estimate of the parameter is usually the mean or median of the sampled values after the "burn-in" period, as previously discussed.

In particular, if $Q(\cdot)$ is a symmetric distribution around $\boldsymbol{\theta}^{(t-1)}$, then (2.4) will be simplified as follows:

$$r = \frac{P\left(\boldsymbol{\theta}^* \mid \mathbf{X}\right)}{P\left(\boldsymbol{\theta}^{(t-1)} \mid \mathbf{X}\right)}.$$

And this algorithm variant is called the Metropolis algorithm.

In Metropolis Hastings, the newly proposed values are all generated from a corresponding proposal distribution. How to choose an appropriate proposal distribution can be challenging.

The proposal distribution family with its initial parameters can be chosen based on some pre-run simulations or previous studies. For example, the proposal distribution of a mixing proportion can be modeled by a Dirichlet distribution. However, those proposal variance parameters can also be updated over a certain period based on the acceptance rate (how many times the newly proposed values $\boldsymbol{\theta}^*$ are accepted during this period). If the acceptance rate is too low [102] (e.g. below 0.2), the proposal distribution variance will be reduced, so we provide a smaller range for the newly proposed values; If the acceptance rate is too high (e.g. above 0.4), we will increase the proposal distribution variance as it indicates that a suitable range of values for the parameter has not been found. Otherwise, we leave the variance unchanged.

2.4.2 McMC: Gibbs Sampling

Gibbs sampling is also known as heat bath method [109]. Compared to Metropolis Hastings, the benefit of using Gibbs sampling is that all the McMC samples will be accepted (an acceptance rate of 1). In addition, it can directly sample values from the conditional posterior distributions without needing any proposal distributions. This desirable property makes it useful in sampling values from any recognized distributions [41]. So if the conditional posterior distribution is in any recognized distributional form, e.g. normal distribution, t distribution, Wishart distribution; Gibbs sampling can then be used as a technique to generate the McMC samples.

The process of Gibbs sampling will be illustrated by using the following example. Suppose we have two parameters $\boldsymbol{\theta} = (\theta_1, \theta_2)$ and their corresponding conditional posterior distributions are in some recognized distributions of form. We start from $t = 0$, the Gibbs sampling procedure at t^{th} iteration is as follows:

$$\theta_1^{(t)} \sim P\left(\theta_1 \mid \theta_2^{(t-1)}, \mathbf{X}\right),$$

$$\theta_2^{(t)} \sim P\left(\theta_2 \mid \theta_1^{(t)}, \mathbf{X}\right).$$

As $t \to \infty$, the conditional posterior distributions will converge to their full posterior distributions, $P\left(\theta_1 \mid \theta_2^{(t)}, \mathbf{X}\right) \to P\left(\theta_1 \mid \theta_2, \mathbf{X}\right)$ and $P\left(\theta_2 \mid \theta_1^{(t)}, \mathbf{X}\right) \to P\left(\theta_2 \mid \theta_1, \mathbf{X}\right)$. The final point estimates of the parameters will usually be the means or medians of the sampled values after the "burn-in" period.

2.5 Multidimensional Scaling

As mentioned in Section 1.1, there are two common types of clustering data. One is the relationship between pairs of objects, resulting in dissimilarity or similarity data. The other type is coordinate data. Although a lot of well developed clustering techniques use relationship data, coordinate data are still desirable for some types of clustering techniques, such as finite mixture models. Multidimensional scaling (MDS) is a technique to estimate coordinate data based on the between object relationships [86]. The main application of MDS is to help users to visualize the data by reducing it from a high

dimensional to a low dimensional space and then identifying the underlying structures or clusters of the data [86]. MDS will be further illustrated in Chapter 3.

2.6 Procrustes Transformation

In this section, I will discuss the Procrustean transformation, which is used to solve the problems caused by lack of uniqueness in multidimensional scaling solutions. In addition, I will define some matrix notation and review the matrix calculations related to the Procrustean transformation.

2.6.1 Matrix Notation

Suppose $X_{(n\times P)}$ denotes a matrix with n rows and P columns, then its transpose matrix is denoted as $X^T_{(P\times n)}$ and if $P = n$, then the inverse matrix is denoted as $X^{-1}_{(n\times n)}$. Their corresponding mathematical forms are shown below:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1P} \\ x_{21} & x_{22} & \cdots & x_{2P} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nP} \end{bmatrix},$$

$$\mathbf{X}^T = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1P} & x_{2P} & \cdots & x_{nP} \end{bmatrix}.$$

If $\mathbf{X}$ is non-singular, then $\mathbf{X}^{-1}$ is its inverse matrix, where $\mathbf{X}\mathbf{X}^{-1} = \mathbf{X}^{-1}\mathbf{X} = \mathbf{I}$.

2.6.2 Singular Value Decomposition

Singular value decomposition (SVD) is a technique which helps to decompose a matrix into several matrices, i.e. $\mathbf{X} = \mathbf{U}\sum\mathbf{V}^T$, which is also used in the Procrustes transformation.

$\mathbf{U}$ is an $n \times n$ orthogonal matrix, which means its transpose $\mathbf{U}^T$ is its inverse, i.e. $\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix (all the diagonal elements are 1s, the rest of entries are 0s). $\sum$ is a $n \times P$ matrix with all the diagonal values are non-negative values, e.g. $\sum_{ii} \geq 0$ for $i = 1, \cdots, \min(n, P)$, and non-diagonal entries are all 0. $\mathbf{V}^T$ (a $P \times P$ matrix) is an orthogonal matrix.

2.6.3 Details of the Procrustes Transformation

The Euclidean distances between a set of objects in Euclidean space are invariant under rotation, reflection and translation [20]. Therefore, for a given distance matrix, there could be an infinite number of positions lying in one Euclidean space to represent the same set of objects. This can be visually illustrated by using the example in Figure 2.2,

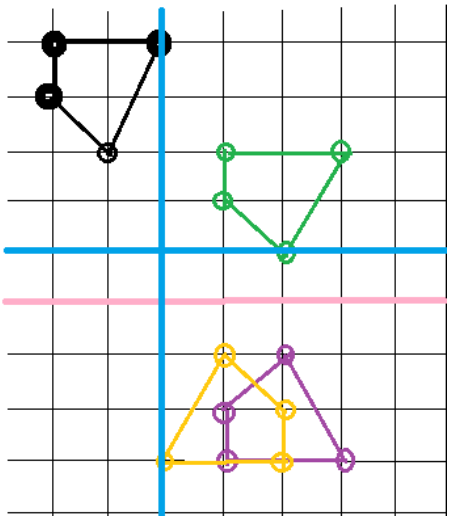


FIGURE 2.2: Transformation & Reflection & Rotation

Assuming the black configuration is the actual configurations of these four objects, if we move these data 3 steps to the right hand side and 2 steps downward, then the black configuration will be transformed into the green configuration; if we reflect the green configuration based on the pink line, we then will get the purple configuration; if we use two blue crossing lines to create an intersect point, then we rotate the black configuration based on this intersect point, we then will get the yellow configuration. As we can see from this example, all of these configurations share one common distance matrix. So it is easy to tell that for a given distance matrix, there are many possible

configurations displaying the same set of distances.

This issue of lack of uniqueness in multidimensional scaling solutions can be addressed by applying the Procrustean transformation. The procedure will be illustrated below.

Suppose the target or the actual configuration of the data is $\mathbf{X}_{ac} = (\mathbf{X}_{ac,1}, \cdots, \mathbf{X}_{ac,N})$, $\mathbf{X}_{ac,i} \in \mathcal{R}^P$ and we hope to use $\mathbf{X}_{ac}$ as a reference configuration across the study, then we need to transform all the other estimated configurations $\mathbf{X} = (\mathbf{X}_1, \cdots, \mathbf{X}_N)$ as close to $\mathbf{X}_{ac}$ as possible with the help of an orthogonal transformation matrix $\mathbf{Q}_{ac,(P\times P)}$. This means to transform $\mathbf{X}$ into $\mathbf{X}_{ac}$ by using a combination of transformations, rotations, reflections [20], i.e. $\mathbf{X}_{ac} = \mathbf{X}\mathbf{Q}_{ac}$.

$\mathbf{Q}_{ac}$ is calculated by minimizing the difference between the estimated configuration and the target configuration, this difference can be measured by the Euclidean norm (or called the Frobenius norm) [44] of them, which is expressed as follows:

$$\|\mathbf{X}\mathbf{Q}_{ac} - \mathbf{X}_{ac}\| = \sqrt{\text{trace}\left(\mathbf{X}^T\mathbf{X} + {\mathbf{X}_{ac}}^T\mathbf{X}_{ac}\right) - 2\text{trace}\left({\mathbf{X}_{ac}}^T\mathbf{X}\mathbf{Q}_{ac}\right)}. \tag{2.5}$$

As the first part of (2.5) does not have $\mathbf{Q}_{ac}$, so the minimal value of $\|\mathbf{X}\mathbf{Q}_{ac} - \mathbf{X}_{ac}\|$ should be achieved by maximizing $2\text{trace}\left({\mathbf{X}_{ac}}^T\mathbf{X}\mathbf{Q}_{ac}\right)$. The singular value decomposition of ${\mathbf{X}_{ac}}^T\mathbf{X}$ is

$$({\mathbf{X}_{ac}}^T\mathbf{X})_{(P\times P)} = \mathbf{U}_{ac,(P\times P)}\mathbf{S}_{ac,(P\times P)}{\mathbf{V}_{ac,(P\times P)}}^T,$$

then

$$\text{trace}\left({\mathbf{X}_{ac}}^T\mathbf{X}\mathbf{Q}_{ac}\right) = \text{trace}\left(\mathbf{S}_{ac}{\mathbf{V}_{ac}}^T\mathbf{Q}_{ac}\mathbf{U}_{ac}\right) = \text{trace}\left(\mathbf{S}_{ac}\mathbf{H}_{ac}\right) = \sum_{j=1}^{P} s_{ac(j)}h_{ac(jj)}.$$

As

$$\begin{aligned}\text{trace}\left(\mathbf{S}_{ac}\mathbf{H}_{ac}\right) = \text{trace}\left(\mathbf{S}_{ac}{\mathbf{V}_{ac}}^T\mathbf{Q}_{ac}\mathbf{U}_{ac}\right) &= \text{trace}\left(\mathbf{S}_{ac}{\mathbf{V}_{ac}}^T\mathbf{Q}_{ac}^{1/2}\mathbf{Q}_{ac}^{1/2}\mathbf{U}_{ac}\right)\\ &= \left\langle \mathbf{S}_{ac}{\mathbf{V}_{ac}}^T\mathbf{Q}_{ac}^{1/2}, \mathbf{Q}_{ac}^{1/2}\mathbf{U}_{ac}\right\rangle,\end{aligned}$$

by the Cauchy-Schwarz inequality [25] and the invariance of the $\|\cdot\|$ under orthogonal transformations (both ${\mathbf{V}_{ac}}^T\mathbf{Q}_{ac}^{1/2}$ and $\mathbf{Q}_{ac}^{1/2}\mathbf{U}_{ac}$ are orthogonal matrices), we get

$$\text{trace}\left(\mathbf{S}_{ac}\mathbf{H}_{ac}\right) \le \left\|\mathbf{S}_{ac}{\mathbf{V}_{ac}}^T\mathbf{Q}_{ac}^{1/2}\right\| \cdot \left\|\mathbf{Q}_{ac}^{1/2}\mathbf{U}_{ac}\right\| = \|\mathbf{S}\| \cdot \|\mathbf{I}\| = \text{trace}\left(\mathbf{S}_{ac}\mathbf{I}\right).$$

Since $s_{ac(j)}$ are non-negative numbers, so trace $(\mathbf{S}_{ac}\mathbf{H}_{ac})$ will be maximized when $h_{ac(jj)} = 1$ for $j = 1, \cdots, P$ [104],

$$\mathbf{H}_{ac} = \mathbf{I} = \mathbf{V}_{ac}{}^{T}\mathbf{Q}_{ac}\mathbf{U}_{ac},$$

in another words,

$$\mathbf{Q}_{ac} = \mathbf{V}_{ac}\mathbf{U}_{ac}{}^{T}$$

is the required transformation matrix in the Procrustean transformation. So all the other estimated configurations can be transformed as much as possible close to the target configuration by using $\mathbf{Q}_{ac}$.

Chapter 3

Multidimensional Scaling

The main target of this thesis is to cluster areal units. As mentioned in Section 1.1, there are two common types of clustering data, one type is similarity or dissimilarity data, the other is coordinate data. Some cluster techniques, such as hierarchical clustering or Chameleon hierarchical clustering (details will be given in Section 4.5.2) group the objects based on dissimilarity or similarity data. However, some other cluster techniques require coordinate data, such as model-based clustering (details will be given in Section 4.6). The two types of clustering data can be transformed from one to another by using some statistical techniques. If only dissimilarity data are available, we can estimate coordinate data by applying multidimensional scaling (MDS) techniques. On the other hand, if only coordinate data are available, we can use different types of distance functions to transform the coordinate data into dissimilarity data, e.g. Euclidean distance, which will be further discussed in Section 4.1. Similarly, dissimilarity data can also be transformed into similarity data by applying any monotonically decreasing function, and vice versa.

In this chapter, I will introduce three types of multidimensional scaling techniques. The first one is classical multidimensional scaling (CMDS), which is one of the most commonly used and well developed MDS methods, its details will be given in Section 3.1. The second MDS technique is Bayesian multidimensional scaling (BMDS) [86], which proposed an McMC algorithm to obtain a Bayesian solution for the object configuration, this will be further illustrated in Section 3.2. Split-and-combine classical multidimensional scaling [113] introduced in Section 3.3 is a technique to deal with large-scale data. In addition, how to make decisions about the number of dimensions will be covered in Sections 3.1.1 and 3.2.1.

3.1 Classical Multidimensional Scaling

The most commonly used MDS technique is classical multidimensional scaling (CMDS) which was proposed by Warren S. Torgerson [112]. Suppose the object configuration in a P dimensional Euclidean space is expressed as $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N\}$ and its i^{th} object's configuration is $\mathbf{X}_i = \{X_{11}, X_{12}, \cdots, X_{1P}\}$, each dimension representing an attribute of the object; d_{ij} denotes the observed distance between objects i and j, then the goal of CMDS is to estimate $\mathbf{X}$ from its observed distance matrix $\mathbf{D}(d_{ij})$.

The procedure of CMDS is as follows [112]:

1. Calculate a matrix $\mathbf{B}_{(N\times N)} = -\frac{1}{2}\mathbf{J}_{(N\times N)}\mathbf{D}_{(N\times N)}\mathbf{J}_{(N\times N)}$, where $\mathbf{J} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$. Specifically, $\mathbf{I}$ is a $N \times N$ identity matrix whose diagonal elements are all 1 and off-diagonal are all 0; $\mathbf{1}_{(N\times 1)}$ is column vector of N 1's.
2. For a given dimensionality P, extract the largest eigenvalues $\mathbf{\Lambda}_P = \{\lambda_1, \cdots, \lambda_P\}$ and their corresponding eigenvectors $\mathbf{E}_P = \{e_1, \cdots, e_P\}$ from $\mathbf{B}$.
3. $\mathbf{X}$ can be calculated from $\mathbf{X} = \mathbf{E}_P\mathbf{\Lambda}_P^{1/2}$.

3.1.1 Choice of Dimension: Scree Plot

One challenge in applying MDS is decisions about the number of dimensions (attributes) P. The higher the dimensionality is, the smaller the difference will be between the distance in the constructed points, $\|\mathbf{X}_i - \mathbf{X}_j\|$, and the original distances, d_{ij}. However, sometimes one of the uses of MDS is to reduce the dimensionality of the data [17], e.g. so that users can visually detect data patterns. So if the number of dimensions is too large, then MDS will fail to accomplish this aim and the interpretations of the P dimensions will be problematic. Also, for any overly large P, MDS could be modeling noise in the data as well as signal.

One of the most commonly used techniques in determining the number of dimensions is the scree plot with dimensionality plotted against its stress. Stress is defined as follows:

$$\text{STRESS} = \sqrt{\frac{\sum_{i>j}\left(\delta_{ij} - d_{ij}\right)^2}{\sum_{i>j}\delta_{ij}^2}},$$

where δ_{ij} denotes the calculated distance between objects i and j,

$$\delta_{ij} = \sqrt{\sum_{k=1}^{P} (X_{ik} - X_{jk})^2},$$

where X_{ik} is the k^{th} attribute value of object i. If the line levels off after the dimensionality hits P_o, then P_o will be the number of dimensions for the data. An example of such a scree plot is shown in Figure 3.1.

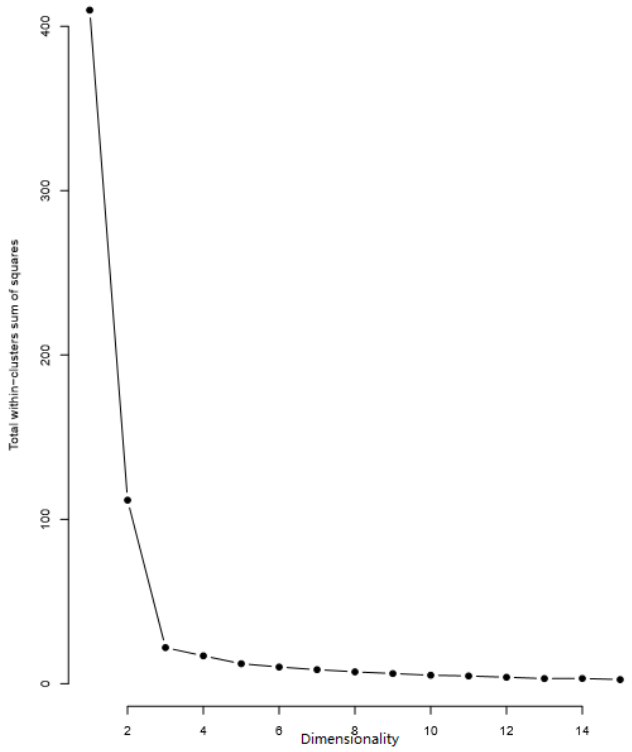


FIGURE 3.1: CMDS Scree Plot

In Figure 3.1, stress keeps dropping as the number of dimensions increasing, but the stress only decreased sharply when the number of dimensions changed from 1 to 3, so it seems reasonable to use a three-dimensional Euclidean space to display the data configuration. So the selected number of dimensions from the available range of dimensions is the one with a relatively small stress value but low dimensionality.

3.2 Bayesian Multidimensional Scaling with Dissimilarities

Bayesian multidimensional scaling with dissimilarities (BMDS) was proposed by Man-Suk Oh and Adrian E. Raftery [86]. It is a technique using Bayesian inference to estimate the data configuration along with determining the data dimensions.

Unlike CMDS which does not use any models, BMDS uses a Euclidean distance model and assumes Gaussian measurement error in the observed dissimilarities $\mathbf{D}(d_{ij})$. It then uses an McMC algorithm to seek a Bayesian solution for the model [86]. It assumes that the observed distances are generated from a truncated normal distribution with means equal to the distances between points,

$$d_{ij} \sim N\left(\delta_{ij}, \sigma^2\right) \mathrm{I}\left(d_{ij} > 0\right) i \neq j, i, j = 1, \cdots, N, \tag{3.1}$$

where $\delta_{ij} = \sqrt{\sum_{k=1}^{P} \left(X_{ik} - X_{jk}\right)^2}$ denotes the distance between objects i and j, P is the number of dimensions in the Euclidean space, X_{ik} and X_{jk} denote the k^{th} attribute of objects i and j on the k^{th} dimension.

The reason for using a truncated normal distribution is due to the need for all distances to have non-negative values. The application of a truncated distribution can successfully guarantee this property. For $a \leq X \leq b$, the truncated normal probability density of X with mean μ and variance σ^2 is expressed as follows:

$$f\left(X; \mu, \sigma, a, b\right) = \frac{\frac{1}{\sigma}\phi\left(\frac{X-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)},$$

where $\Phi\left(\cdot\right)$ is the cumulative distribution function (CDF) of the standard normal distribution and $\phi\left(\cdot\right)$ is the probability density function (PDF) of the standard normal distribution. In the case of distances, we should set $a = 0$ and $b = +\infty$ in order to limit all distances to non-negative values. So the distance's truncated normal distribution is defined as follows:

$$f\left(X; \mu, \sigma, a, b\right) = \frac{\frac{1}{\sigma}\phi\left(\frac{X-\mu}{\sigma}\right)}{1 - \Phi\left(\frac{-\mu}{\sigma}\right)}.$$

So the likelihood function of the unknown parameters $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N\}$ and σ^2

can then be expressed as follows:

$$\mathcal{L}\left(\mathbf{X}, \sigma^2 \mid \mathbf{D}\right) \propto \left(\sigma^2\right)^{-m/2} \exp\left[-\frac{1}{2\sigma^2}\mathrm{SSR} - \sum_{i>j} \log \Phi\left(\frac{\delta_{ij}}{\sigma}\right)\right], \tag{3.2}$$

where $m = N\left(N-1\right)/2$ is the number of pairs of dissimilarities, $\mathrm{SSR} = \sum_{i>j}\left(d_{ij} - \delta_{ij}\right)^2$ denotes the sum of squared residuals.

In BMDS, the prior distributions of the unknown parameters, $\mathbf{X}$, σ^2 are assumed as follows:

$$\mathbf{X}_i \sim \mathrm{N}\left(\mathbf{0}, \mathbf{\Lambda}\right), \text{ for } i = 1, \cdots, N.$$

The prior distribution of $\mathbf{X}_i$ is a multivariate normal distribution with a diagonal covariance matrix $\mathbf{\Lambda} = \mathrm{Diag}\left(\lambda_1, \cdots, \lambda_p\right)$ according to the published paper Oh and Raftery [87]. We assume the dimensions are independent, which is similar to the assumption in the principal component analysis (PCA) [92], for which components are independent. If we incorrectly assume the dimensions are independent, then we are more likely to identify more dimensions than the actual dimensionality. More simulations about the dependent dimensions will also be conducted in Chapter 9 simulation section, they will show if the assumption is not true, how the simulation results will be changed. We further assume that σ^2 follows an inverse gamma distribution,

$$\sigma^2 \sim \mathrm{IG}\left(a, b\right).$$

The inverse gamma distribution has an advantage over other prior distributions for σ^2 as it can guarantee all variances have non-negative values and is also the conjugate to the likelihood function. For this reason, a prior distribution $\mathrm{IG}\left(\alpha, \beta_j\right)$ is chosen for λ_j. As different dimensions will have different variances, so we set different scale parameter β_j for different λ_j. In addition, Oh and Raftery [87] suggested setting the shape parameter α to a small constant (e.g. $1/2$), so the prior information roughly corresponds to the unit informative prior which is the data dependent prior and is consistent with the setting of the other tuning parameters.

As mentioned in Section 2.1.2.1, parameters α, $\boldsymbol{\beta}(\beta_j)$, a and b in the prior distributions are called hyperparameters. Empirical Bayes methods [24] are procedures for statistical inference in which the prior distribution is estimated from the data. This approach stands in contrast to standard Bayesian methods, for which the prior distribution is fixed before any data are observed. The benefits of empirical Bayes includes requiring less time and data, which reduces the study cost [24]. However, the disadvantage of empirical Bayes can be contradicting the definition of the prior distribution that should

be based on information before seeing any data. α and a are shape parameters, $\boldsymbol{\beta}(\beta_j)$ and b are scale parameters. The shapes of the densities of an inverse gamma distribution with different shape and scale parameters are shown in Figure 3.2. From Figure 3.2 we can see that it is reasonable to set shape parameters to be smaller values so that the inverse gamma distribution can cover a wider range of values.

The scale parameter b is chosen to make the prior mean of σ^2 equal to $\text{SSR}^{(0)}/m$ according to the published paper of Oh and Raftery [87], where SSR is the initial sum of squared residuals estimated from a preliminary run CMDS result. Similarly, we can set β_j to be the j^{th} diagonal element of $S_x = \frac{1}{N}\sum_{i=1}^{N} \mathbf{X}_i{}'\mathbf{X}_i$, where S_x is the covariance matrix of $\mathbf{X}$ (from an initial CMDS run). However, there are some limitations in using the preliminary run results to initialize the parameters. Bayesian inference refers to the general idea of placing a prior distribution or an initial belief on the parameters, then updating the parameters by using the newly observed values. As with the empirical inference of the parameters proposed in Oh and Raftery [87] we will use the same data twice as in empirical Bayes, both in estimating the initial values and updating the parameters. The alternative way to deal with this limitation can be weakly informative priors which expresses more about the researcher's attitude towards the parameters.

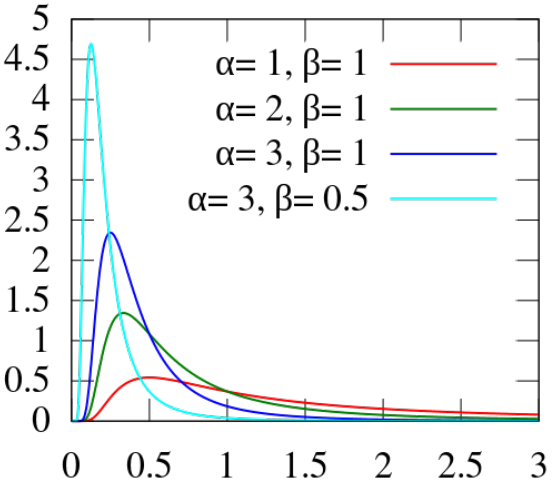


FIGURE 3.2: Inverse Gamma Distributions with Different Parameters
Source: https://en.wikipedia.org/wiki/Inverse-gamma_distribution

The full posterior distribution of all unknown parameters $\left(\mathbf{X}, \sigma^2, \boldsymbol{\Lambda}\right)$ can be found by multiplying all prior distributions with the likelihood function (3.2). The full posterior

distribution of all the unknown parameters can be expressed as follows:

$$f\left(\mathbf{X}, \sigma^2, \mathbf{\Lambda} \mid \mathbf{D}\right) \propto \left(\sigma^2\right)^{-(m/2+a+1)} \prod_{k=1}^{P} \lambda_j^{-N/2-\alpha-1} \exp\left\{-\frac{1}{2}\sum_{i=1}^{N} \mathbf{X}_i{}' \mathbf{\Lambda}^{-1} \mathbf{X}_i - \frac{b}{\sigma^2} - \frac{\mathrm{SSR}}{2\sigma^2} - \sum_{i>j} \log \Phi\left(\frac{\delta_{ij}}{\sigma}\right) - \sum_{k=1}^{P} \frac{\beta_k}{\lambda_k}\right\}.$$

As generating new values from the full posterior distribution directly would be difficult, instead we can construct Markov chains separately based on each parameter and then generate the new values from their conditional posterior distributions given the current step's values. Take $\mathbf{X}_i$ for example, at the t^{th} $(t > 0)$ iteration, the new values $\mathbf{X}_i^{(t)}$ are generated from its conditional posterior distribution,

$$f\left(\mathbf{X}_i \mid \mathbf{X}_i^{(t-1)}, \sigma^{2,(t-1)}, \mathbf{\Lambda}^{(t-1)}, \mathbf{D}\right),$$

so the future state's values only depend on the current state's values.

The initial values of unknown parameters $\mathbf{X}_i^{(0)}$ will be estimated from a preliminary run simulation by using CMDS [86]. The initial $(\sigma^2)^{(0)}$ can then be $\mathrm{SSR}^{(0)}/m$ and $\Lambda^{(0)}$ be the sample variance of the j^{th} coordinates of $\mathbf{X}^{(0)}$ [86].

The conditional posterior distribution of λ_j is

$$f(\lambda_j \mid \cdots) \propto (\lambda_j)^{-\alpha-N/2-1} \exp\left(-\frac{\beta_j + s_j/2}{\lambda_j}\right),$$

which is also a recognized distribution, an inverse Gamma distribution $\mathrm{IG}(\alpha + N/2, \beta_j + s_j/2)$, where s_j/N is the sample variance of j^{th} dimension for $\mathbf{X}$. So Gibbs sampling will be used to generate new values of $\mathbf{\Lambda}$ from its conditional posterior distribution directly.

The conditional posterior distribution of $\mathbf{X}_i$ is expressed as follows:

$$f\left(\mathbf{X}_i \mid \cdots\right) \propto \exp\left[-\frac{1}{2}\left(\frac{\sum_{j=1, j\neq i}\left(\delta_{ij} - d_{ij}\right)^2 / \sigma^2}{\sigma^2} + \mathbf{X}_i{}' \mathbf{\Lambda}^{-1} \mathbf{X}_i\right) - \sum_{i>j} \log \Phi\left(\frac{\delta_{ij}}{\sigma}\right)\right]. \tag{3.3}$$

This conditional posterior distribution is not in the form of any recognized distributions, so Metropolis Hastings will be used to simulate $\mathbf{X}_i$.

In the Bayesian multidimensional scaling method, Oh and Raftery [86] used a multivariate normal distribution as the proposal distribution of $\mathbf{X}$ with the mean equals to values estimated from the current step. From Equation (3.3), it is easy to tell that the

conditional posterior distribution of $\mathbf{X}_i$ is mainly dominated by the term with the highest power, which is the quadratic term in this case. In $\sum_{j=1,j\neq i}\left(\delta_{ij}-d_{ij}\right)^2/\sigma^2$, there are $N-1$ quadratic terms of $\mathbf{X}_i$ with a leading coefficient $1/\sigma^2$. The other quadratic term is $\mathbf{X}_i{}'\mathbf{\Lambda}^{-1}\mathbf{X}_i$ with a coefficient $\mathbf{\Lambda}^{-1}$, so the dominated term will be decided by $\sum_{j=1,j\neq i}\left(\delta_{ij}-d_{ij}\right)^2/\sigma^2$ as it has more quadratic terms, then we can set the variance of the proposal distribution of $\mathbf{X}_i$ proportional to $\sigma^2/(N-1)$ [86]. However, the variances of this proposal can also be updated over a certain period based on the acceptance rate (how many times the newly proposed $\mathbf{X}_i$'s are accepted during this period), details of this have been discussed in Section 2.4.1.

The conditional posterior distribution of σ^2 is expressed as follows:

$$f\left(\sigma^2 \mid \cdots\right) \propto \left(\sigma^2\right)^{-(m/2+a+1)} \exp\left[-\frac{1}{\sigma^2}\left(\mathrm{SSR}/2+b\right)-\sum_{i>j}\log\Phi\left(\frac{\delta_{ij}}{\sigma}\right)\right].$$

The conditional posterior distribution of σ^2 is not in the form of any recognized distributions either, so Metropolis Hastings will be used to to simulate σ^2.

The proposal distribution of σ^2 is approximated by a truncated normal distribution with a variance proportional to the variance of $\mathrm{IG}\left(m/2+a, \mathrm{SSR}/2+b\right)$, which is the conjugate prior distribution of σ^2 when the likelihood function is an univariate normal distribution.

3.2.1 Choice of Dimension: A Bayesian Approach

The method introduced in Section 3.2 uses Bayesian inference to estimate the object configuration for a given dimensional Euclidean space. In order to select an optimal number of dimensions for the model, Oh and Raftery [86] proposed a Bayesian dimension selection technique to do this. So in this section, I will discuss how to make decisions about the number of dimensions with a Bayesian approach.

A Bayes factor is an index to quantify the evidence between two potential hypotheses [45]. It is the ratio of the posterior distributions of the two potential hypotheses. For K

dimensional data, if the full posterior distribution is expressed as:

$$\begin{aligned} f\left(\mathbf{X}, \sigma^2, \mathbf{\Lambda}, K \mid \mathbf{D}\right) &\propto \mathcal{L}(\mathbf{X}, \sigma^2, K \mid \mathbf{D}) f(\mathbf{X} \mid \mathbf{\Lambda}, K) f(\sigma^2) f(\mathbf{\Lambda} \mid K) \\ &\propto \sigma^{-m} \exp\left[-\frac{1}{2\sigma^2} SSR - \sum_{i>j} \log \Phi(\delta_{ij}/\sigma)\right] \\ &\times (2\pi)^{-NK/2} \prod_{k=1}^{K} \lambda_k^{-N/2} \exp\left[-\sum_{k=1}^{K} \frac{1}{2\lambda_k} s_k\right] \\ &\times \Gamma(a)^{-1} b^a (\sigma^2)^{-(a+1)} \exp\left[-\frac{b}{\sigma^2}\right] \\ &\times \Gamma(\alpha)^{-K} \prod_{k=1}^{K} \beta_k^{\alpha} \lambda_k^{-(\alpha+1)} \exp\left[-\frac{\beta_k}{\lambda_k}\right] \\ &\propto A(K) \cdot h(\sigma^2, \mathbf{X}) \cdot g(\mathbf{\Lambda}, \mathbf{X}, K), \end{aligned}$$

where

$$\begin{aligned} A(K) &= (2\pi)^{(NK)/2} \Gamma(\alpha)^{-K} \prod_{k=1}^{K} \beta_k^{\alpha}, \\ h(\sigma^2, \mathbf{X}) &= (\sigma^2)^{-(m/2+a+1)} \exp\left[-\frac{SSR/2 + b}{\sigma^2} - \sum_{i>j} \log \Phi\left(\frac{\delta_{ij}}{\sigma}\right)\right], \\ g(\mathbf{\Lambda}, \mathbf{X}, K) &= \prod_{k=1}^{K} \lambda_k^{-(N/2+\alpha+1)} \exp\left[-\frac{s_k/2 + \beta_k}{\lambda_k}\right] \end{aligned}$$

then conditional posterior distribution $f\left(\mathbf{X}, K \mid \mathbf{D}\right)$ of $(\mathbf{X}, K)$ can be expressed as

$$f\left(\mathbf{X}, K \mid \mathbf{D}\right) = A(K) \int h(\sigma^2, \mathbf{X}) d\sigma^2 \int g(\mathbf{\Lambda}, \mathbf{X}, K) d\mathbf{\Lambda},$$

where $\int g(\mathbf{\Lambda}, \mathbf{X}, K) d\mathbf{\Lambda}$ can be estimated by taking advantage of the inverse gamma distribution, then $\int g(\mathbf{\Lambda}, \mathbf{X}, K) d\mathbf{\Lambda} = \left(\Gamma(N/2+\alpha)\right)^K \prod_{k=1}^{K} (s_k/2+\beta_k)^{-(N/2+\alpha)}$. $\int h(\sigma^2, \mathbf{X}) d\sigma^2 \approx (2\pi)^{1/2} (1/m)^{1/2} (SSR/m)^{-m/2+1} \exp(-m/2)$, s_k/N is the sample variance of k^{th} dimension for $\mathbf{X}$. Further details about the integration of $h(\sigma^2, \mathbf{X})$ are shown in Appendix A.1.

$$\begin{aligned} f\left(\mathbf{X}, K \mid \mathbf{D}\right) &= A(K) \int h(\sigma^2, \mathbf{X}) d\sigma^2 \int g(\mathbf{\Lambda}, \mathbf{X}, K) d\mathbf{\Lambda} \\ &\propto (2\pi)^{-NK/2} \left(\Gamma(\alpha)\right)^{-K} \prod_{k=1}^{K} \beta_k^{\alpha} \\ &\times \left(\Gamma(N/2+\alpha)\right)^K (SSR/m)^{-m/2+1} \prod_{k=1}^{K} (s_k/2 + \beta_k)^{-(N/2+\alpha)}. \end{aligned}$$

So the ratio of the conditional posterior distributions between two potential dimensions (K and $K+1$) (the Bayes factor) can be expressed as:

$$\begin{aligned} R_K =& \frac{f\left(\mathbf{X}^{(K+1)}, K+1 \mid \mathbf{D}\right)}{f\left(\mathbf{X}^{*(K+1)}, K+1 \mid \mathbf{D}\right)} \\ =& \left(\frac{SSR^{(K+1)}}{SSR^{*(K+1)}}\right)^{-m/2+1} \prod_{k=1}^{K+1} \left(\frac{s_k^{(K+1)}/2+\beta_k}{s_k^{*(K+1)}/2+\beta_k}\right)^{-(N/2+\alpha)} \\ =& \left(\frac{SSR^{(K+1)}}{SSR^{(K)}}\right)^{-m/2+1} \prod_{k=1}^{K} \left(\frac{s_k^{(K+1)}/2+\beta_k}{s_k^{(K)}/2+\beta_k}\right)^{-(N/2+\alpha)} \times \left(\frac{s_{K+1}^{(K+1)}/2+\beta_{K+1}}{\beta_{K+1}}\right)^{-(N/2+\alpha)}, \end{aligned}$$

where $\mathbf{X}^{(K)}$ indicates the data configuration on the K dimensional space and $\mathbf{X}^{*(K+1)} = (\mathbf{X}^{(K)} : \mathbf{0})$. This is because $f\left(\mathbf{X}, \sigma^2, \mathbf{\Lambda}, K \mid \mathbf{D}\right)$ depends on the scale of $\mathbf{X}$. Given the same Euclidean distance, the higher the dimension is, the closer the coordinates of $\mathbf{X}$ to the origin will be, so the smaller the variance in each coordinate can be [86]. For example, the Euclidean distance between -1 and 1 in one-dimensional space is equal to the Euclidean distance between $(1/\sqrt{2}, 1/\sqrt{2})$ and $(-1/\sqrt{2}, -1/\sqrt{2})$ in two-dimensional space, but the variance in each coordinate in two dimensional space is smaller. So in order to overcome this issue, we project all objects on the same dimensional space, but the extra coordinate in the K^{th} dimension are set to $\mathbf{0}$.

When there is no strong prior information, Oh and Raftery [86] suggested to use $\alpha = 1/2$ and $\beta_k = \frac{1}{2} s_k^{(K+1)}$, so that the prior information roughly corresponds to the unit information prior as the simple and convenient BIC approximation corresponds most closely to the unit information prior [97]. The logarithm of R_K (roughly equals to the BIC approximation) can then be expressed as:

$$\mathrm{LR}_K = (m-2)\log\left(\mathrm{SSR}_{K+1}/\mathrm{SSR}_K\right) + \left\{(N+1)\sum_{k=1}^{K}\log\left[\frac{r_k^{(K+1)}(N+1)}{\left(N+r_k^{(K+1)}\right)}\right] + (N+1)\log(N+1)\right\},$$

where $r_k^{(K+1)} = h_k^{(K+1)}/h_k^{(K)}$ and $h_k = \sum_{i=1}^{N} X_{ik}^2$, $\mathrm{SSR}_K = \sum_{i>j}(d_{ij} - \delta_{ij})^2$ is the sum of squared residuals in K dimensional space. A positive LR_K indicates K is preferable over $K+1$, otherwise, $K+1$ is preferable over K. Alternatively, we can define MDSIC (Multidimensional Scaling Information Criterion) as

$$\mathrm{MDSIC}_1 = (m-2)\log \mathrm{SSR}_1$$

$$\mathrm{MDSIC}_P = \mathrm{MDSIC}_1 + \sum_{p=1}^{P-1} \mathrm{LR}_p \text{ for any } P > 1,$$

then the selected dimension will be the one giving the minimal MDSIC value. An example of MDSIC is shown in Figure 3.3,

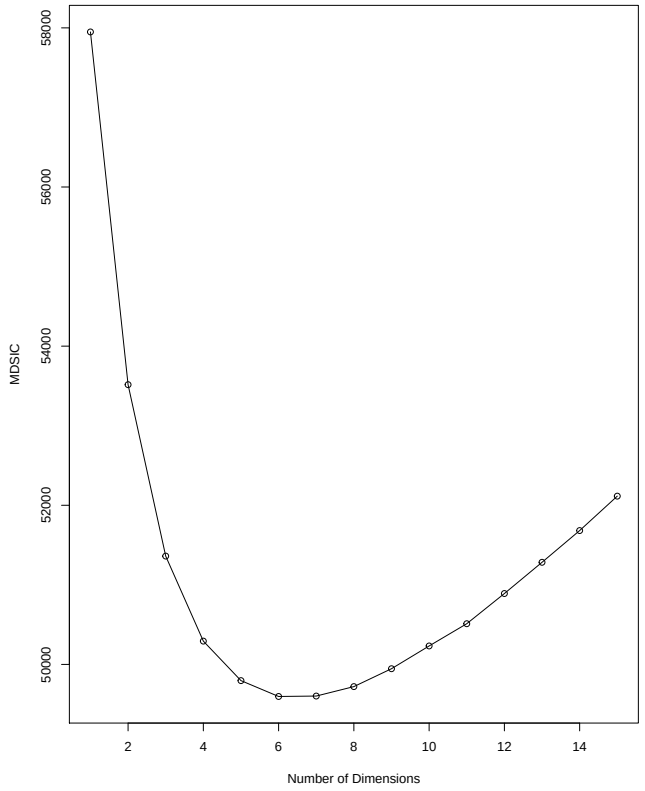


FIGURE 3.3: MDSIC

The optimal number of dimensions in Figure 3.3 is $P = 6$, where it reached the minimal value of MDSIC.

3.3 Split-and-Combine Classical Multidimensional Scaling

Both CMDS and BMDS are good at dealing with small or medium sized data. However, if the data size is too large, then it will take a longer time to get the coordinate data. In order to deal with the large-sized data issue, Jengnan Tzen [113] proposed a split-and-combine classical multidimensional scaling (SC-CMDS) method to overcome this difficulty.

The SC-CMDS technique is conducted by splitting a larger dissimilarity matrix **D**

into two smaller equal-sized dissimilarity matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ with certain overlapping objects. We denote $\mathbf{X}_1$ and $\mathbf{X}_2$ as coordinates of the overlapping objects from applying CMDS on $\mathbf{D}_1$ and $\mathbf{D}_2$ separately. Although both $\mathbf{X}_1$ and $\mathbf{X}_2$ are configurations for the same overlapping distance matrix, they may still be different due to lack of uniqueness. This issue can be addressed by setting a reference configuration, such as the configuration of the overlapping objects from $\mathbf{D}_1$, then projecting all the other configurations to this reference configuration by an affine mapping, $g(\cdot) + g_0$, where $g(\cdot)$ denotes the unitary operator (i.e. rotation and reflection), g_0 denotes the shifting operator g_0 (i.e. translation) [113]. This affine mapping will be estimated by mapping the overlapping objects from $\mathbf{X}_2$ to $\mathbf{X}_1$.

In order to find out the unitary operator $g(\cdot)$ and the shifting operator g_0, we need a tool to decompose $\left(\mathbf{X}_1 - \bar{\mathbf{X}}_1\right)^T$ and $\left(\mathbf{X}_2 - \bar{\mathbf{X}}_2\right)^T$. QR factorization is a factorization technique used to decompose any matrices into two parts, e.g. $\mathbf{A} = \mathbf{QR}$, where $\mathbf{Q}$ denotes an orthogonal matrix ($\mathbf{QQ}^T = \mathbf{I}$, $\mathbf{I}$ denotes an identity matrix), $\mathbf{R}$ denotes an upper triangular matrix with lower triangular entries are all 0, i.e. $r_{ij} = 0$, $i > j$. With the help of QR factorization, we can decompose $\left(\mathbf{X}_1 - \bar{\mathbf{X}}_1\right)^T$ and $\left(\mathbf{X}_2 - \bar{\mathbf{X}}_2\right)^T$ into,

$$\left(\mathbf{X}_1 - \bar{\mathbf{X}}_1\right)^T_{(P\times N)} = \mathbf{Q}_{1(P\times N)}\mathbf{R}_{1(N\times N)}$$

and

$$\left(\mathbf{X}_2 - \bar{\mathbf{X}}_2\right)^T_{(P\times N)} = \mathbf{Q}_{2(P\times N)}\mathbf{R}_{2(N\times N)},$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are orthogonal matrices, $\mathbf{R}_1$ and $\mathbf{R}_2$ are upper triangular matrices, $\bar{\mathbf{X}}_1$ and $\bar{\mathbf{X}}_2$ are the mean vectors of the overlapping configurations $\mathbf{X}_1$ and $\mathbf{X}_2$ separately. Since $\mathbf{X}_1$ and $\mathbf{X}_2$ are configurations for the same distance matrix, so $\mathbf{R}_1$ and $\mathbf{R}_2$ should be the same,

$$\mathbf{Q}_1^T\left(\mathbf{X}_1 - \bar{\mathbf{X}}_1\right)^T = \mathbf{Q}_2^T\left(\mathbf{X}_2 - \bar{\mathbf{X}}_2\right)^T,$$

which then results in

$$\mathbf{X}_1 = \mathbf{X}_2\mathbf{Q}_2\mathbf{Q}_1^T - \bar{\mathbf{X}}_2\mathbf{Q}_2\mathbf{Q}_1^T + \bar{\mathbf{X}}_1.$$

So the unitary operator is $g = \mathbf{Q}_2\mathbf{Q}_1^T$ and the shifting operator is $g_0 = -\bar{\mathbf{X}}_2\mathbf{Q}_2\mathbf{Q}_1^T + \bar{\mathbf{X}}_1$. More generally, for even larger data, we can split the dissimilarity data into more than two smaller dissimilarity data matrices as long as all of them use the same sub-dissimilarity data as the reference data.

Chapter 4

Cluster Analysis

Cluster analysis is a technique to group objects into clusters, where objects from the same cluster share more similar characteristics or are closer to each other than objects assigned to different clusters. There are applications of cluster analysis in many different fields. For example, in website searching, it helps web users to find out the most related links; in the housing market, it helps customers to identify groups of substitutable areas, etc.

Clustering techniques can be mainly divided into two categories, model-free clustering techniques, e.g. hierarchical clustering and Chameleon hierarchical clustering, and model-based clustering techniques, e.g. model-based clustering. The categorizations of different types of clustering techniques also depend on the types of data used. For example, hierarchical clustering mainly uses dissimilarity data, while Chameleon hierarchical clustering uses similarity data and finite mixture clustering uses coordinate data. Although the aim of clustering is to get a stable and objective clustering, there may be no unique objective 'true' or 'best' classification in the data set. The most suitable clustering depends on the purpose and the context of the study [53].

In this chapter, Section 4.2 will introduce some common types of distance measurements. In Section 4.3, I will define some graph terminology, which is required for Chameleon hierarchical clustering and this method will be explored in Section 4.5. Hierarchical clustering will be described in Section 4.4. In Section 4.6, I will discuss a model-based clustering technique and its estimation using both EM and Bayesian approaches. In Sections 4.8 to 4.10, I will explore the criteria used in making number of clusters decisions, model comparisons and clustering comparisons. The last part of this chapter

will introduce a Bayesian clustering related issue, the label switching phenomenon, and how to deal with this issue.

4.1 Common Clustering Data Types

As mentioned in Section 1.1, there are mainly two types of clustering data. One of them is coordinate data. This type of data is usually displayed in an $N \times P$ matrix,

$$\mathbf{X} = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1P} \\ X_{21} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ X_{N1} & \cdots & \cdots & X_{NP} \end{pmatrix},$$

where N is the number of objects and P is the number of dimensions (attributes), or it can also be expressed in an array, $\mathbf{X} = (\mathbf{X}_1, \cdots, \mathbf{X}_N)$, where $\mathbf{X}_i = (X_{i1}, \cdots, X_{iP})$ for all objects $i = 1, \cdots, N$ and X_{ip} is the i^{th} object's value along the p^{th} dimension or the value of its p^{th} variable. $\mathbf{X}_i$ might contain different types of variables, such as continuous variables, nominal variables or ordinal variables.

The other type of data is relationship data, resulting in dissimilarity or similarity data., which are usually expressed in an $N \times N$ symmetric matrix. For a dataset with N objects, its dissimilarity or similarity data can be expressed in the form of

$$\mathbf{S} = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1N} \\ s_{21} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ s_{N1} & \cdots & \cdots & s_{NN} \end{pmatrix}.$$

The entries are the similarities or dissimilarities between pairs of objects. One of the commonly used similarity data measures is Pearson's correlation [40], for which all the diagonal elements are 1 and non-diagonal elements are the similarities between pairs of objects. In particular, all entries s_{ij} have to be some values between -1 and 1, if $s_{ij} = -1$, then the pair of objects are perfectly negatively correlated; if $s_{ij} = 0$, then the pair of objects are uncorrelated; if $s_{ij} = 1$, then the pair of objects are perfectly positively correlated. However, a correlation matrix can only be used to explain linear relationships between pairs of continuous observations, it fails to explain non-linear relationships or relationships between categorical values. For more complicated scenarios, more advanced

measurements need to be explored, such as Cross Price Elasticity of Price (CPEP) which has been introduced in Section 1.3.1, etc.

4.2 Common Distance Measures

In Chapter 3, I introduced some MDS techniques used in obtaining coordinate from dissimilarity data. However, if only coordinate data are available, we may wish to obtain dissimilarity data from coordinate data. So in this section, I will introduce some commonly used distance measurements for transforming coordinate data, either continuous or categorical variables, into dissimilarity data.

4.2.1 Continuous Variables Distance Measures

The most commonly used distance measurement is Euclidean distance [31], which is defined as follows:

$$d_E\left(\mathbf{X}_i, \mathbf{X}_j\right) = \sqrt{\sum_{p=1}^{P}\left(X_{ip} - X_{jp}\right)^2}.$$

In this thesis, object distances will be calculated using Euclidean distance as there is no particular reason to suggest using an alternative.

However, we may wish different variables to have different influences on distances. In order to show different impacts coming from different variables, weighted Euclidean distance [31] was proposed, which is defined as follows:

$$d_{w,E}\left(\mathbf{X}_i, \mathbf{X}_j\right) = \sqrt{\sum_{p=1}^{P} w_p\left(X_{ip} - X_{jp}\right)^2},$$

where $\mathbf{w} = (w_1, \cdots, w_P)$, w_p is the weight on the p^{th} dimension.

The next distance is Mahalanobis distance [77], which is defined as follows:

$$d_{S_x}\left(\mathbf{X}_i, \mathbf{X}_j\right) = \sqrt{\left(\mathbf{X}_i - \mathbf{X}_j\right)' \mathbf{S}_x^{-1}\left(\mathbf{X}_i - \mathbf{X}_j\right)},$$

where $\mathbf{S}_x$ is a covariance matrix, which indicates whether the data is largely spread out or condensed or correlated. In particular, if $\mathbf{S}_x$ is an identity matrix, then Mahalanobis distance will be equal to Euclidean distance.

Another commonly used distance is Manhattan distance [31]. As its name suggests, the intuition for Manhattan distance came from the street layout in Manhattan. It uses objects' projections along different dimensions, which are similar to blocks in Manhattan, to define the distances. Its expression is shown as follows:

$$d_M(\mathbf{X}_i, \mathbf{X}_j) = \sum_{p=1}^{P} |X_{ip} - X_{jp}|.$$

We can also use maximum distance [31] to define the distance between pairs of objects. It is expressed as follows:

$$d_{\max}(\mathbf{X}_i, \mathbf{X}_j) = \max_{1 \le p \le P} |X_{ip} - X_{jp}|.$$

In this type of distance, distances between pairs of objects are only determined by the maximum distance over all dimensions.

Another type of distance is cosine distance [31], which takes account of the angle between objects. It is defined as follows:

$$\cos(\mathbf{X}_i, \mathbf{X}_j) = \frac{\mathbf{X}_i \cdot \mathbf{X}_j}{\|\mathbf{X}_i\| \|\mathbf{X}_j\|} = \frac{\sum_{p=1}^{P} X_{ip} X_{jp}}{\sqrt{\sum_{p=1}^{P} X_{ip}^2} \cdot \sqrt{\sum_{p=1}^{P} X_{jp}^2}},$$

where $\|\cdot\|$ is the Euclidean norm. 1 indicates two objects' directions are completely the same or equal, -1 means the two objects' directions are completely different or opposite to each other. This type of distance emphasizes data patterns more than size.

4.2.2 Categorical Variables Distance Measures

All the distances discussed above are distance measurements for continuous data, they cannot be applied to measure distances between binary data. Assume we have two objects, each with some binary traits with 1 indicating the presence of this trait and 0 its absence, then the proportion of matching pairs between two objects can be used to define the distances between the two. Its definition can be further illustrated by using Table 4.1.

TABLE 4.1: Example Contingency Table of Frequencies

		Object $\mathbf{X}_j$	
		0	1
Object $\mathbf{X}_i$	0	a	b
	1	c	d

In Table 4.1, a indicates the total number of traits neither $\mathbf{X}_i$ nor $\mathbf{X}_j$ has; b indicates the total number of traits only $\mathbf{X}_j$ has; c indicates the total number of traits only $\mathbf{X}_i$ has; d indicates the total number of traits both $\mathbf{X}_j$ and $\mathbf{X}_i$ have. So the distance between $\mathbf{X}_j$ and $\mathbf{X}_i$ [93] is defined as

$$d\left(\mathbf{X}_i, \mathbf{X}_j\right) = \frac{b+c}{a+b+c+d}.$$

For instance, if $P = 6$, $\mathbf{X}_i = (0, 1, 0, 0, 1, 1)$, $\mathbf{X}_j = (1, 1, 0, 0, 0, 1)$, so $a = 2$, $b = 1$, $c = 1$ and $d = 2$, then the distance between objects i and j is 0.33.

We can also use the Jaccard distance [55] to measure the distance between binary data,

$$J_d\left(\mathbf{X}_i, \mathbf{X}_j\right) = \frac{b+c}{b+c+d}.$$

For the same example mentioned above, the Jaccard distance is equal to 0.5.

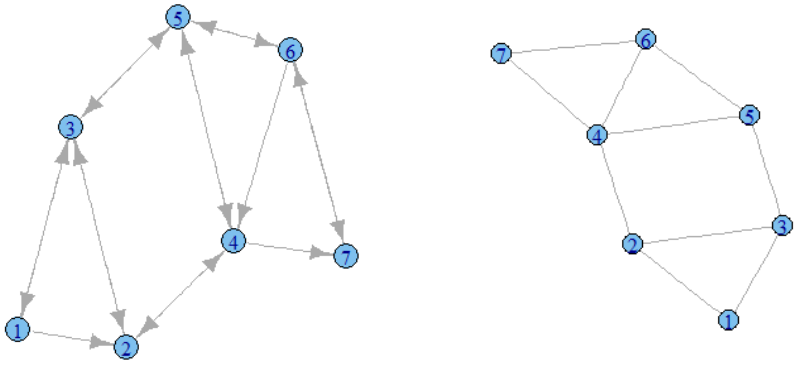
4.3 Graph Notation

A graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is composed of two main sets of elements, nodes and edges. Nodes ($\mathbf{V}$), sometimes called vertices, are research objects, such as people, computers, houses, etc. Edges ($\mathbf{E}\{u, v\}$) are the connections between pairs of nodes (u and v), where u,$v \in \mathbf{V}$. Edges are also known as links.

4.3.1 Undirected and Direct Graphs

In a graph $\mathbf{G}$, for a pair of distinct vertices u and v, if edge $\{u, v\}$ is distinct from edge $\{v, u\}$, then graph G is a directed graph, otherwise, the graph is an undirected graph [66]. Take the graph in Figure 4.1(b) for example, it is an undirected graph, because all the edges are undirected, i.e. the edge $\{4, 7\}$ is same one as the edge $\{7, 4\}$. The graph shown in the Figure 4.1(a) is a directed graph, as not all the edges are the same in both

directions, e.g. the edges connecting vertices 4 and 7 are not mutual, there is only one edge starting from vertex 4 to vertex 7, but without any edges coming back to vertex 4 from vertex 7.



(a) Directed Graph (b) Undirected Graph

FIGURE 4.1: Example Graphs

4.3.2 Adjacency Matrix

Graphs are not the only way to express relationships between vertices, we can also use some mathematical representations, such as the adjacency matrix ($\mathbf{A}\,(a_{ij})$, also known as the sociomatrix). Suppose we have a set of N objects, then its adjacency matrix will be an $N \times N$ binary matrix with all entries are either 0 or 1. 1 indicates a relationship between pairs of objects, 0 represents no relationship. Taking the undirected graph in Figure 4.1(b) for example, its adjacency matrix is expressed as follows:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

The adjacency matrix of an undirected graph is a symmetric matrix, for which diagonal entries are all zero, which implies no self loops. The adjacency matrix of the directed

graph shown in Figure 4.1(a) is:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

This adjacency matrix of a directed graph is potentially an asymmetric matrix, which implies the potentially different relationships between pairs of objects.

4.3.3 Graph Density

A subgraph $G_U = (U, E_U)$ [50] of a graph $G(V, E_V)$ is formed from a vertex subset U, where $U \subset V$, and all connecting edges within U, which is denoted as $E_U \subseteq E_V$. A fully connected graph is a graph in which every pair of distinct vertices is connected by a unique edge. A clique [50] of a graph $G = (V, E)$ is a fully connected subgraph of G. For a given subgraph $G_U = (U, E_U)$ of graph G, its edge density [50] is defined as $2 \times |E_U| / (|U| \times (|U| - 1))$, where $|E_U|$ is the total number of edges in the subset U, $|U|$ is the total number of vertices in subset U. If the edge density tends towards 1, then G_U will be close to being a clique.

4.4 Hierarchical Clustering

Hierarchical clustering [114] is a traditional and well-developed clustering algorithm in grouping data. It groups objects by constructing a hierarchy of clusterings based on the dissimilarities between pairs of objects. There are two types of hierarchical clustering, one is agglomerative hierarchical clustering [114], the other is divisive hierarchical clustering [103]. In the agglomerative hierarchical clustering algorithm, in the beginning, all the objects are their own clusters, then at each iteration, the most similar two clusters are merged into one new cluster and the cluster distances between pairs of clusters are recalculated. This merging continues until in the end, all the objects are in one cluster. Divisive hierarchical clustering works the other way. In the beginning, all the objects are in one cluster, then at each step, the most different cluster is split in two,

repeating this procedure until all the clusters only have single objects. Both methods will result in a hierarchy of clusterings. A single result from the hierarchical clustering is usually obtained by cutting the hierarchy, then all the objects from the same branch will be grouped into a cluster. In this thesis I will only discuss agglomerative hierarchical clustering.

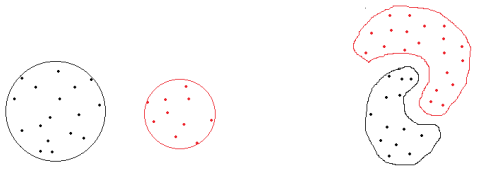
4.4.1 Linkage Methods

The common distance measurements mentioned in Section 4.2 are used to measure the difference between two individual objects. In this section, I will explore the techniques used to measure the distances between pairs of clusters, where a cluster contains more than one object. This type of distance measurement is called the linkage between clusters [93]. Linkages are important measures used in hierarchical clustering and Chameleon hierarchical clustering. Different types of linkages and their advantages will be discussed next.

Complete linkage [107] defines the distance between a pair of clusters A and B, by using the maximum distance among all pairs of objects with one object from cluster A, the other from cluster B. It is expressed as follows:

$$\mathrm{L}\left(\mathbf{A},\mathbf{B}\right)=\max_{a,b}\left(\mathrm{dist}\left(\mathbf{X}_a,\mathbf{X}_b\right)\right):\mathbf{X}_a\in\mathbf{A},\mathbf{X}_b\in\mathbf{B}.$$

Complete linkage is good at dealing with spherical groups, i.e. the ones in Figure 4.2(a), and separating overlapping clusters. But complete linkage has difficulty in identifying non-convex groups, e.g. the ones in Figure 4.2(b) [93].



(a) Convex Groups (b) Non-Convex Groups

FIGURE 4.2: Convex and Non-Convex Groups

Single linkage [74] defines the distance between pairs of clusters, clusters **A** and **B**, by using the minimum distance among all pairs of objects connecting both clusters. It is

expressed as follows:

$$\mathrm{L}\left(\mathbf{A},\mathbf{B}\right)=\min_{a,b}\left(\mathrm{dist}\left(\mathbf{X}_a,\mathbf{X}_b\right)\right):\mathbf{X}_a\in\mathbf{A},\mathbf{X}_b\in\mathbf{B}. \tag{4.1}$$

Single linkage is good at identifying groups with non-ellipse shapes, such as non-convex groups. It can also be used to identify objects which behave very differently from the rest of the objects, so single linkage is a useful tool in identifying outliers or noise points.

The next commonly used linkage is centroid linkage [106]. The distance between clusters **A** and **B** is determined by the centroids of two clusters. The centroid is the mean position of all the objects in the cluster. It is expressed as follows:

$$\mathrm{L}\left(\mathbf{A},\mathbf{B}\right)=\mathrm{dist}\left(\bar{\mathbf{X}}_\mathbf{A},\bar{\mathbf{X}}_\mathbf{B}\right);\bar{\mathbf{X}}_\mathbf{A}=\frac{1}{|\mathbf{A}|}\sum_{\mathbf{X}_a\in\mathbf{A}}\mathbf{X}_a,\ \bar{\mathbf{X}}_B=\frac{1}{|\mathbf{B}|}\sum_{\mathbf{X}_b\in\mathbf{B}}\mathbf{X}_b,$$

where $|\mathbf{A}|$ and $|\mathbf{B}|$ represent the number of objects in clusters A and B respectively. Specifically, centroid linkage requires us to know coordinate data and it tends to form large-sized clusters for the reason of the centroid linkage ignores the shape of clusters and the centroid of the merged two clusters is weighted toward the large cluster. The cluster size refers to clusters' extent in the variable space rather than cluster memberships [68]. So large-sized clusters mean clusters whose points take up a large volume of space.

Another linkage is average linkage [106], which defines distances between pairs of clusters by using the average distance of all pairs of distances with objects coming from different clusters. It can be expressed as follows:

$$\mathrm{L}\left(\mathbf{A},\mathbf{B}\right)=\frac{1}{\text{no. of pairs }\{a,b\}}\sum_{\{a,b\}:\mathbf{X}_a\in\mathbf{A},\mathbf{X}_b\in\mathbf{B}}d\left(\mathbf{X}_a,\mathbf{X}_b\right).$$

The dissimilarity between a point and a cluster is defined to be equal to the average of the distances between this point and each point in the cluster; when two clusters merge, their dissimilarity is equal to the average of the distances between each point in one cluster with each point in the other cluster. So compared with centroid linkage, average linkage is less cluster size dependent [68].

The last commonly used linkage is Ward's linkage [115]. Distances between pairs of clusters A and B are measured by the difference between the sum of squares of the combined cluster resulting from merging clusters A and B, and the sum of the sum of squares of the two individual clusters. It can be expressed as follows:

$$\mathrm{L}\left(\mathbf{A},\mathbf{B}\right)=\mathrm{SS}\left(\mathbf{A},\mathbf{B}\right)-\left(\mathrm{SS}\left(\mathbf{A}\right)+\mathrm{SS}\left(\mathbf{B}\right)\right),$$

$$\mathrm{SS}(\mathbf{A}) = \sum_{a:\mathbf{X}_a \in \mathbf{A}} \left\|\mathbf{X}_a - \bar{\mathbf{X}}_{\mathbf{A}}\right\|^2, \mathrm{SS}(\mathbf{B}) = \sum_{b:\mathbf{X}_b \in \mathbf{B}} \left\|\mathbf{X}_b - \bar{\mathbf{X}}_{\mathbf{B}}\right\|^2, \tag{4.2}$$

$$\mathrm{SS}(\mathbf{A}, \mathbf{B}) = \sum_{\mathbf{X} \in \mathbf{A} \bigcup \mathbf{B}} \left\|\mathbf{X} - \bar{\mathbf{X}}\right\|^2, \bar{\mathbf{X}} = \frac{1}{|\mathbf{A}| + |\mathbf{B}|} \sum_{\mathbf{X} \in \mathbf{A} \cup \mathbf{B}} \mathbf{X}.$$

The goal of Ward's linkage is to minimize the within-cluster sum of squares, so it tends to produce equal-sized spherical clusters and is very sensitive to outliers. Ward's linkage tends to form large-sized volume clusters.

4.4.2 Dendrogram

A dendrogram can help to visualize the cluster hierarchy. A dendrogram is composed of two parts, an axis and a tree structure in the middle. The axis shows dissimilarity levels between clusters. The leaves at the bottom of the plot are the object indices, but listed in a certain order to avoid crossing branches. In agglomerative hierarchical clustering, merges that happened in the beginning stages are shown in the lower part of the dendrogram, merges that happened at the latter stages are shown at the top of the dendrogram. The heights of merges indicate the dissimilarities between the two merged clusters. Sometimes, the dendrograms are displayed by rotating the dendrograms counter clockwise 90 degrees, then heights will be moved to the bottom, the objects will be listed on the right hand side axis.

An example of a dendrogram with 40 objects is shown in Figure 4.3. In this case, Ward's linkage is chosen to measure between cluster distances. In the beginning, all 40 objects are their own clusters. At the first step, the most similar two clusters will be merged together, which are objects 35 and 40, then we need to update the distances between this newly formed cluster and the other clusters by using Ward's linkage. This procedure was repeated 39 times until all objects were in one cluster. All formed clusterings at each iteration can be investigated from this dendrogram.

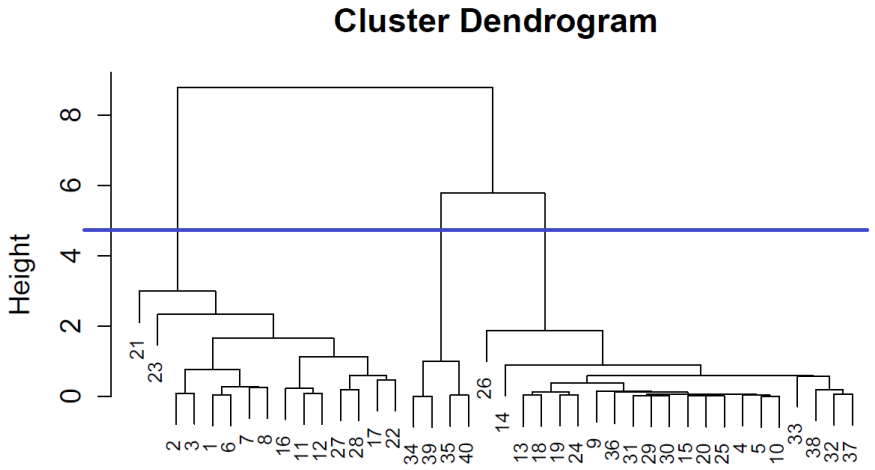


FIGURE 4.3: Dendrogram of Hierarchical Clustering with Added Line at 4.8

There is no single universally agreed criterion for deciding the numbers of clusters [35]. The decisions might be based on the study background and the study purpose. However, the dendrogram can provide visual assistance for the decision, e.g. cutting the hierarchy where there is the a huge jump in distance [52]. Take the dendrogram in Figure 4.3 for example, it is reasonable to set the number of clusters as 3, as a large gap occurs at this stage, e.g. choosing a cutoff line at 4.8. So
cluster 1= $\{21, 23, 2, 3, 1, 6, 7, 8, 16, 11, 12, 27, 28, 17, 22\}$, cluster 2= $\{34, 39, 35, 40\}$, cluster 3= $\{26, 14, 13, 18, 19, 24, 9, 36, 31, 29, 30, 15, 20, 25, 4, 5, 10, 33, 38, 32, 37\}$. Other number of clusters decision criteria will be discussed in Section 4.8.

4.5 Chameleon Hierarchical Clustering

Chameleon hierarchical clustering is a clustering technique proposed by George Karypis and Vipin Kumar [60] which addresses some issues that occur in regular hierarchical clustering. In agglomerative hierarchical clustering (later will be shortened for hierarchical clustering), at each step the most similar two clusters will be merged into one. If there are N objects in a scenario, it will need to take $N - 1$ steps before all objects grouped together, which means linkages between pairs of groups will be updated $N/2$ times. In particular, if the data size N is too large, it will take a very long time to get the full clustering hierarchy. Also, in hierarchical clustering, if objects are merged incorrectly at some stage in the process, they remain merged for the remainder of the steps. There is no opportunity to correct this.

In Chameleon hierarchical clustering, George Karypis and Vipin Kumar [59] proposed a method for grouping large-scaled datasets with diverse cluster shapes by using the internal characteristics of clusters. The intuition of Chameleon hierarchical clustering came from the scenario where users fail to identify a suitable mixture clustering model (e.g. model-based clustering with dissimilarities introduced in Section 4.6) to model the data, or the parameters in the models are difficult to estimate. Chameleon hierarchical clustering mainly consists of three stages: K-NN graph stage, graph partitioning stage and merging stage. In the beginning, we will construct a K-nearest neighbour graph by identifying and connecting the most similar K neighbours to each object. At the second stage, the finest graph will be coarsened into a small-sized graph, this procedure will be repeated several times and the graph formed at the last step is called the coarsest graph. The initial partition will be conducted based on this coarsest graph. The next step is to project the coarsest graph back to the previous stage finer graphs and, at the same time, a refinement technique will be applied to potentially move the bordering objects to adjacent clusters in order to modify the partition. At the last stage, merging the most similar partitions at each step into one and give a hierarchy of possible clusterings.

One characteristic of Chameleon hierarchical clustering is that it uses similarity relationships between objects or transformed similarity relationships data (e.g. dissimilarity data) directly. There is no requirement for coordinate data (the objects' coordinates in Euclidean space). Also, Chameleon hierarchical clustering works much faster than hierarchical clustering, because the partitioning is only conducted on the coarsest graph with fewer objects and it does not calculate all linkages in the original graph but only in the finest partitioned graph. I will describe the algorithm in greater detail in Section 4.5.2.

4.5.1 Chameleon Hierarchical Clustering Notation

There is some terminology that needs to be defined before introducing the procedure for Chameleon hierarchical clustering.

4.5.1.1 Breadth-First Search

At the M-partitioning stage in Chameleon hierarchical clustering, we will use the Breadth-first search (BFS) to form the initial partition. BFS is an algorithm for traversing or searching tree or graph data structures. It can start from any randomly selected vertex,

then explore the adjacent vertices first before moving to the next level neighbours [67]. Breadth-first search can be illustrated by using the example below.

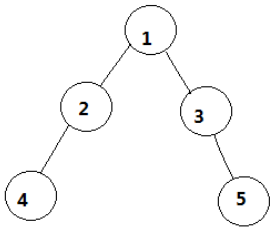


FIGURE 4.4: Breadth-first Search

If the starting point is vertex 1, then the first explored vertices are 2 and 3 before moving to the next level neighbours 4 and 5.

4.5.1.2 M-Way Partition Algorithm and Partitioning Constraint

BFS introduced in Section 4.5.1.1 can help to form the initial bisected partitions if we stopping searching the next object when half of the objected have been searched. This formed bisected partitions can be further recursively bisected into multiple partitions, which means based on the formed bisected partitions, we partition each of them into another two bisected partitions and this process will be repeated for a certain number of times, then this recursive bisection process is known as the M-way partition algorithm [60]. For a given M (the formed number of components) and C (imbalance tolerance, which indicates the relative size difference between the two components and $C > 1$), M-way partition is defined on the formed M components which are disjoint, the size of each component $|V_j|$ is $|V|/(CM) < |V_j| \leq C|V|/M$, where $|V|$ is the total number of vertices in the finest graph, $|V_j|$ is the number of vertices in component j. This is known as the partitioning constraint [61].

4.5.1.3 Gain

For a pair of adjacent clusters (A and B), gain is used to measure whether an object should be moved to its adjacent cluster or not, i.e. whether an object belonging to cluster A should be moved to cluster B or not. (4.3) is the definition of gain for i^{th} object [60].

$$g_i(B) = \sum_{j:e_{i,j}\in\mathbf{E},j\in B} s_{i,j} - \sum_{j:e_{i,j}\in\mathbf{E},j\in A} s_{i,j}, \text{ for } i \in A, \tag{4.3}$$

$e_{i,j}$ denotes the edge connecting vertices i and j, $\mathbf{E}$ is the edge set. $s_{i,j}$ is the similarity between objects i and j. In particular in (4.3), A and B have to be adjacent clusters

sharing at least one common boundary. The former part of (4.3) counts the total similarities between object i and other objects coming from B. The latter part of (4.3) counts the total similarities between object i and the rest of objects from A.

4.5.1.4 Bipartitioning

Here I describe two definitions used in the minimum bipartition problem (these two definitions will be used in the Merging stage in Chameleon hierarchical clustering in calculating relative interconnectivity and relative closeness).

Definition 1. Fixing a parameter $\frac{1}{2} \leq \alpha < 1$, the minimum α balanced bipartition problem is to find two subsets S and $V \setminus S$ ($S \cup (V \setminus S) = V$ and $S \cap (V \setminus S) = \varnothing$), so that $|S| \leq \alpha |V|$ and $|V \setminus S| \leq \alpha |V|$ and the total similarities connecting the two subsets are minimized [88].
Note: $V \setminus S$ is the complementary set of V; $|V|$, $|S|$ and $|V \setminus S|$ are the total number of edges in V, S and $V \setminus S$ respectively.

Definition 2. Assuming $|V|$ to be even, the minimum bipartition problem is to find two subsets S and $V \setminus S$, where $|S| = |V \setminus S| = \frac{|V|}{2}$, with minimum total similarities connecting S and $V \setminus S$. It can be regarded as the special case of Definition 1 with $\alpha = \frac{1}{2}$ [88].
Note: Code is given in Appendix A.2 [69]

Minimum bipartitioning starts by partitioning a graph into two roughly equal-sized parts (subject to Definition 1), S and $V \setminus S$, then swapping elements (such as $a \in S$ and $b \in V \setminus S$) from one part to the other which can reduce the total similarities connecting two parts.

4.5.1.5 RIRC Linkage

In Chameleon hierarchical clustering, George Karypis and Vipin Kumar [60] proposed a new linkage, RIRC linkage, which is a function of relative interconnectivity and relative internal closeness. For two non-singleton clusters (clusters with at least two objects) A and B, interconnectivity ($\mathrm{EC}_{A,B}$) is defined as the total edge similarities connecting A and B, $\mathrm{EC}_{A,B} = \sum_{e_{a,b} \in \mathbf{E}, a \in A, b \in B} s_{a,b}$. Internal connectivity (EC_A) is the minimal total edge similarities which can split A into two roughly equal-sized parts. The formula of the relative interconnectivity (RI) between A and B is defined as follows:

$$\mathrm{RI}(A, B) = \frac{\mathrm{EC}_{A,B}}{\frac{\mathrm{EC}_A + \mathrm{EC}_B}{2}}.$$

The closeness between a pair of clusters A and B is denoted as $\overline{\text{EC}}_{A,B}$, it is the average similarity of edges connecting two different clusters $\overline{\text{EC}}_{A,B} = \frac{\text{EC}_{A,B}}{|E_U|}$, where E_U is the set of edges connecting clusters A and B, $E_U = \{e_{a,b}\}$, for $a \in A$ and $b \in B$, $|E_U|$ is the total number of edges in E_U. The internal closeness ($\overline{\text{EC}}_A$) of cluster A can be measured by the average similarity of edges bisecting A into roughly equal sized parts [59]. $\overline{\text{EC}}_A = \frac{\text{EC}_A}{|E_I|}$, E_I is the set of edges bisecting cluster A, $E_I = \{e_{a,a'}\}$, for $a \in A$ and $a' \in A$, $|E_I|$ is the total number of edges in E_I. The definition of relative internal closeness between two non-singleton clusters A and B is expressed as follows:

$$\text{RC}(A, B) = \frac{\overline{\text{EC}}_{A,B}}{\frac{|A|}{|A|+|B|}\overline{\text{EC}}_A + \frac{|B|}{|A|+|B|}\overline{\text{EC}}_B},$$

where $|A|$ is the total number of edges in A and $|B|$ is the total number of edges in B.

The RIRC linkage is a function of $\text{RI}(A, B)$ and $\text{RC}(A, B)$,

$$\text{RIRC}(A, B) = \text{RI}(A, B) \times \text{RC}(A, B)^{\alpha_0}, \tag{4.4}$$

where α_0 is determined by the users. If $\alpha_0 > 1$, more emphasis will be given to the relative internal closeness; if $\alpha_0 < 1$, then more emphasis will be given to the relative internal connectivity. For each step, we will merge the pair of components with the maximal RIRC into one and update the RIRC between the newly formed component with the other adjacent components, then repeat this procedure until no more merges occur.

4.5.2 Chameleon Hierarchical Clustering Algorithm

Chameleon hierarchical clustering algorithm can be mainly divided into three stages. The first stage involves constructing a K-Nearest Neighbours graph (K-NN graph), which means for each object, the most similar K objects will form edges to this selected object. K needs to be set before constructing the K-NN graph. We then need to coarsen the original graph to a coarser graph by using Heavy Random Matching (HRM) [60]. HRM coarsens a graph by merging possible pairs of unmerged objects into one. All vertices will be visited in a random order, then we combine each vertex with an unmerged vertex with the highest similarity to it. The selected unmerged objects are the ones have not been merged with any visited objects in the previous steps, which means in each iteration each object cannot merge with more than one object. This process will be repeated until there are no more than $M \times C$ clusters left [62], M is the number of components that will

be formed in the M-partitioning stage, C is a user-specified value. We then apply the M-way partition algorithm [60] to this coarsest graph to form the initial M components. The formed partition is the optimal M-partitioning only to the coarsest graph, however, we still need to reflect them back to the finest graph iteratively (original graph). This partition might not be as good for the finer graphs. So a refinement technique will be used to potentially change the membership of bordering objects. The last stage is the merging stage, merging the most similar partitions (measured by RIRC) at each step into one and give a hierarchy of possible clusterings.

The procedure of Chameleon hierarchical clustering can be illustrated by Figure 4.5,

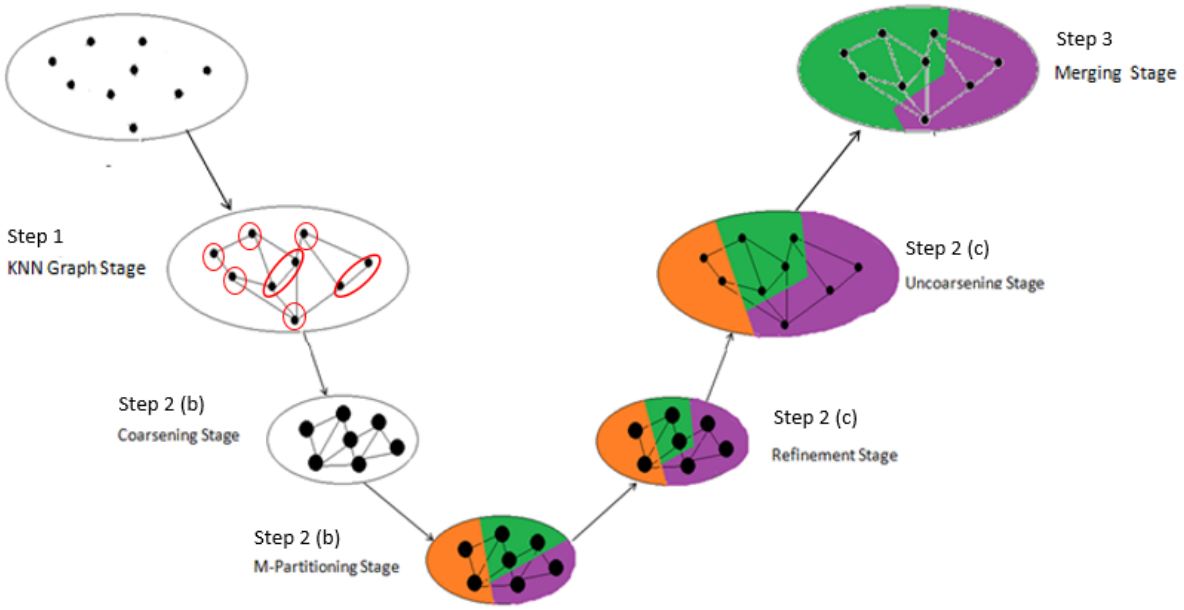


FIGURE 4.5: The Procedure of Chameleon Hierarchical Clustering

The algorithm is given below:

1. K-Nearest Neighbour Graph Stage (see Section 4.5.2.1 for greater detail)
   (a) Set a value for K, which is the number of neighbours of each object.
   (b) Construct a K-Nearest Neighbour graph by identifying and connecting the most similar K neighbours to each object.
2. Graph Partitioning Stage
   (a) Set a value for M, which is the number of components that will be formed at the end at M-partitioning of the stage and also chose a value for $C > 1$.
   (b) Coarsening Stage (see Section 4.5.2.2 for greater detail)

i. Visit each object in a random order and merge the object with an unmerged and connected object with the maximum similarity. Skip the objects which have been merged in the previous steps.

Detail:

A. In each iteration, each object can merge with at most one object.

B. The similarities (weights) between the newly formed cluster and its adjacent clusters are the total similarities between its cluster member vertices with those adjacent clusters [60].

ii. As the aim is to form an M-partition graph, so the number of components left should be a function of M. Repeat step 2 (b) i until the number of components is no more than $M \times C$. The formed graph at the last coarsening stage is called the coarsest graph.

(c) M-partitioning Stage (see Section 4.5.2.3 for greater detail)

i. The initial bisection is formed based on the breadth-first algorithm. It starts with one randomly picked vertex, then grows a region around it until two parts have roughly equal size.

ii. Repeat step 2 (c) i until M components formed and these M components subject to the partitioning constraint (see Section 4.5.1.2 for further detail), this process is called the M-way partition algorithm [60].

iii. If M is not large, steps 2 (c) i and 2 (c) ii will be repeated by starting from a different vertex many times until finding all the possible M-partitions, as different starting points may create different M-partitions [58] (We usually only create at most a small number of partitions for the reason given in Section 4.5.2.3). The optimal M-partition at this stage will be the one with the minimum total between-components similarities at the coarsest graph stage and this partition will be used in the following stages.

(d) Uncoarsening Stage (see Section 4.5.2.4 for greater detail)

i. The partition of the coarser graph will be projected back to the next level finer graph successively by splitting the multi-nodes into next level individual nodes.

ii. Apply the hyperedge refinement (HER) algorithm (see Section 4.5.2.4 for further detail) to the bordering vertices. The movement of a bordering vertex depends on the gain (see (4.3) for further detail) and subjects to the partitioning constraint (see Section 4.5.1.2 for further detail).

iii. Repeat steps 2 (d) i and 2 (d) ii until the graph is projected back to the finest graph (the original graph).

3. Merging Stage (see Section 4.5.2.5 for greater detail)

   (a) Set a value for α_0 in (4.4). In bipartitioning, if the number of objects in each component is an even number, we set $\alpha = 0.5$, otherwise, we set $\alpha = 0.55$, for which can roughly bipartition components into equal-sized parts.

   (b) Calculate the pairs of clusters' RIRC value by using (4.4), merge the pair of clusters with the maximal RIRC value, then update the RIRC between the newly formed components and its adjacent components.

   (c) Repeat step 3 (b) until no more merges occur.

   (d) The optimal number of clusters will be decided by cluster decision techniques introduced in Section 4.8.

More details will be given in the following sections and decisions in hyperparameters will be further explored in Section 7.3.9.

4.5.2.1 K-Nearest Neighbour Graph Stage

For a given K, the K-nearest neighbours graph (K-NN graph) is a process to connect all objects with their nearest K neighbours. Take the object i in a graph for example, all similarities between it and the rest of objects will be listed in an decreasing order, then we form the connections between it and the most similar K neighbours (with the highest K similarities). This stage will be illustrated using the following example.

Suppose there are 30 objects generated from two different clusters, $\mathrm{N}(0, 0.5)$ and $\mathrm{N}(20, 0.5)$, where all red points are generated from $\mathrm{N}(0, 0.5)$, all green points are generated from $\mathrm{N}(20, 0.5)$. The dissimilarity data can be obtained by using the Euclidean distance mentioned in Section 4.1, so we can then obtain the similarity from the reciprocal of the dissimilarity data. However, the reciprocal transformation might be unstable when the values approach to zero, so a possible alternative transformation is $\exp(-x)$. If $K = 2$, then all objects will connect to their two most similar neighbours. Each object will have at least two connected neighbours, which is shown in Figure 4.6. The figures printed on the edges are the similarities between the pairs of objects. The higher a value is, the stronger the similarity is.

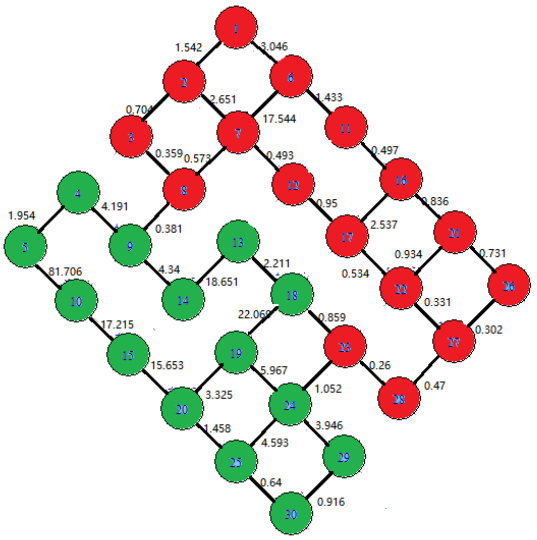


FIGURE 4.6: K-NN Graph, $K = 2$

4.5.2.2 Coarsening Stage

There are several matching techniques for coarsening. Random matching (RM) [60] uses the random matching algorithm to coarsen graphs. All vertices are visited in a random order. If vertex u has not been matched yet, it will merge with a randomly selected unmerged vertex v. The second matching technique is the heavy edge matching (HEM) [60], its procedure is similar to RM, each vertex is visited in a random order, but will only merge with its most similar unmerged vertex. Another matching technique is light edge matching (LEM) [60], which is an opposite technique to the HEM, it seeks to merge vertex u with another unmerged vertex with the minimum similarity. So the average degree of the finer graph produced by the LEM is much higher than the original graph, then it will take less time to coarsen a graph. The last matching technique is heavy clique matching (HCM) [60], which computes a matching by collapsing vertices which have high edge density. In this thesis, I will use the HEM to coarsen graphs as it only visits the most similar unmerged vertex. According to the experiments conducted by Karypis and Kumar [60], the complexity of computing an HEM is O($|E|$) (E is the total number of edges), which is similar to the time for computing the RM. However, the

aim of the clustering is to group the most similar objects into one, so HEM is preferred over RM. One of the weakness of HEM is that it does not guarantee all vertices will merge with the vertices with the maximum weight. This is because all vertices are visited in a random order and the vertex with the maximum weight might have already been merged with another vertices.

In the HEM implementation, each vertex will be visited in turn in a random order. Each visited vertex can merge with another unmerged vertex which has the maximum similarity to this current visited vertex. The merged vertices have to have two qualities. Firstly, those unvisited vertices have to be vertices which have not been merged with other vertices in the previous steps. Secondly, the merged vertex has to be the most similar vertex to the current visited vertex. However, if this visited vertex cannot merge with its the most similar vertex because this vertex has merged with other vertices, we can merge this visited vertex with the vertex with the second largest similarity. If a vertex fails to find a similar unvisited vertex, then we will leave this vertex alone.

At the coarsening stage, all merged together vertices will be treated as one component. When they are treated as one component, the internal edges within the newly formed component will be ignored, which means there will be no within-component edges inside the new components. The between-components edge similarities are the sum of all the individual components' vertices' between-components edge similarities.

The coarsening stage will be repeated until there are no more than $M \times C$ components left. $C > 1$ is a user-specified value, M is the number of components expected at the M-partitioning stage. These $M \times C$ components formed the coarsest graph G_0. The easiest method to conduct M-partitioning is to repeat the coarsening process many times until only M components are left, then these M components form the initial clustering. However, these formed partitions will not always have the minimum total similarities between pairs of components, so the M-way partition algorithm [60] is used to obtain the initial clustering with M components from the $M \times C$ initial components.

In our example from Figure 4.6, to simplify the calculation, we set $M = 2$, $C = 2$, all 30 vertices will be visited in a random order, 27 16 6 11 21 25 4 29 5 28 24 12 1 18 23 8 14 30 17 7 10 22 9 15 19 20 2 26 13 3. The first visited vertex is 27, the most similar and connected vertex to it is vertex 28, then we will merge these two into one and name it as the new component 27. The internal edges will be deleted (remove $e_{27,28}$), the updated similarities between it and its adjacent clusters are the sum of similarities between vertex 27 and vertex 28 to their adjacent clusters. Later, the most similar and connected unvisited vertex to vertex 11 is 6, but it has merged with vertex 7, and there

are no other connected vertices to vertex 11, so we will leave it alone. The components formed after the first coarsening stage is shown in Figure 4.7.

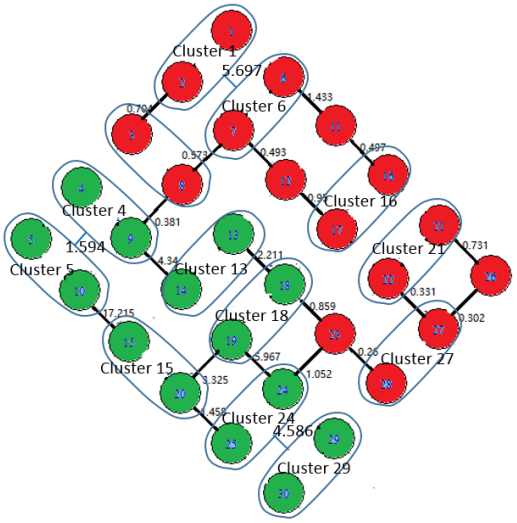


FIGURE 4.7: First Coarsened Graph

At the second coarsening stage, all these newly formed components will be visited in a random order, 3 1 6 11 12 5 4 13 15 18 16 21 23 24 29 27 26. The clustering after the second coarsening stage is shown in Figure 4.8.

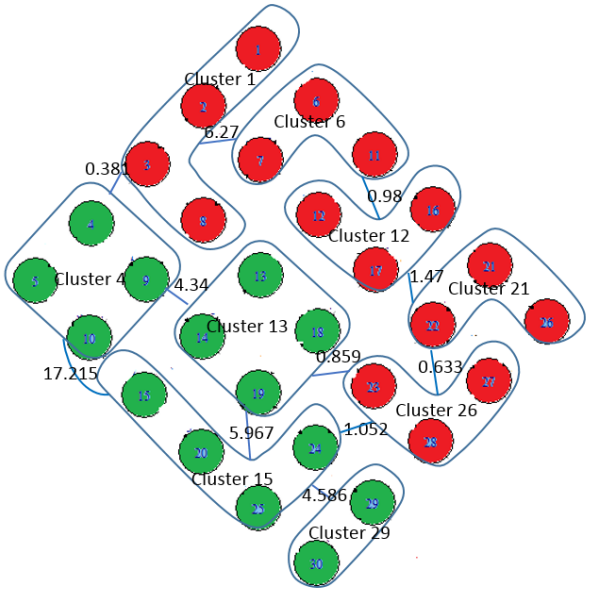


FIGURE 4.8: Second Coarsened Graph

At the third coarsening stage, all these newly formed components will be visited in a random order, 13 21 6 1 29 23 15 12 4. The clustering after the third coarsening stage is shown in Figure 4.9.

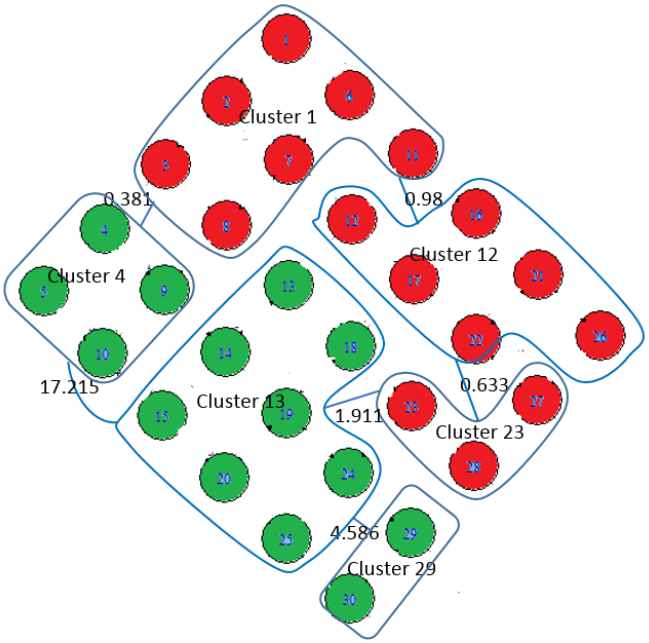


FIGURE 4.9: Third Coarsened Graph

At the third coarsening stage, all these newly formed components will be visited in a random order, 1 29 13 12 23 4. The clustering after the fourth coarsening stage is shown in Figure 4.10.

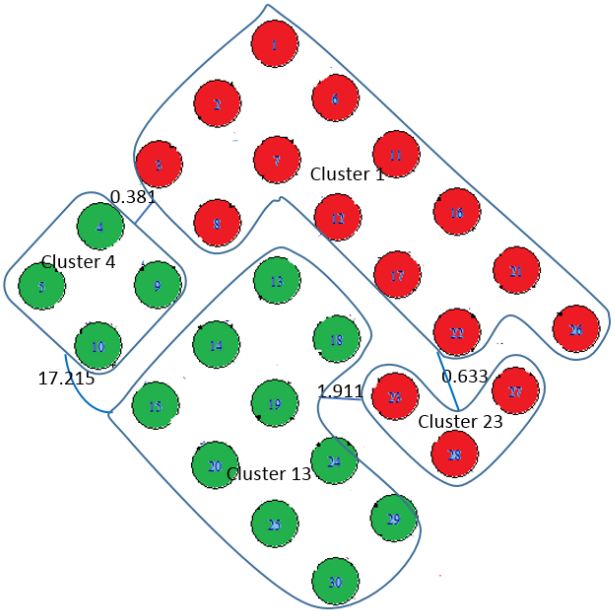


FIGURE 4.10: Fourth Coarsened Graph

The components formed in Figure 4.10 can be simplified as the ones in Figure 4.11.

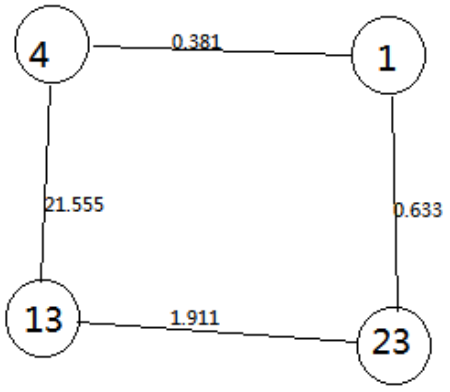


FIGURE 4.11: Simplified Version of Graph in Figure 4.10

We stop coarsening at this stage as the remaining number of clusters is equal to $M \times C = 4$.

4.5.2.3 M-Partitioning Stage

The aim of conducting coarsening stages is to successively reduce the graph size so that the partition based on the simpler graph does not have heavier total between-component similarities than the partition obtained from the original graph [60].

Firstly, we roughly divide the coarsest graph ($G_0 = (V_0, E_0)$) into two different components (A and B) with roughly equal number of vertices if possible. In detail, suppose there are P_0 components left in the coarsest graph G_0. The initial bisection clustering is formed by applying the breadth-first algorithm, so we randomly select one vertex as the starting vertex, then group all the connected components to this selected component (e.g. component A). The remaining components will then be grouped into the other component (e.g. component B). This bisection procedure will be repeated until M components form and these M components subject to the partitioning constraint (see Section 4.5.1.2 for further detail), $|V|/(CM) < |V_m| \leq C|V|/M$, where $|V_m|$ is the number of vertices in each component [61]. This is called the M-way partition algorithm [60]. It is noticeable that different starting vertices will lead to different partitions, so if M is not large, we propagate all the possible initial partitions at the coarsest graph and choose the optimal partition with the minimal total between component similarities

subject to the partitioning constraint as the initial partition. However, the improvement in reducing the total between-component weighs will be at the expense of the longer running time. In addition, the minimal total between-component similarities in the coarsest graph may not lead to the minimal total between-component similarities in the finer graphs [58]. In the experiments reported in Karypis et al. [58], it found that ten initial partitions at the coarsest graph can reduce the total between-component similarities by 3%-4% and the running time will not increase too much. Computing and propagating more partitions will not reduce the total between component similarities largely [58]. So based on this finding, we only propagate at most 10 partitions.

The partitioning constraint in our example is $\frac{30}{2\times 2} = 7.5 < |V_m| \leq \frac{30\times 2}{2} = 30$. In all possible partitions, the optimal partition formed by grouping component 1 in the coarsest graph in Figure 4.11 into component A, the rest of components assigned to component B. Both components satisfy the partitioning constraint, so they are the initial partition of the coarsest graph in Figure 4.11, component A= $\{1, 13, 23\}$ and component B= $\{4\}$.

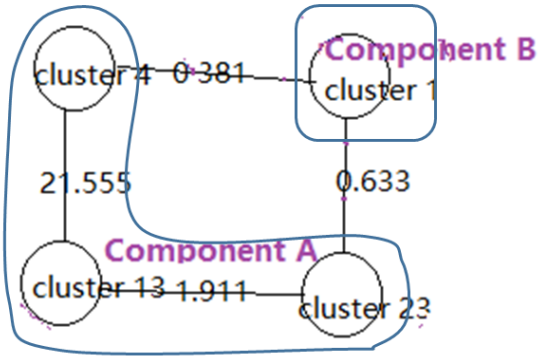


FIGURE 4.12: Initial Bisection Partitioning

4.5.2.4 Uncoarsening Stage

The idea of hierarchical clustering is to produce a hierarchy of clusterings. In each iteration, the most similar two clusters will be merged into one larger cluster. The drawback of hierarchical clustering is that once two clusters are merged into one cluster, it is impossible to separate them in the later stages. The merged objects will be kept together until the last stage. Chameleon hierarchical clustering successfully overcomes this limitation. Merged groups might be split again at the uncoarsening stage, wrong combinations have a chance to be corrected at this stage.

The next stage is the uncoarsening and refinement stage. At the uncoarsening stage, the initial formed partition will be projected back to the next level finer graph. However, the partition with minimal total between component similarities shown in the coarsest graph might still not form the best clustering with minimal total between component similarities at the finest graph, so a refinement algorithm is required to reduce the total between component similarities at each uncoarsening stage without violating the partitioning constraint. There are several different refinement methods. The first one is Fiduccia and Mattheyses (FM) refinement algorithm [36], which starts with an initial bisected partition and changes the membership of any vertices until it finds two components with minimal total connecting similarities and does not violate the partitioning constraint, which means this process will be repeated many times until the total connecting similarity between pairs of adjacent components cannot be reduced any more. However, in some cases, it cannot find two such subsets, then the modification is that the algorithm will stop with a local minimal total connecting similarities [64]. The other technique is the hyperedge refinement (HER), which starts with an initial bisected partition and changes the membership of bordering vertices until two components with minimal total connecting similarities are achieved and does not violate the partitioning constraint. This method is a modified method based on FM refinement algorithm, and will be the method used in this thesis. This refinement algorithm is similar to the FM algorithm, but the advantage over the FM algorithm is that it only checks the vertices lying on the border of two adjacent clusters, which will reduce the time taken.

In both the FM refinement and the HER refinement algorithms, the moving of a vertex is determined by its gain value (4.3). We only move vertices with positive gains, if the gain is a negative value, the vertex will be kept in its current cluster. In particular, the gain can only be compared between two clusters (the current cluster and a neighbouring cluster), it cannot be compared between more than two clusters at a time. For the cases of a vertex with more than one positive gain, we will move the vertex to the adjacent cluster with the maximal positive gain.

In Figure 4.12, there are three bordering vertices, 1, 4, 23. All of them have negative gains, so we do not need to make any changes. The next level finer graph is shown in Figure 4.13.

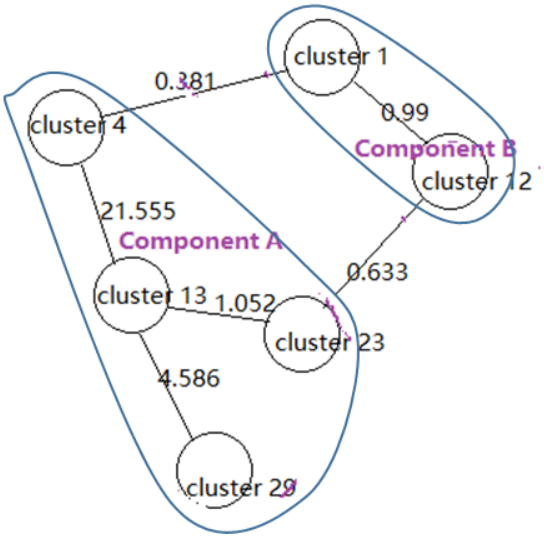


FIGURE 4.13: First Uncoarsening Phase

In the finer graph in Figure 4.13, the bordering vertices are 1, 4, 12, 23, all of them have negative gains, so there is no need to move any of them to the adjacent component.

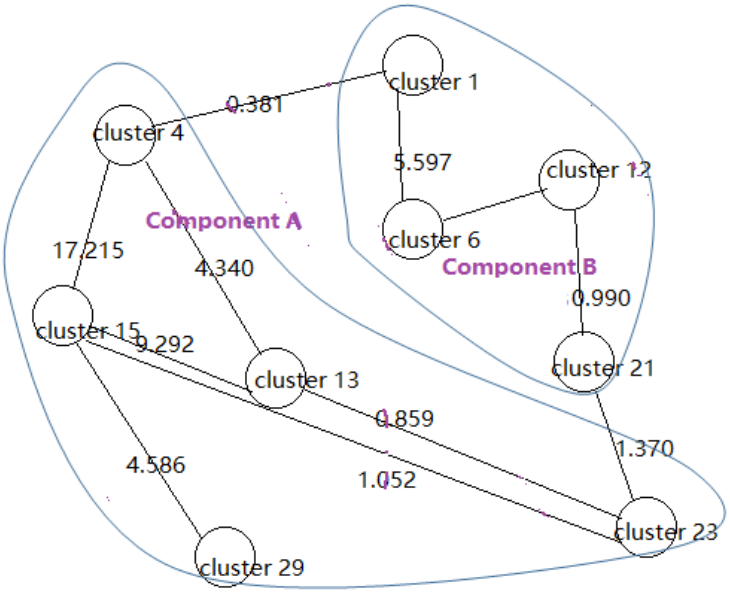


FIGURE 4.14: Second Uncoarsening Phase

In the finer graph in Figure 4.14, all the bordering vertices are 1, 4, 13, 15, 23. The next level finer graph is shown in Figure 4.15.

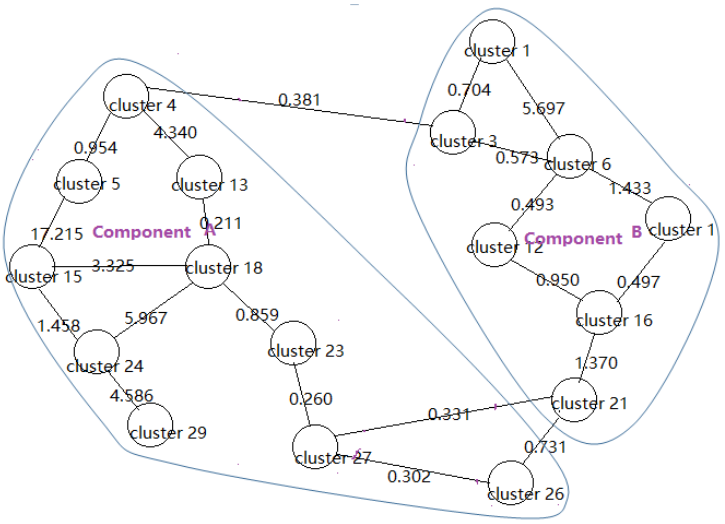


FIGURE 4.15: Third Uncoarsening Phase

The bordering vertices in Figure 4.15 are 3, 4, 21, 26, 27. Only vertex 27 has positive gain, all the other vertices have negative gains, so we need to move vertex 27 from component A to component B subject to the partitioning constraint. The graph after refinement is shown in Figure 4.16.

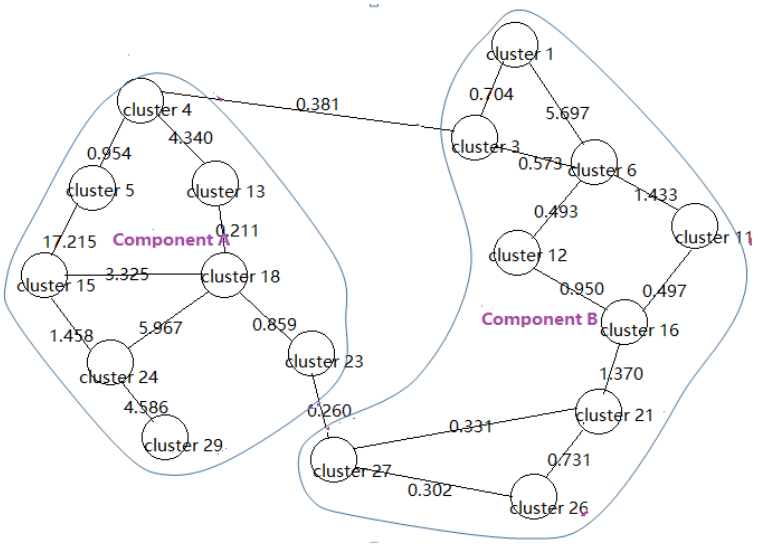


FIGURE 4.16: Refinement Phase

The next level finer graph is shown in Figure 4.17.

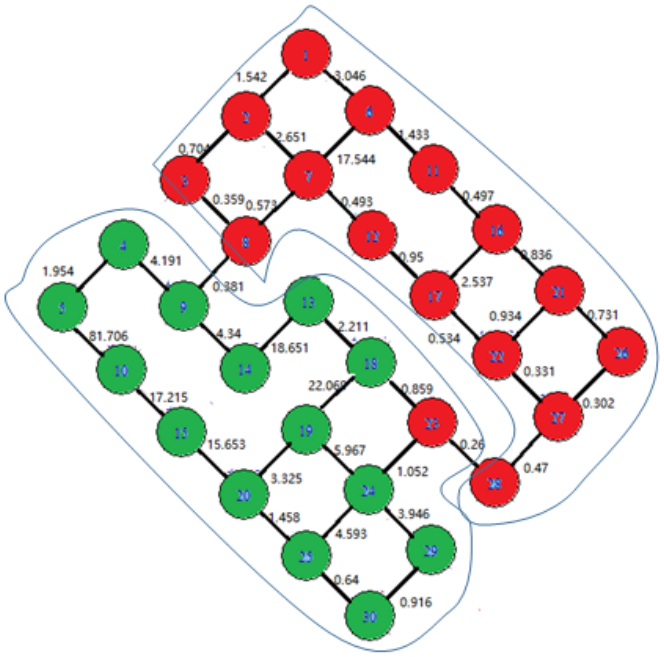


FIGURE 4.17: Forth Uncoarsening Phase

The bordering vertices are 8, 9, 23, 28, all of them have negative gains, so there is no need to move any vertices.

4.5.2.5 Merging Stage

The merging stage is a process to correct any unnecessary partitioning that happened at the M-partitioning stage by merging these components back into one. The merging stage merges similar pairs of components into one by measuring their RIRC. RIRC was proposed by George Karypis and Vipin Kumar [59] to capture more characteristics of graphs by using the graph statistics, graph connectivity and closeness. Details of RIRC have been shown in Section 4.5.1.5, where the default α_0 is set to be 1 if no further information can be used in balancing the weight between the relative connectivity and relative closeness.

At this stage, we will check all pairs of components and merge the pair of clusters with the maximal RIRC, then update the RIRC between the newly formed components and adjacent components and repeat the merging stage until no more merges occur.

In our example from Figure 4.17, we use the default $\alpha_0 = 1$. To bipartition A and

B, If $|A|$ or $|B|$ is an even number, we set $\alpha = 0.5$, otherwise, we set $\alpha = 0.55$, for which can closely equally bipartition components. The internal connectivity of component A is 0.590, the internal connectivity of component B is 1.447, the interconnectivity between components A and B is 0.641. The internal closeness of component A is 11.220, the internal closeness of component B is 2.008, the intercloseness between components A and B is 0.321. So $RIRC_{A,B}$ is 0.031. The hierarchy of clustering (two clusterings) is generated with one clustering has one cluster (including all green and red points) only, the second clustering has two clusters (one cluster is all green objects plus 23, the other cluster is all red objects except for 23). By comparing Pearson version of Hubert's τ of these two clusterings, the estimated clustering with the maximal PH is the one shown in Figure 4.17 (PH of the clustering with two components is 0.825). More details about Pearson version of Hubert's τ will be introduced in Section 4.8.5. The reason for using Pearson version of Hubert's τ (PH) was it obtained estimated clustering results closer to the true classifications with higher Adjusted Rand Index (see Section 4.10.3 for details) compared with Average Silhouette Width (see Section 4.8.3 for details) and Calinski and Harabasz Index (see Section 4.8.4 for details) based on the preliminary runs. We can see that the estimated clustering is very similar to the true classification, only one object, 23, was assigned to the wrong cluster. The adjusted Rand index comparing the estimated clustering and the true classification is 0.866, which is reasonably high.

4.6 Model-Based Clustering

Model-based cluster analysis is a technique which uses a finite mixture model to identify a clustering structure. It assumes that the data are generated from a distribution containing different component densities with some set of mixing proportions [34]. Model-based clustering can be preferred over hierarchical clustering as it can estimate the probabilities of an object belonging to different components, while hierarchical clustering gives hard cluster assignments. So in this section, I will explore the EM algorithm in estimating the model-based clustering model which was described by Jeffrey D. Banfield and Adrian E. Raftery [15].

4.6.1 EM Algorithm for Mixture Models

Assuming all the objects are generated from a finite mixture of J multivariate normal distributions with certain mixing proportions and the coordinate data $\mathbf{X}$ is known, then the finite mixture model form can be written as follows:

$$f\left(\mathbf{X}_i \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{T}\right) = \sum_{j}^{J} \pi^j \phi\left(\mathbf{X}_i; \boldsymbol{\mu}_j, \mathbf{T}_j\right),$$

where $\phi(\cdot)$ is the multivariate Gaussian distribution, $\boldsymbol{\mu}_j$ is the mean vector of the j^{th} Gaussian component, $\mathbf{T}_j$ is the j^{th} covariance matrix. π^j indicates the proportion of the population in the j^{th} component among all components, $0 < \pi^j \leq 1$, $\sum_{j=1}^{J} \pi^j = 1$. If only dissimilarity or similarity data are available, meaning coordinate data are unknown, then the MDS approaches introduced in Chapter 3 can be used to construct a data configuration before modeling.

As mentioned in Chapter 2, the EM algorithm [30] simplifies estimation by introducing latent variables $\mathbf{Z} = (Z_1, \cdots, Z_N)$, which are distributed according to a multinominal distribution and take values from 1 to J. If $Z_i = j$, it indicates that object i belongs to the j^{th} component. The log likelihood of the complete data is expressed as follows:

$$l\left(\boldsymbol{\mu}, \mathbf{T}, \boldsymbol{\pi} \mid \mathbf{X}, \mathbf{Z}\right) = \sum_{i=1}^{N} \sum_{j=1}^{J} \log\left[\left(\pi^j \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j, \mathbf{T}_j\right)\right)^{I(Z_i=j)}\right],$$

where $I(Z_i = j)$ is an indicator function equal to 1 if the statement that the i^{th} object belongs to the j^{th} component is true, 0 otherwise. The EM algorithm uses an iterative method to find the maximum likelihood of the estimated parameters. It is composed of two steps, an E step and an M step. At the E step, the expectation function of complete data log-likelihood given the current estimated parameter values is calculated. At the M step we find parameter values which maximize the expected log-likelihood from E step.

- The expected log-likelihood function of the complete data is shown as follows:

$$Q\left(\boldsymbol{\mu}, \mathbf{T}, \mathbf{p} \mid \boldsymbol{\mu}, \mathbf{T}, \mathbf{Z}\right) = \sum_{i=1}^{N} \sum_{j=1}^{J} E\left\{I(Z_i = j) \mid \mathbf{X}_i, \boldsymbol{\mu}_j, \mathbf{T}_j\right\}\left[\log \pi^j + \log \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j, \mathbf{T}_j\right)\right]. \tag{4.5}$$

- E step: The conditional expectation of $I(Z_i = j)$ at the t^{th} iteration is denoted as $\boldsymbol{\omega}_i^{(t)}$, which is found as follows:

$$\omega_i^{j,(t)} = \mathrm{E}\left(I(Z_i = j) \mid \boldsymbol{\pi}^{(t-1)}, \boldsymbol{\mu}^{(t-1)}, \mathbf{T}^{(t-1)}\right) = \frac{\pi^{j,(t-1)}\phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)}\right)}{\sum_{k=1}^{J}\pi^{k,(t-1)}\phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_k^{(t-1)}, \mathbf{T}_k^{(t-1)}\right)}.$$

- M step: Find out the values for the distribution parameters which maximize $Q\left(\boldsymbol{\mu}, \mathbf{T}, \boldsymbol{\pi} \mid \boldsymbol{\pi}^{(t)}\boldsymbol{\mu}^{(t)}, \mathbf{T}^{(t)}, \mathbf{Z}\right)$.

$$\pi^{j,(t)} = \frac{\sum_{i=1}^{N}\omega_i^{j,(t)}}{N}, \text{ for all } j = 1, \cdots, J,$$

$$\boldsymbol{\mu}_j^{(t)} = \frac{\sum_{i=1}^{N}\omega_i^{j,(t)}\mathbf{X}_i}{\sum_{i=1}^{N}\omega_i^{j,(t)}}, \text{ for all } j = 1, \cdots, J,$$

$$\mathbf{T}_j^{(t)} = \frac{\sum_{i=1}^{N}\omega_i^{j,(t)}\left[\mathbf{X}_i - \boldsymbol{\mu}_j^{(t)}\right]\left[\mathbf{X}_i - \boldsymbol{\mu}_j^{(t)}\right]^T}{\sum_{i=1}^{N}\omega_i^{j,(t)}}, \text{ for all } j = 1, \cdots, J.$$

The E and M steps are repeated many times until all the parameters converge. More details about assessing convergence have been given in Section 2.3.1. The conditional expectation of $\mathbf{Z}$ at the last iteration will then be used to cluster objects, each object will be grouped into the component with the largest ω_i^j, for $1 \leq j \leq J$.

4.7 Bayesian Model-based Clustering with Dissimilarities (BMBCD)

A proposed technique for estimating model-based clustering on dissimilarity data with a Bayesian approach will be introduced in this section [87]. The BMBCD approach combines both multidimensional scaling and model-based clustering. It extends the BMDS introduced in Section 3.2 from a single normal density to a mixture of normal distributions in order to group objects. So if the only available data are dissimilarity data $\mathbf{D}\left(d_{ik}\right)$ or transformed dissimilarity data (e.g. similarity data, $1/\mathbf{D}\left(d_{ik}\right)$), we can still use a model-based clustering approach to model the data.

The true dissimilarity between unknown objects i and k in a P-dimensional Euclidean

space is usually defined as follows:

$$\delta_{ik} = \sqrt{\sum_{p=1}^{P} (X_{ip} - X_{kp})^2}.$$

The relationship between the observed dissimilarities and true dissimilarities can be expressed as:

$$d_{ik} = \delta_{ik} + \varepsilon_{ik}.$$

As all distances are all non-negative, so we assume the observed dissimilarity follows a truncated Gaussian distribution,

$$d_{ik} \sim N\left(\delta_{ik}, \sigma^2\right) \mathrm{I}\left(d_{ik} > 0\right), i \neq k, i, k = 1, \cdots, N.$$

Note that δ_{ij} is related to $\mathbf{X}_i$ which denotes the i^{th} object configuration. To represent the clustering, we assume $\mathbf{X}_i$ is modeled by a finite mixture of multivariate normal distributions,

$$\mathbf{X}_i \sim \sum_{j=1}^{J} p^j \phi\left(\boldsymbol{\mu}_j, \mathbf{T}_j\right),$$

where J is the number of clusters and $\phi(\cdot)$ is a multivariate normal density. The likelihood function can be expressed as follows:

$$\mathcal{L}(\mathbf{X}, \sigma^2; \mathbf{D}) \propto (\sigma^2)^{-m/2} \cdot \exp\left[-\frac{1}{2\sigma^2}\mathrm{SSR} - \sum_{i=1}^{N}\sum_{i>k} \log \Phi\left(\frac{\delta_{ik}}{\sigma}\right)\right],$$

where $SSR = \sum_{i=1}^{N-1} \sum_{k=i+1}^{N} \left(\delta_{ik} - d_{ik}\right)^2$ and $\Phi(\cdot)$ is the cumulative density function of a standard normal distribution.

The prior distributions for Bayesian model-based clustering with dissimilarities (BMBCD) [87] are modeled as follows:

$$\sigma^2 \sim \mathrm{IG}\left(a, b\right),$$

$$\left(p^1, \cdots, p^J\right) \sim \mathrm{Dirichlet}\left(1, \cdots, 1\right),$$

In Dirchlet $(\alpha_1, \cdots, \alpha_J)$, the smaller the α_j (for $j = 1, \cdots, J$) is, the more sparse the distribution will be, so the distribution with $\alpha_j = 1, \forall j$ will cover a wider range of values giving less prior information about $\mathbf{p}$.

$$\boldsymbol{\mu}_j \sim \phi\left(\boldsymbol{\mu}_{0j}, \mathbf{T}_j\right),$$

$$\mathbf{T}_j \sim \mathrm{IW}\left(\alpha, \mathbf{B}_j\right),$$

where a, b, $\boldsymbol{\mu}_{j0}$, α and $\mathbf{B}_j$ are the hyperparameters, which can be set using other studies or preliminary runs [87]. According to the published work of Oh and Raftery [87], the prior distribution of $\mathbf{T}_j$ is set as the same as the covariance matrix of observations $\mathbf{X}$. However, a possible alternative covariance matrix is $\mathbf{T}_j/N_j$. The posterior distribution is expressed as follows:

$$\begin{aligned} f(\mathbf{X}, \sigma^2, \boldsymbol{\mu}, \mathbf{T}, \mathbf{p} \mid \mathbf{D}) \propto (\sigma^2)^{-m/2} \times \exp\left[-\frac{1}{2\sigma^2}\mathrm{SSR} - \sum_{i=1}^{N-1}\sum_{k=i+1}^{N} \log\Phi\left(\frac{\delta_{ik}}{\sigma}\right)\right] \\ \times \prod_{i=1}^{N}\sum_{j=1}^{J}\left\{p^j \left|\mathbf{T}_j\right|^{-1/2} \exp\left[-\frac{\left(\mathbf{X}_i - \boldsymbol{\mu}_j\right)^T \mathbf{T}_j^{-1}\left(\mathbf{X}_i - \boldsymbol{\mu}_j\right)}{2}\right]\right\} \\ \times \left\{(\sigma^2)^{-a-1} \exp\left[-\frac{b}{\sigma^2}\right]\frac{1}{B(\mathbf{p})}\right\} \\ \times \prod_{j=1}^{J}\left\{(p^j)^{1-1}\left|\mathbf{T}_j\right|^{-1/2} \times \exp\left[-\frac{\left(\boldsymbol{\mu}_j - \boldsymbol{\mu}_{j0}\right)^T \mathbf{T}_j^{-1}\left(\boldsymbol{\mu}_j - \boldsymbol{\mu}_{j0}\right)}{2}\right]\right. \\ \left. \times \left|\mathbf{T}_j\right|^{-\frac{\alpha+P+1}{2}} \exp\left[-\frac{1}{2}tr(\mathbf{B}_j\mathbf{T}_j^{-1})\right]\right\}. \end{aligned}$$

As we know, the finite mixture model of $\mathbf{X}_i$ can be simplified by involving a latent variable Z_i,

$$\mathbf{X}_i \mid Z_i = j \sim \phi_j\left(\boldsymbol{\mu}_j, \mathbf{T}_j\right),$$

then the conditional posterior distributions of parameters given the other unknowns can be expressed as follows:

$$f\left(\mathbf{X}_i \mid Z_i = j, \text{others}\right) \propto \exp\left\{-\frac{1}{2}(\mathbf{X}_i - \boldsymbol{\mu}_j)^T\mathbf{T}_j^{-1}\left(\mathbf{X}_i - \boldsymbol{\mu}_j\right) - \frac{\sum_{k\neq i}^{N}\left(\delta_{ik} - d_{ik}\right)^2}{2\sigma^2}\right\} \times \frac{1}{\prod_{k\neq i}^{N}\Phi\left(\frac{\delta_{ik}}{\sigma}\right)} \tag{4.6}$$

$$\text{for } i = 1, \cdots, N \text{ and } j = 1 \cdots, J, \tag{4.7}$$

$$f\left(\sigma^2 \mid \text{others}\right) \propto \left(\sigma^2\right)^{-(m/2+a+1)} \exp\left\{-\frac{\mathrm{SSR}/2 + b}{\sigma^2} - \sum_{i=1}^{N-1}\sum_{k=i+1}^{N}\log\Phi\left(\frac{\delta_{ik}}{\sigma}\right)\right\},$$

$$f\left(p^1, \cdots, p^J \mid \text{others}\right) \sim \text{Dirichlet}\left(N_1 + 1, \cdots, N_J + 1\right),$$

$$f\left(\boldsymbol{\mu}_j \mid \text{others}\right) \sim \phi\left(\frac{N_j\bar{\mathbf{X}}_j + \boldsymbol{\mu}_{j0}}{N_j+1}, \frac{\mathbf{T}_j}{N_j+1}\right),$$

$$f\left(\mathbf{T}_j \mid \text{others}\right) \sim \text{IW}\left(\alpha + N_j/2, \mathbf{B}_j + \mathbf{S}_j/2\right),$$

$$\text{P}\left(Z_i = j \mid \text{others}\right) = \frac{p^j \phi_j\left(\mathbf{X}_i; \boldsymbol{\mu}_j, \mathbf{T}_j\right)}{\sum_{k=1}^{J} p^k \phi_k\left(\mathbf{X}_i; \boldsymbol{\mu}_k, \mathbf{T}_k\right)},$$

where

$$m = \frac{(N-1)\times N}{2},$$

$$\text{SSR} = \sum_{i=1}^{N-1}\sum_{k=i+1}^{N}\left(\delta_{ik} - d_{ik}\right)^2,$$

$$N_j = \sum_{i=1}^{N}\text{I}\left(Z_i = j\right),$$

$$\mathbf{S}_j = \sum_{i=1}^{N}\left(\mathbf{X}_i - \boldsymbol{\mu}_j\right)\left(\mathbf{X}_i - \boldsymbol{\mu}_j\right)^T \text{I}\left(Z_i = j\right).$$

It is easy to tell that the conditional posterior distributions of $\mathbf{X}_i$ and σ^2 do not follow any recognized distributions, so their new values will be generated by using the Metropolis Hastings method, while the rest of parameters will be generated from their conditional posterior distributions by using the Gibbs sampling method.

In the BMBCD approach, we will use a multivariate normal distribution as proposal distribution of $\mathbf{X}$ with the mean equal to values estimated from the previous step. In Section 3.2, we talked about the proposal distribution for $\mathbf{X}_i$ for the one component multivariate normal distribution model. Given the latent position variable Z_i, $\mathbf{X}_i$ in a mixture model will also follow a one component multivariate normal distribution model, so we can modify the proposal distribution of $\mathbf{X}_i$ introduced in Section 3.2 to the mixture model. So we can set the variance of the proposal distribution of $\mathbf{X}_i$ be proportional to $\sigma^2/(N_j - 1)$, where N_j is the number of objects in the j^{th} component.

The distribution of σ^2 does not rely on the mixture model's component parameters, so we can use the same proposal distribution as the one used in one component model, which is a truncated normal distribution with a variance proportional to $\text{IG}\left(m/2 + a, \text{SSR}/2 + b\right)$.

The initial clustering is estimated by using hierarchical clustering, the initial values

of $\mathbf{X}^{(0)}$ and $(\sigma^2)^{(0)}$ are obtained from Bayesian multidimensional scaling introduced in Section 3.2, $(\sigma^2)^{(0)} = SSR^{(0)}/m$. The initial values of $\boldsymbol{\mu}^{(0)} = \left(\boldsymbol{\mu}_1^{(0)}, \cdots, \boldsymbol{\mu}_J^{(0)}\right)$ and $\mathbf{T}^{(0)} = \left(\mathbf{T}_1^{(0)}, \cdots, \mathbf{T}_J^{(0)}\right)$ can be obtained from the initial clustering by using hierarchical clustering [87]. $\boldsymbol{\mu}_0$ are the pre-determined group means. For the hyperparameters in the prior distribution of σ^2, we set a to be a smaller value so that the inverse gamma distribution can cover larger values with higher probabilities. b will be set to be the value that can make the prior mean of σ^2 equal to $\text{SSR}^{(0)}/m$ [87]. The hyperparameters in prior distribution of $\mathbf{T}_j$ are set to be $\alpha = P+4$ and $\mathbf{B}_j = (\alpha - P - 1)\mathbf{S}_j$ for the reasons given below, where P is the number of dimensions and S_j is the covariance matrix of initial cluster j [87]. Further details of the hyperparameters have also been explained in Section 3.2.

The reasons behind the hyperparameters setting in the inverse Wishart distribution are shown below. The expectation of the inverse Wishart distribution is

$$E(\mathbf{T}_j) = \frac{\mathbf{B}_j}{\alpha - P - 1},$$

the variance is

$$\text{var}(\mathbf{T}_j) = \frac{(\alpha - P + 1)B_{ij} + (\alpha - P - 1)B_{ii}B_{jj}}{(\alpha - P)(\alpha - P - 1)^2(\alpha - P - 3)}.$$

If $\mathbf{B}_j = (\alpha - P - 1)\mathbf{S}_j$, then it will set the mean of $\mathbf{T}_j$ to be the initial j^{th} cluster covariance. As any variance must be a non-negative value, so $\alpha = P + 4$ can ensure the variance is non-negative.

Finally, each object will be assigned to the component with the highest probability, i.e. $k_i = \underset{1 \le j \le J}{\text{argmax}} P\left(Z_i = j \mid \mathbf{X}_i\right), \forall i \in 1, \cdots, N.$

4.8 Choice of the Number of Clusters

One of the challenges in cluster analysis is how to decide the number of clusters J, as in most cases the true number of clusters is unknown. So in this section, I will introduce several model-free clustering techniques in determining the number of clusters.

4.8.1 Elbow Plot

The elbow plot [110] provides a visual aid for identifying the optimal number of clusters. The plot is constructed based on the statistic W_J, the total within cluster sum of squares for a number of clusters J against the number of clusters for a range of possible numbers of clusters. The definition of W_J for the J clustering solution is expressed as follows:

$$W_J = \sum_{j=1}^{J} \sum_{i \in C_j} \sum_{i' \in C_j, i' \neq i} d\left(\mathbf{X}_i, \mathbf{X}_i'\right)^2,$$

C_j is the cluster membership which the i^{th} and the i'^{th} objects belonging to, $d\left(\mathbf{X}_i, \mathbf{X}_i'\right)$ is the distance between two objects from the same cluster, W_J equals to the sum of squared distances between objects coming from the same cluster for all clusters.

In an elbow plot, the X-axis indicates the range of the possible number of clusters and the Y-axis indicates the range of W_J. The optimal number of clusters is chosen to be the one where an elbow or bend in the plot suggests the decrease in W_J is levelling off.

The decision about the optimal number of clusters made by an elbow plot is very subjective as there is no reference distribution to judge the correctness the decision.

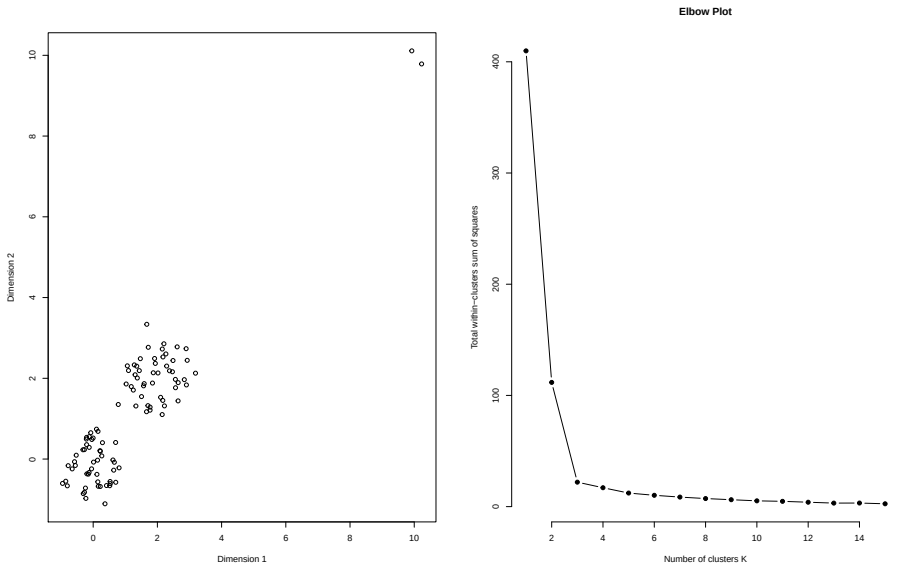
Assume there are 100 data points generated from three different distributions:

$$\mathbf{X} \sim \text{MVN}\left((0,0), \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}\right),$$

$$\mathbf{X} \sim \text{MVN}\left((10,10), \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}\right),$$

$$\mathbf{X} \sim \text{MVN}\left((2,2), \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}\right).$$

There are 49 data points generated from the first and 49 data points generated from the third distribution, another two data points are generated from the second distribution. The data configuration and its corresponding elbow plot are shown in Figure 4.18.



(a) Data (b) Elbow Plot

FIGURE 4.18: Data with Corresponding Elbow Plot

The data configuration in Figure 4.18(a) shows the configurations of these three groups. There are two groups lying on the left bottom of the space, the third group locates on the right top of the space. Figure 4.18(b) is the elbow plot, it has a large drop from $J = 1$ to $J = 3$ and after $J = 3$, the total with-in cluster sum of squares hardly changed, so $J = 3$ will be the choice for the number of clusters.

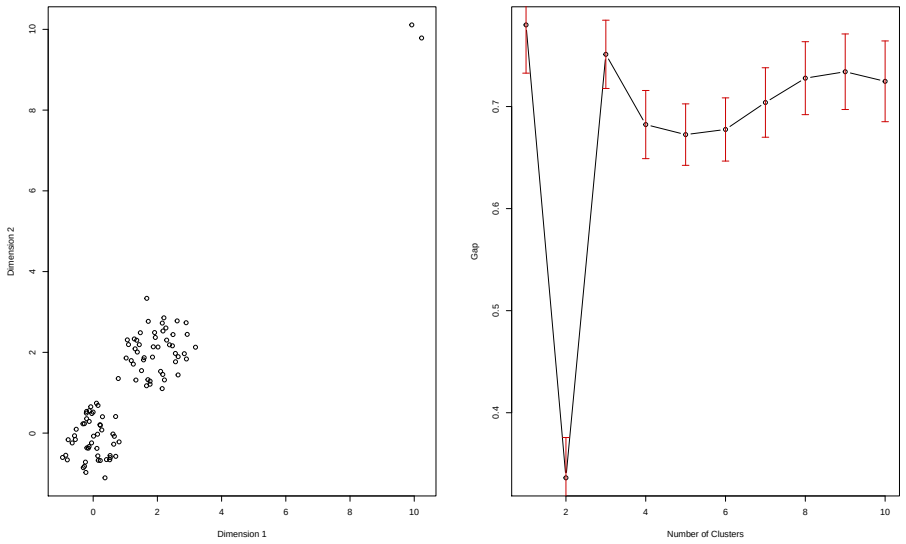
4.8.2 Gap Statistic

The gap statistic [111] is another technique used to identify the optimal number of clusters, which can overcome some of the drawbacks occurring in elbow plots and can obtain a more objective solution to decisions on the number of clusters. The gap statistic standardizes the curve of $\log(W_J)$ by comparing it with a reference distribution, its expression is shown as follows:

$$\text{Gap}(J) = \text{E}(\log(W_J)) - \log(W_J),$$

where $\text{E}(\log(W_J))$ is the expectation taken with respect to the reference distribution. In practice, we use bootstrapping, simulating data from a distribution with the same characteristics as the observed data, but without group structure, to compute the Gap statistic. To obtain an ideal clustering, the optimal number of clusters should be the one

can maximize the gap statistic. However, this will lead to a clustering with an overfitted number of clusters, so we choose the smallest J for which $\text{Gap}(J) > \text{Gap}(J+1)$.



(a) Data (b) Gap Statistic Plot

FIGURE 4.19: Data with a Corresponding Gap Statistic Plot

For the same simulated data as mentioned in Section 4.8.1, the corresponding gap statistic plot is shown in Figure 4.19(b). The smallest J for which $\text{Gap}(J) > \text{Gap}(J+1)$ is $J = 1$, so 1 is the choice for the number of clusters. However, it is a not reasonable value of the number of clustering of this example. The possible explanation for this unreliable decision in the number of clusters can be attributed to the bootstrapping, where the simulated data is not very close to the actual data. So it produced an unreliable reference distribution, resulting in the computed gap statistic being a poor estimate.

4.8.3 Average Silhouette Width (ASW)

The Average Silhouette Width (ASW) was proposed by Leonard Kaufman and Peter J.Rousseeuw [63]. It is a technique to measure how consistent an object is with all other data within the same cluster. For a given dissimilarity matrix $\mathbf{D}(d_{ij})$, the silhouette of

the i^{th} object is defined as :

$$s(i) = \frac{b\,(i) - a\,(i)}{\max\,(a\,(i)\,, b\,(i))} \text{ for } i = 1, \cdots, N,$$

where $a\,(i)$ is the average dissimilarity of the i^{th} object with all other data within the same cluster, $b\,(i)$ is the average dissimilarity between the i^{th} object to the data of its closest adjacent cluster. ASW is the average of all $s(i)$, so the optimal number of clusters is the one with the maximal ASW value.

4.8.4 Calinski and Harabasz Index (CH)

The Calinski and Harabasz (CH) index was proposed by Tadeusz B. Calinski and Joachim Harabasz [22], and is defined as the ratio of between cluster variance (BCSS) and within cluster variance (WCSS). For a given dissimilarity matrix $\mathbf{D}(d_{ij})$, its BCSS and WCSS for J clusters are defined as:

$$\text{BCSS} = \frac{1}{2}\left((J-1)\,\bar{d}^2 + (N-J)\,A_J\right)$$

and

$$\text{WCSS} = \frac{1}{2}\sum_{j=1}^{J}(N_j - 1)\,\bar{d}_j^2,$$

where

$$A_J = \frac{1}{N-J}\sum_{j=1}^{J}(N_j - 1)\left(\bar{d}^2 - \bar{d}_j^2\right)$$

and N is the number of objects, J is the number of clusters, $\bar{d}_j = \dfrac{1}{N_j}\displaystyle\sum_{i\in C_j, i'\in C_j, i'\neq i} d_{ii'}$,

N_j is the number of observations in cluster j, $\bar{d} = \dfrac{1}{N}\displaystyle\sum_{j=1}^{J} N_j\bar{d}_j$, $j = 1, \ldots, J$.

CH can be expressed as follows:

$$\text{CH} = \frac{\text{BCSS}/(J-1)}{\text{WCSS}/(N-J)},$$

For a given set of clusterings with different number of clusters, the optimal number of clusters is chosen to be the one with the maximal CH value.

4.8.5 Pearson version of Hubert's Γ (PH)

Pearson version of Hubert's k_τ [53] compares different number of clusters by measuring the Pearson correlation between the vectorised dissimilarity matrix $\mathbf{D}\,(d_{ij})$ and a binary cluster collocation matrix $\mathbf{B}\,(b_{ij})$ (entries are either 1 or 0). If two different objects i and j are in the same cluster, then $b_{ji} = b_{ij} = 1$, otherwise $b_{ji} = b_{ij} = 0$.

$$k_\tau = \rho\,(d, b) = \frac{\sum_{i<j}\left(d_{ij} - \bar{d}_{ij}\right)\left(b_{ij} - \bar{b}_{ij}\right)}{\sqrt{\sum_{i<j}\left(d_{ij} - \bar{d}_{ij}\right)^2}\sqrt{\sum_{i<j}\left(b_{ij} - \bar{b}_{ij}\right)^2}}. \tag{4.8}$$

So for a range of selected number of clusters, the one with the maximal k_τ will be selected as the optimal number of clusters for the data.

4.8.6 Summary of Choice of the Number of Clusters

In the later chapters, I will use both Elbow plot and PH to select the optimal number of clusters for non-model based clustering. As we can see from Sections 4.8.1 and 4.8.2, both the elbow plot and the gap statistic can visually display the decisions made in the selection of number of clusters. However, the gap statistic is shown to not always be very reliable based on the simple example given in Section 4.8.2 as it also depends on the simulated data. So I will use the elbow plot to decide the number of clusters in non-model based clustering. Among the non-visual number of clusters selection methods, I will choose PH to determine the number of clusters, as it is more stable according to the bootstrap stability selection in the published paper of Spiliopoulou et al. [108].

4.9 Model Comparison

In Section 4.8, I covered the techniques used for making decisions about number of clusters decisions in model-free clustering techniques, e.g. hierarchical clustering, K-means and Chameleon hierarchical clustering. However, if the data can be modeled by statistical clustering models, e.g. model-based clustering with dissimilarities, then the

Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) can also be used in number of clusters decisions.

4.9.1 Akaike Information Criterion (AIC)

The Akaike Information Criterion (AIC) [11] is a criterion to measure the goodness of fit of a model. Theoretically speaking, as the number of parameters involved in a model increases, the model will fit the data better. However, this might lead to an overfitting model. So in order to overcome this difficulty, the number of parameters will be treated as a penalty term in the AIC formula, which can avoid the AIC always decreasing as the number of parameter increases. AIC is defined as follows:

$$\mathrm{AIC} = 2 \cdot K - 2 \cdot \ln \hat{L},$$

where K is the number of independent parameters in the model and $\hat{L}$ is the maximum likelihood of the model. For a given number of candidate models, the model with the minimal AIC will be the optimal one among the selected models.

4.9.2 Bayesian Information Criterion (BIC)

The Bayesian information criterion BIC [37] is another criterion similar to the AIC. It is expressed in the form of

$$\mathrm{BIC} = K \cdot \ln(N) - 2 \cdot \ln \hat{L},$$

where N is the number of objects, K is the number of independent parameters and $\hat{L}$ is the maximum likelihood value. Both the AIC and BIC take the number of parameters in the model into consideration. The difference between the BIC and AIC is the size of penalty, the BIC penalizes the model more heavily than the AIC does as the BIC also involves the data size and $\log(N) > 2$ for most situations.

For model-based clustering, each different number of clusters defines a different model which can be compared using either AIC or BIC. In this thesis, I will use BIC to compare different models to determine the number of clusters as it also takes the sample size into consideration.

4.10 Clustering Comparison Indices

Clustering comparison is a process which is used to compare two different clusterings on the same data. It is particularly helpful in simulations, when the true classification is known, where we can use clustering comparison techniques to compute the consistency between the estimated clusterings and the true classification. In this section, I will explore several clustering comparison techniques.

If only one clustering is available, then for a pair of objects, there are two different possibilities, either both objects are assigned to the same cluster or assigned to different clusters. If there are two clusterings, one clustering with K clusters denoted as $\mathcal{C} = \{\mathcal{C}_1, \cdots, \mathcal{C}_K\}$, the other clustering structure with L clusters denoted as $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_L\}$, then there are four different possibilities for each pair of objects,

1. The pair of objects are assigned to the same cluster in both clusterings.
2. In the first clustering, the pair of objects are assigned to the same cluster, while in the second clustering, the pair of objects are assigned to different clusters.
3. In the first clustering, the pair of objects are assigned to different clusters, while in the second clustering, the pair of objects are assigned to the same cluster.
4. neither of objects are assigned to the same cluster for both clusterings.

The count of four different possibilities is in the table below.

Table 4.2: Contingency Table of Frequencies

		Under Clustering $\mathcal{D}$	
		Pairs of elements assigned to the same cluster	Pairs of elements assigned to different clusters
Under Clustering $\mathcal{C}$	Pairs of elements assigned to the same cluster	a	b
	Pairs of elements assigned to different clusters	c	d

$a + b + c + d = \binom{N}{2}$

4.10.1 Jaccard Index

The Jaccard index [55] is also known as the Jaccard similarity coefficient. It is defined as follows:

$$J(\mathcal{C}, \mathcal{D}) = \frac{a}{a+b+c}.$$

The Jaccard index takes values between 0 and 1. The larger the Jaccard index is, the higher the agreements between the two clusterings.

For example, for a set of data with five objects, there are 10 different combinations (1,2), (1,3), (1,4), (1,5), (2,3), (2,4), (2,5), (3,4), (3,5), (4,5) for those five objects. If the first clustering structure is $\mathcal{C} = \{1, 2, 2, 3, 1\}$, the second clustering structure is $\mathcal{D} = \{1, 1, 2, 2, 1\}$ and we take combination $(1, 2)$ for example. This combination in $\mathcal{C}$ is $\{1, 2\}$ and in $\mathcal{D}$ is $\{1, 1\}$, we can see that the first pair of this combination is different, but the second pair is the same, so c will increase 1. For these data and these two clusterings, the Jaccard index is 0.333.

4.10.2 Rand Index

Another clustering comparison technique is the Rand Index, which was first proposed by William M. Rand [99]. Using Table 4.2, the Rand Index [99] is defined as follows:

$$R = \frac{a+d}{a+b+c+d}.$$

The Rand Index takes values from 0 to 1, if the Rand Index is 1, it indicates the perfect agreement between two clusterings. For the previous example, the Rand index is 0.3.

4.10.3 Adjusted Rand Index

The adjusted Rand index (ARI) is the corrected version of the Rand index. Hubert Lawrence and Arabie Phipps [71] extended the Rand Index by making its expectation under random clustering 0. If the adjusted Rand Index between two clusterings is close to zero, then there is little consistency between these two clusterings; The closer to 1 the

adjusted Rand Index between two clusterings is, the higher the consistency between the two clusterings. The definition of the adjusted Rand Index can be explained by using Table 4.3,

TABLE 4.3: Adjusted Rand Index

	$\mathcal{D}_1$	$\mathcal{D}_2$	$\cdots$	$\mathcal{D}_L$	Marginal Sum
$\mathcal{C}_1$	N_{11}	N_{12}	$\cdots$	N_{1L}	a_1
$\mathcal{C}_2$	N_{21}	N_{22}	$\cdots$	N_{2L}	a_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathcal{C}_K$	N_{K1}	N_{K2}	$\cdots$	N_{KL}	a_K
Marginal Sum	b_1	b_2	$\cdots$	b_L	N

N_{ij} denotes the number of objects in clusters $\mathcal{C}_i$ and $\mathcal{D}_j$. The definition of adjusted Rand Index is expressed as follows (based on Table 4.3):

$$\text{Adjusted Rand Index} = \frac{\sum_{ij}\binom{N_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{N}{2}}.$$

For the same example in Section 4.10.1, the contingency table is shown below,

TABLE 4.4: Contingency Table for Example in Section 4.10.1

	Clustering $\mathcal{D}_1$	Clustering $\mathcal{D}_2$	Marginal Sum
Clustering $\mathcal{C}_1$	2	0	2
Clustering $\mathcal{C}_2$	1	1	2
Clustering $\mathcal{C}_3$	0	1	1
Marginal Sum	3	2	5

so the ARI of this example is 0.091, which is very low. So it indicates that clusterings $\mathcal{C}$ and $\mathcal{D}$ are not very similar.

4.11 Relabeling

Label switching is a problem which can occur in Bayesian inference for mixture models.

If Γ_J denotes the set of permutations of indices 1 to J, then there are $J!$ different permutations, $\boldsymbol{\tau} = (\tau(1), \cdots, \tau(J)) \in \Gamma_J$. If the model parameter array and the mixing probability vector are $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_J)$, $\mathbf{p} = \left(p^1, \cdots, p^J\right)$ respectively and all the objects are independent, then the observed likelihood function with permutation τ_c is expressed as follows:

$$\begin{aligned} L_c\left(\boldsymbol{\theta}, \mathbf{p}; \mathbf{X}\right) &= \prod_{i=1}^{N} P_c\left(\mathbf{X}_i \mid \boldsymbol{\theta}, \mathbf{p}_i\right) \\ &= \prod_{i=1}^{N} \sum_{j=1}^{J} p^{\tau_c(j)} f\left(\mathbf{X}_i \mid \boldsymbol{\theta}_{\tau_c(j)}\right), \end{aligned}$$

This observed likelihood function of the mixture model will be invariant for any permutations of the parameters [89]. If the prior distributions of the parameters are also permutation invariant,

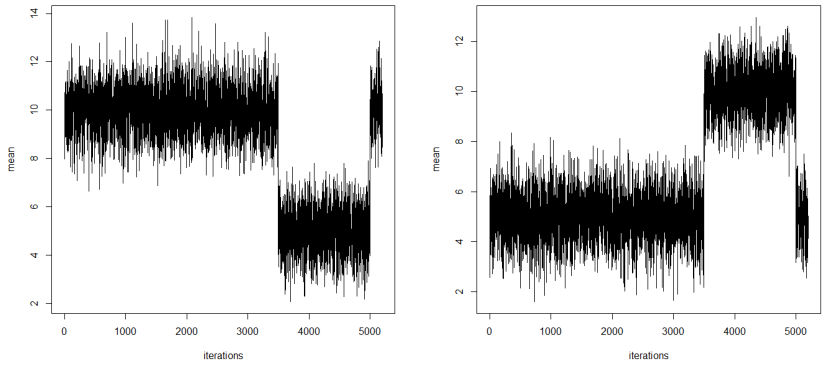
$$\begin{aligned} &P_{\boldsymbol{\tau}_c}\left(\boldsymbol{\theta}\right) = P_{\boldsymbol{\tau}_r}\left(\boldsymbol{\theta}\right) \text{ for any } \tau_r \neq \tau_c, \text{ where} \\ &P_{\boldsymbol{\tau}_c}\left(\boldsymbol{\theta}\right) = P(\boldsymbol{\theta}_{\tau_c(1)}, \cdots, \boldsymbol{\theta}_{\tau_c(J)}), \end{aligned}$$

then the posterior distribution of the mixture model will also hold the same invariant property, $P_{\boldsymbol{\tau}_c}\left(\boldsymbol{\theta}, \mathbf{p} \mid \mathbf{X}_i\right) = P_{\boldsymbol{\tau}_r}\left(\boldsymbol{\theta}, \mathbf{p} \mid \mathbf{X}_i\right)$ for any $\boldsymbol{\tau}_r \neq \boldsymbol{\tau}_c$, via Bayes' Theorem:

$$P\left(\boldsymbol{\theta}, \mathbf{p} \mid \mathbf{X}\right) \propto L\left(\boldsymbol{\theta}, \mathbf{p}; \mathbf{X}\right) P\left(\boldsymbol{\theta}\right).$$

If the simulated output generated from any McMC samplers (i.e. Gibbs sampling, Metropolis Hastings) has converged to an invariant posterior distribution, then the generated parameter values can be switched between $J!$ areas of the posterior distribution [89]. This behaviour is known as the label switching phenomenon.

Take the simulation results shown in Figure 4.20 for example.



(a) McMC Chain for Mean of Cluster 1 (b) McMC Chain for Mean of Cluster 2

FIGURE 4.20: Label Switching Problem 2 Clusters

There are 2 components in this mixture model. It is easy to tell that the component means swapped during the 3500^{th} to 4900^{th} iterations.

One of the solutions to this problem is using the Kullback-Leibler relabelling algorithm [23]. The aim of the Kullback-Leibler relabeling is to minimize the Kullback-Leibler divergence between the average matrix of classification probabilities across all McMC iterations and the classification matrix at each McMC iteration. The average matrix of classification probabilities across all McMC iterations is defined in (4.9). The classification probability at each McMC iteration can be expressed as follows:

$$\omega_i^j = \frac{p^j f\left(\mathbf{X}_i \mid \boldsymbol{\theta}_j\right)}{\sum_{l=1}^{J} p^l f\left(\mathbf{X}_i \mid \boldsymbol{\theta}_l\right)}, \text{ for all } j = 1, \cdots, J.$$

If the initial permutation is $\boldsymbol{\tau}^{(0)}$, e.g. $\boldsymbol{\tau}^{(0)} = \{1, \cdots, J\}$, then the Kullback-Leibler relabeling process [23] at the t^{th} draw, $t = 1, \cdots, T$, follows the following steps.

- Step 1: Calculate

$$q_{ij} = \frac{1}{T} \sum_{t=1}^{T} \omega_i^{\tau^{(t)}(j),(t)}, \tag{4.9}$$

  for $\forall i = 1 \cdots, N$, $j = 1, \cdots, J$.

- Step 2: For $t = 1, \cdots, T$, find a permutation $\boldsymbol{\tau}^{(t)}$ which can minimize

$$h_t^{\tau^{(t)}} = \sum_{i=1}^{N} \sum_{j=1}^{J} \omega_i^{\tau^{(t)}(j),(t)} \log\left(\frac{\omega_i^{\tau^{(t)}(j),(t)}}{q_{ij}}\right).$$

- Step 3: If a reduction can be made to

$$H = \sum_{t=1}^{T} h_t^{\tau^{(t)}} = \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{J} \omega_i^{\tau^{(t)}(j),(t)} \log \left(\frac{\omega_i^{\tau^{(t)}(j),(t)}}{q_{ij}} \right),$$

  then go to Step 1, otherwise, finish.

Chapter 5

Glasgow Housing Market

In this chapter, I will introduce Glasgow housing market background and the data will be used by this thesis. This Glasgow housing market application will be a running example in all methods chapters (Chapters 6, 7, 8 and 9) in order to find out the potential submarkets.

5.1 Housing Market Background

Exploring the substitutability in a housing market is an interesting topic discussed by Grigsby [47] and Rapkin [100]. The idea of substitutability in the housing market is defined as two areal units being substitutable if a property with particular physical characteristics in one areal unit is equally attractive or has the same marginal benefit to a consumer as a similar type of property in the other areal unit. Superior partitioning of the city into submarkets has obvious attractions for real estate agents where it can help them target sales towards customers in a more efficient fashion. An updated, accurate representation of the group structure in the areas of the city in terms of housing demand could also have implications for updating the council tax bands and also for investors looking to diversify their housing portfolios. Also of interest is improving the infrastructures in an area, e.g. schools, types of properties, distances to services, etc, that might drive substitutability between pairs of areas.

Glasgow is the most populous city in Scotland and the third largest city in United

Kingdom with around 600,000 population in total [85]. The area of Glasgow city council is around 175 km^2, including 133 administrative units which are defined as intermediate zones. Intermediate zones are built up from aggregates of data zones and fit within council area boundaries. Each intermediate zone contains at least 2,500 residents [4]. The Glasgow intermediate zones in 2001 are shown in Figure 5.1,

FIGURE 5.1: Map of Glasgow Intermediate Zones

Property deals take a large proportion of the household cost [2] and these properties are either for household living or for investments and renting. The factor which can vary households and investors' decisions in properties purchasing is largely decided by the properties' value over the years, such as whether the properties have kept their values in the past decades or how likely they will keep their values in the future. Glasgow was once one of the most significant cities in UK in terms of manufacturing, which generated a great deal of the city's wealth. Based on the survey done by Select Property Group [1], there are mainly three reasons leading to investment in properties in Glasgow. The first is that Glasgow property prices are ranked as one of the top three fastest accelerating in the UK. An improvement of the city economy and a greater accessibility

to mortgage finance has increased the level of confidence among property buyers and sellers in Glasgow since the beginning of 2014, and this trend has been carried forward since then [1]. Secondly, the buyers' demand is fast outpacing supply, e.g. the desirable areas in 2015, West End and Partick experienced a 50% increase in transactions over the same period one year previous [1]. The third reason is that properties in Glasgow are more affordable than those in London or overseas. For the same period of the year, Glasgow's average property price is more than 50% lower than London [1]. All these reasons make a detailed analysis of Glasgow a matter of urgently.

Many studies have been conducted in grouping areas within housing markets. One of the cluster analyses in housing markets is the Polish housing market case given in [90]. In order to identify the homogeneous features among 16 Polish cities, 13 factors were chosen from each local housing market, then agglomerative hierarchical clustering algorithm with Ward's linkage was used to cluster the cities [90]. However, this type of hierarchical clustering fails to take spatial information into consideration.

In economics, hedonic pricing model is used in grouping housing market. It assumes the housing markets are a set of distinctive submarkets varying from property characteristics and locations, then it utilizes multiple regression to examine the existence of housing submarkets based on the hypothesis that house price can be used to identify and differentiate housing submarkets [9]. The difficulty with applying the hedonic pricing model is how to collect and choose variables which may have potential impact on housing markets. So in this thesis, I am going to explore new techniques to group areas within housing markets.

5.2 Glasgow Housing Market Data

The study region is the city of Glasgow, including 133 administrative units which are defined as intermediate zones. Intermediate zones are built up from aggregates of data zones and fit within council area boundaries. Each intermediate zone contains at least 2,500 residents [4]. As we can see from the Glasgow intermediate zones map in Figure 5.1, there are 133 intermediate zones in Glasgow. The Clyde river divides the Glasgow city into two main parts, which means there are at least two large submarkets in the Glasgow housing market without taking the manual connections, e.g. bridges and ferries, into consideration.

The house data extractions come from information supplied to Registers of Scotland (RoS) with new applications to register a house sale. Registration occurs when a transfer of title takes place, regardless of the amount of money involved. The type of the registered properties only contains residential properties with a price range from £20,000 to £1 million.

The aim of clustering Glasgow housing market is to identify the potential housing submarkets or substitutable intermediate zones. In Chapters 6, 7 and 9, the measurement of substitutability is CPEP, which has been introduced in Section 1.3.1. In this application, the time series data are the annual median house prices from years 1993 to 2010. CPEP is calculated by using these time series data between pairs of areal units, so the substitutable areal units in the same clusters are areas which have a similar changing pattern over these years. The reason for choosing years 1993 to 2010 is because the division of the Glasgow city intermediate zones was updated in 2011, and for the same city area, the total number of intermediate zones increased from 133 to 136. So in order to keep the analysis consistent, I chose the data from 1993 to 2010 to calculate CPEP (data source: `http://statistics.gov.scot/data/house-sales-prices`). The sample size required to construct a regression model depends on the aim of modeling. A large sample size usually refers to more than 30 observations [7]. If we apply an OLS regression-based approach, then arguably 4 observations may suffice [29]. On the other hand, the sample size required increases with the number of parameters to be estimated, and the amount of noise in the data. So based on these suggestions, in order to achieve a reliable regression model, we require more years before modeling the updated intermediate zone data. The descriptive statistics of CPEP are listed in Table 5.1. From Table 5.1 we can see that both the mean and median of CPEP are fairly high,

TABLE 5.1: Summary of Descriptive Statistics of CPEP

	CPEP
min	0.472
max	0.995
median	0.855
mean	0.832
std.dev	0.134

which indicate that the tendency of the housing prices in the 133 intermediate zones have strong similarities. The application in Chapters 6, 7 and 9 will use CPEP data to find similar changing patterns or tendencies between pairs of areal units. The areal units in this study are the 133 intermediate zones in Glasgow city making up $\binom{133}{2}$ pairs. The distribution of all pairs of CPEP is shown below,

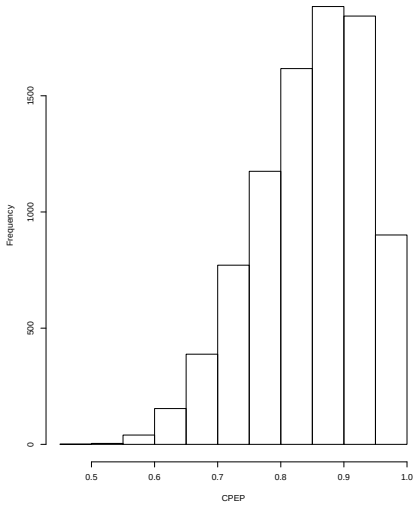


FIGURE 5.2: The Distribution of CPEP in Glasgow Housing Market

As we can see from Figure 5.2, CPEP lies between 0 and 1 (by construction), with a minimal value 0.472 and a maximal value 0.995. However, CPEP data are paired data, so instead of displaying CPEP clustering in Glasgow housing market, I display the clustering of the time series regression's slopes in Figure 5.4. For each areal unit, I regress the area based on its log change in housing prices over 18 years (1993 to 2010) as the regression slope is also can be used to describe the changing pattern of the areal housing prices.

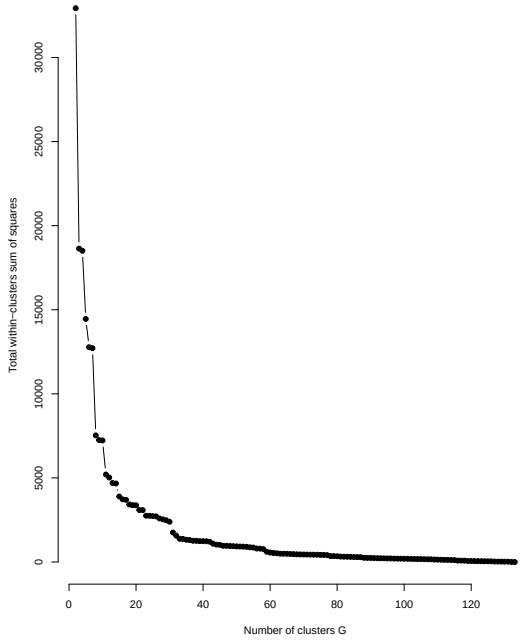


FIGURE 5.3: Elbow Plot of Housing Price Tendency Based on Glasgow City Intermediate Zones

The total within-clusters sum of squares hardly changed after $G = 30$, so the number of clusters for the changing tendency without any geographical connection information is set to be 30.

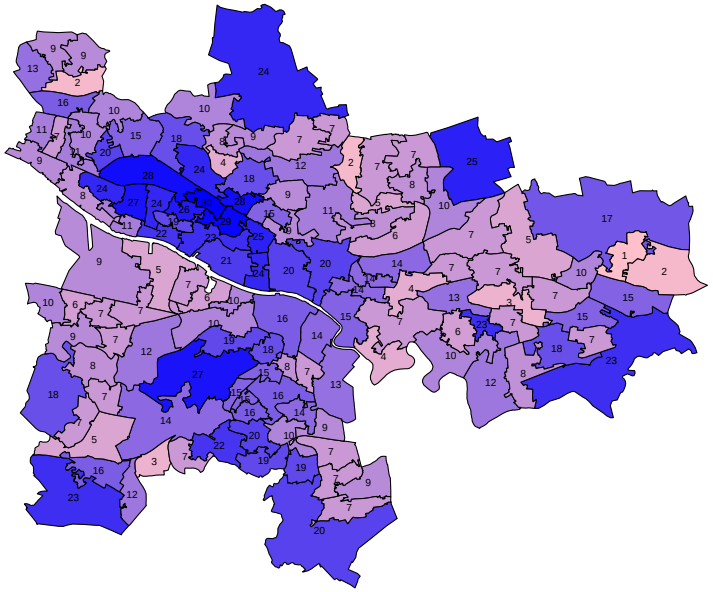


FIGURE 5.4: Tendency Clustering ($G = 30$) Based on Glasgow City Intermediate Zones

All the regression slopes are grouped into 30 clusters without any geographical connection information according to the elbow plot in Figure 5.3. In Figure 5.4, the darker the colour is, the steeper the changing patter will be. So we can see that the west end areas have more darker areas than the other areas.

The application aim in Chapter 8 is to find the areas in the Glasgow housing market where areas in the same cluster have similar household incomes, housing prices and number of over 60 income support claims. The other aim in this chapter is to show the applicability of the spatially constrained finite mixture models proposed in Chapter 8 in a multivariate space. In this application, I will use a three-dimensional data set, with one dimension representing the median house prices in all intermediate zones in 2010, the second dimension representing the weekly median total net household income

in all intermediate zones in 2010 (Source: http://statistics.gov.scot) and the third dimension representing the number of over 60 years old claims of income support (IS) in 2010. Gross income covers income from all sources (wages, salaries, pensions, benefits, rent, interest, maintenance) before the deduction of tax and national insurance contributions [8]. IS is only available to people who are not required to be available for work such as carers, sick and disabled people and lone parents. The number of pensioner claimants will not equal the number of claimants aged 60 or over, as it also considers the age of the partner [8]. The first two dimensions are measured in GBP (£). The descriptive statistics of these three variables are listed in Table 5.2.

TABLE 5.2: Summary of Descriptive Statistics of Housing Prices, Income and Number of Claims

	houseprice	householdincome	low.income.over.60
nbr.val	133.00	133.00	133.00
min	60500.00	255.00	30.00
max	245000.00	559.00	640.00
median	115000.00	344.00	225.00
mean	121685.23	361.71	249.47
std.dev	40718.45	75.04	124.88

From Table 5.2 we can see that the range of number of claims is smaller than the range of housing prices and household incomes, for which maximum number of claims is around 20 times greater than the minimum. It is also easy to tell that both housing prices and household incomes do not contain extreme values, i.e. very low or high housing prices and household incomes.

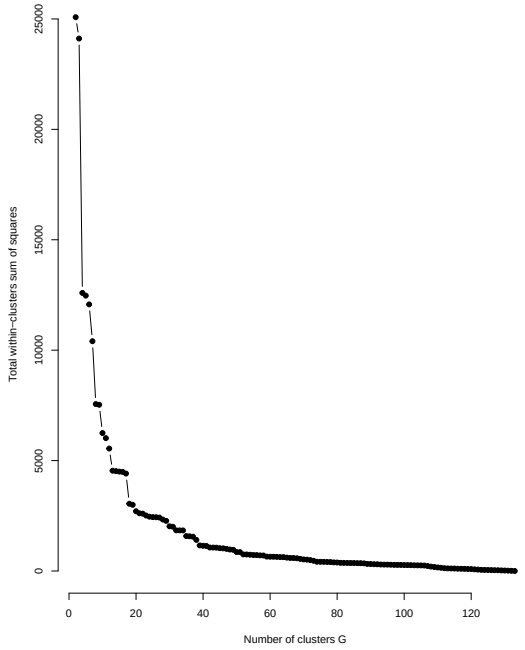


FIGURE 5.5: Elbow Plot of 2010 Housing Prices Data Based on Glasgow City Intermediate Zones

The total within-clusters sum of squares did not change very significantly after $G = 18$, so the number of clusters for housing prices without any geographical connection information is set to be 18.

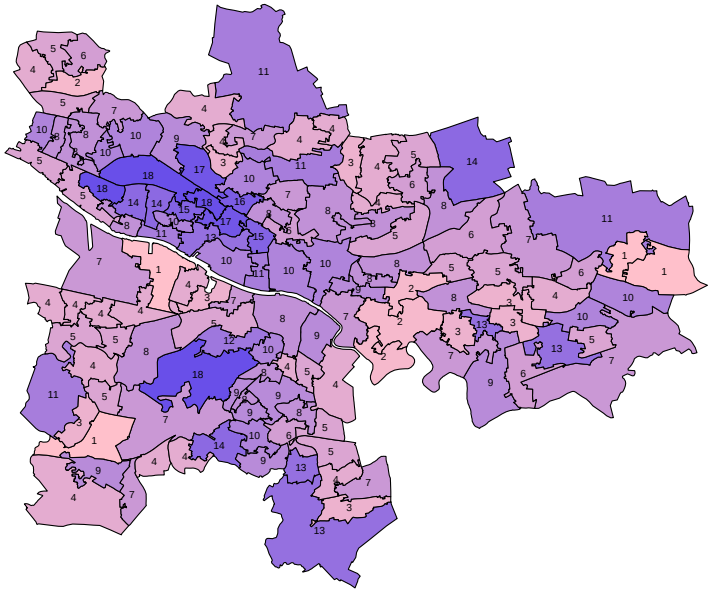


FIGURE 5.6: 2010 Housing Price Clustering ($G = 18$) Based on Glasgow City Intermediate Zones

Figure 5.6 shows the clustering of Glasgow housing price based on the raw housing price data without any geographical connection information. The range of the raw housing price data is from 60,500 to 245,000 and the number of cluster is set to be 18 according to the elbow plot in Figure 5.5. The darker the area is, the higher the housing price will be. For example, the areal units with higher housing prices areas (dark blue areas in Figure 5.6) in 2010 were mostly lying in the west end.

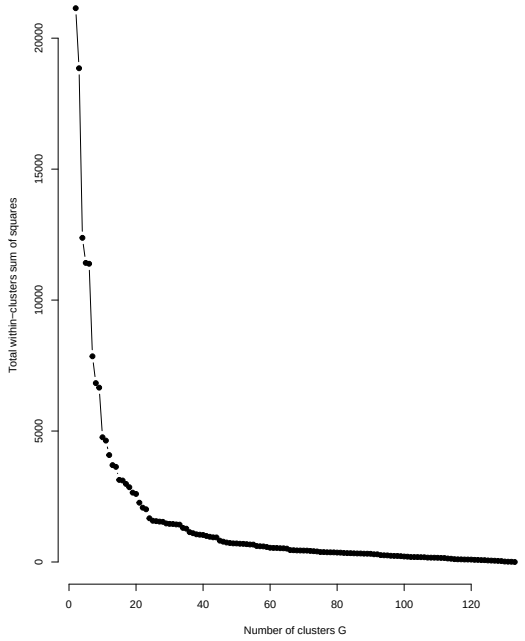


FIGURE 5.7: Elbow Plot of 2010 Household Income Based on Glasgow City Intermediate Zones

The total within-clusters sum of squares hardly changed after $G = 22$, so the number of clusters for household incomes without any geographical connection information is set to be 22.

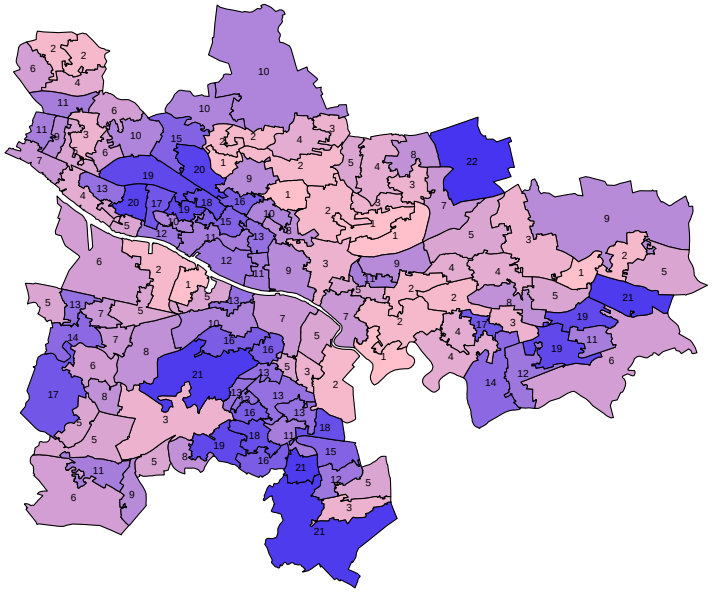


FIGURE 5.8: 2010 Household Income Clustering ($G = 22$) Based on Glasgow City Intermediate Zones

Figure 5.8 shows the clustering of Glasgow household income based on the raw household income data without any geographical connection information. The range of weekly household income is from 255 to 559 and the number of clusters is set to be 22 according to the elbow plot in Figure 5.7. The dark the area is, the higher the income will be. For example, the areal unit with the highest median household income cluster (dark blue areas in Figure 5.8) in 2010 was Robroyston and Millerston.

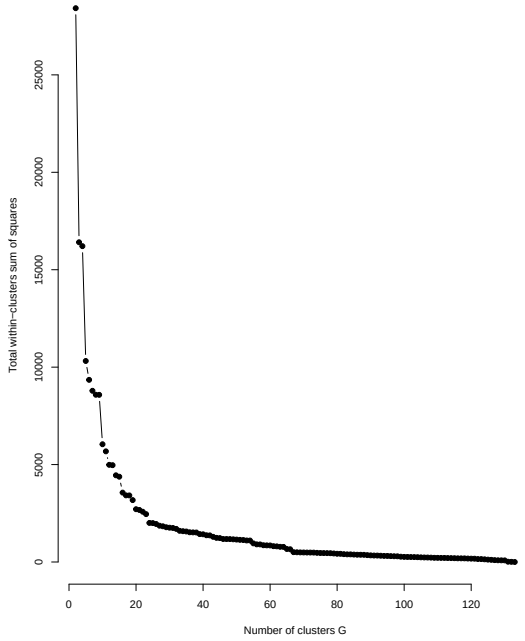


FIGURE 5.9: Elbow Plot of 2010 Over 60 Income Support Claims Based on Glasgow City Intermediate Zones

The total within-clusters sum of squares hardly changed after $G = 21$, so the number of clusters for IS without any geographical connection information is set to be 21.

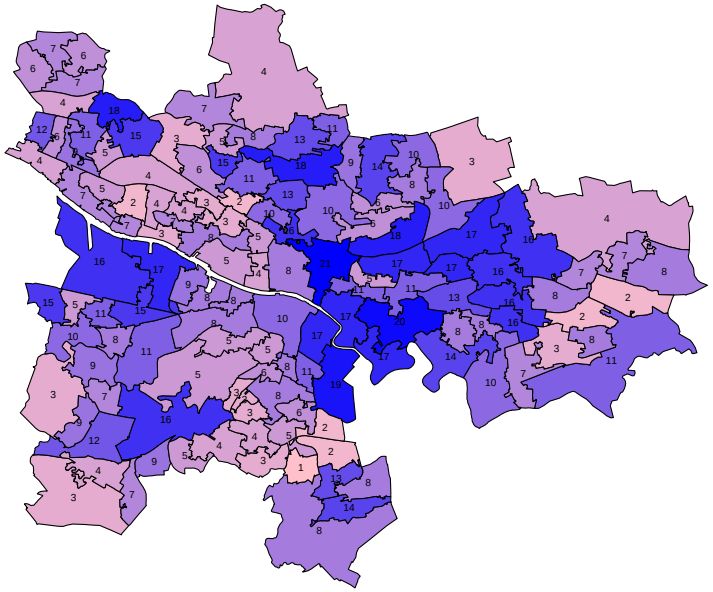


FIGURE 5.10: Clustering of 2010 Over 60 Years Old Income Support Claims Based on Glasgow City Intermediate Zones

Figure 5.10 shows the clustering of Glasgow based on the over 60 years old income support claims data without any geographical connection information. The range of the number of over 60 income support claims is from 30 to 640 and the number of clusters is set to be 21 according to the elbow plot in Figure 5.9. The lighter the colour is, the less the number of income support claims will be. The clustering in Figure 5.10 is very different from figures in Figure 5.4, 5.6 and 5.8, the areas with higher number of income support claims were from east end and the opposed areas with higher increasing trends, housing prices and household incomes.

Chapter 6

Spatial Agglomerative Hierarchical Clustering

The traditional existing clustering techniques introduced in Chapter 4 can only group data based on their characteristics, usually without taking spatial information into account. So in this chapter, I will describe one type of spatial clustering algorithm, the spatial hierarchical clustering algorithm [13], where the areal units grouped together should not only share similar characteristics, but be geographically contiguous as well. This novel idea will later be applied to the other spatial clustering algorithms in Chapters 7, 8 and 9.

6.1 Spatial Agglomerative Hierarchical Clustering Algorithm

The spatial hierarchical clustering algorithm was proposed by Craig Anderson [13]. It combines the areal geographical information with dissimilarity data to group the areal units. In spatial hierarchical clustering, two clusters with minimum distance or dissimilarity may not be merged together unless these two clusters are geographically connected [13].

The novelty of this method compared to traditional hierarchical clustering is that two clusters can only be joined together if they share at least one common border (or their groups share some common borders). Suppose we have two non-singleton clusters, both

with more than one areal unit in them. If these two clusters are also the clusters with minimum dissimilarity at the current iteration, and one of the objects in one cluster is geographically connected to at least one of the objects from the other cluster, then these two clusters will be merged together.

Constructing a hierarchy of spatial clusterings for areal units usually requires us to know both dissimilarity data and spatial data. Spatial data is represented by the neighbourhood matrix $\mathbf{W}(w_{ij})$, which is a symmetric binary matrix, where entries are either 1 or 0. $w_{ij} = 1$ indicates two areal units i and j are spatially connected, otherwise, they are disconnected and $w_{ij} = 0$. For a given dissimilarity matrix $\mathbf{D}(d_{ij})_{(N\times N)}$ and its corresponding neighbourhood matrix $\mathbf{W}(w_{ij})_{(N\times N)}$, the procedure of spatial hierarchical clustering can be described as follows:

1. In the beginning, we choose one of the linkage methods from Section 4.4.1. All areal units ($\boldsymbol{\mathcal{A}} = \{\mathcal{A}_1, \cdots, \mathcal{A}_N\}$) are in their own clusters ($\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \cdots, \mathcal{C}_N\}$).
2. Merging Step
   (a) Merge the least dissimilar of the geographically connected pairs of areal units or clusters $\mathcal{C}_i, \mathcal{C}_j$ into one $\mathcal{C}_{\text{new}} = \mathcal{C}_i \cup \mathcal{C}_j$. For clusters with more than one object, the distances with other clusters will be calculated by using the chosen linkage.
3. Updating Step
   (a) Update the distance matrix, the distances between pairs of clusters will be measured by the chosen linkage.
   (b) Update the neighbourhood matrix between $\mathcal{C}_{new}$ and all other adjacent clusters $\mathcal{C}_k, k \in$ others, $w_{\mathcal{C}_{new},\mathcal{C}_k} = \max\{w_{\mathcal{C}_k,\mathcal{C}_i}, w_{\mathcal{C}_k,\mathcal{C}_j}, \forall k \in \text{others} \neq i, j\}$.
4. Repeat Merging and Updating steps until no more areal units or clusters can be merged, the number of clusters c will change from N to m in this procedure (m is the number of the isolated spatial disconnected components, for which range is from $[1, N]$). In the first step, $c = N$; in the j^{th} step, $c = N - j + 1$ ($N - j + 1 \geq m$); at the last step, $c = m$.

Unlike the classical hierarchical clustering, the last step of the spatial hierarchical clustering might contain more than one cluster. This occurs when final clusters are all geographically disconnected, so they cannot be merged any further.

6.2 Glasgow Housing Market Clustering Results

For the Glasgow housing market application, I will use average linkage to define the between clusters distances in the spatial hierarchical clustering because average linkage does not require coordinate data and is less cluster size dependent. The clustering selected by both elbow plot and PH (introduced in Section 4.8.5) will be discussed.

6.2.1 Spatial Hierarchical Clustering Results of CPEP Data

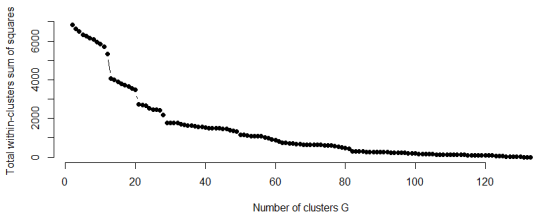


FIGURE 6.1: Elbow Plot of Glasgow City Intermediate Zones

The elbow plot for all possible number of cluster for the CPEP data is shown in Figure 6.1, we can see that the total within cluster sum of squares is dropping sharply until the number of clusters is around 30, after this, the line levels off. In addition, when Pearson version of Hubert's Γ (PH), introduced in Section 4.8.5, is used to select the number of clusters, the maximal PH is 0.280 when $G = 30$. This 30 submarkets clustering of Glasgow housing market is shown in Figure 6.2.

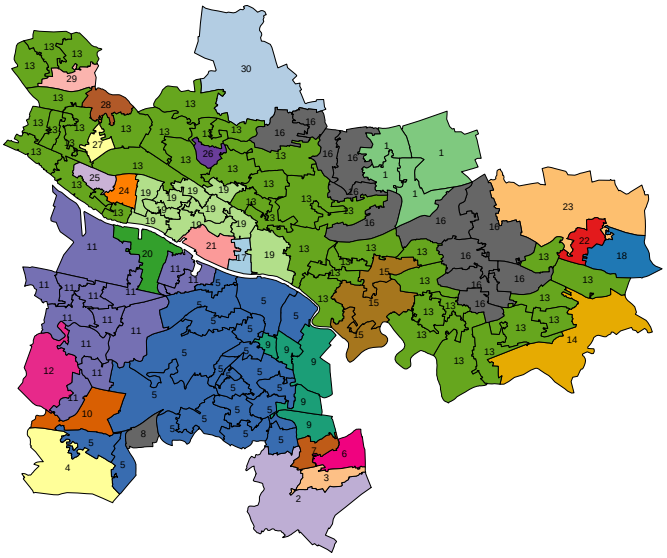


FIGURE 6.2: Housing Market Clustering ($G = 30$) Based on Glasgow City Intermediate Zones

From Figure 6.2, we can see that the north of Clyde river is mainly dominated by cluster 13, where areas belonging to this cluster changed similarly over the observed years. The West End of Glasgow is further divided into clusters 17, 19 and 21. This is reasonable given the current housing market, e.g. there are many newly built properties in Anderston which are likely to have different changing prices.

6.2.2 Spatial Hierarchical Clustering Results of 3-Dimensional Data

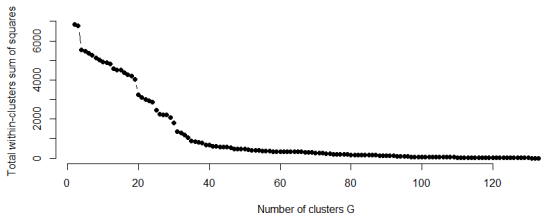


FIGURE 6.3: Elbow Plot of 2010 Data (Household Income, Housing Price and Over 60 Income Support Claimts) Based on Glasgow City Intermediate Zones

The elbow plot of the 2010 data (household income, housing price and over 60 income support claims) is shown in Figure 6.3, we can see that the total within cluster sum of squares is dropping sharply until the number of clusters is around 35, after this, the line levels off. In addition, when PH is used to select the number of clusters, the maximal PH is 0.340 when $G = 35$. So for 2010 data (household income, housing price and over 60 income support claims), the number of clusters will be set to 35.

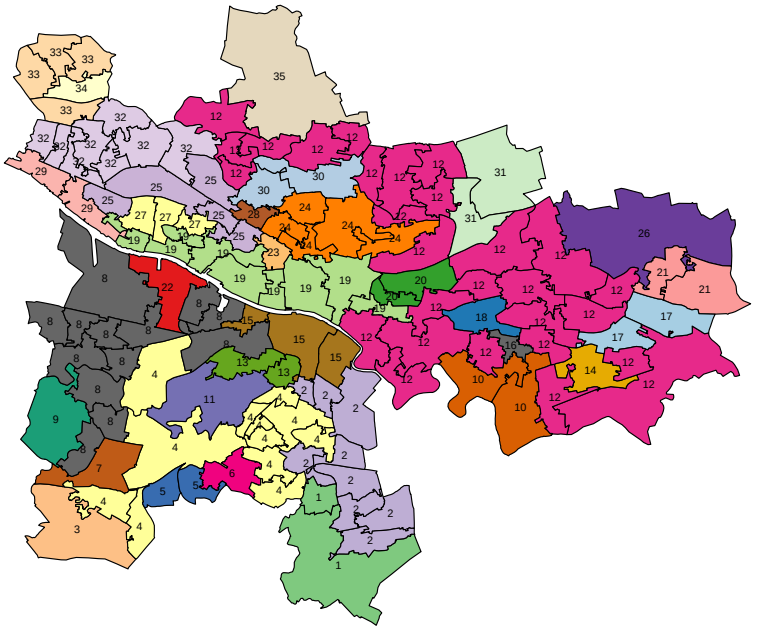


FIGURE 6.4: Clustering ($G = 35$) of 2010 Data (Household Income, Housing Price and Over 60 Income Support Claimts) Based on Glasgow City Intermediate Zones

Compared with the clustering in Figure 6.2, Glasgow has been grouped into more clusters. The three dimensional data show more information about residential financial information rather than providing information for investors or estate agents about the housing prices tendency in different areas. Take cluster 19 for example, although areas in cluster 19 have different housing prices they have similar level of household incomes and number of income support claims relative to the housing prices, i.e. housing price/household income$\approx$ 430, housing price/claims$\approx$ 710 in both Hillhead and Anderston.

6.3 Summary of Spatial Hierarchical Clustering

In this chapter, I describe one type of spatial clustering algorithm, the spatial hierarchical clustering algorithm [13], where the areal units grouped together should not only share similar characteristics, but be geographically contiguous as well. The clustering in Figure 6.2 shows the clusters of changing patterns of Glasgow housing market over years 1993 to 2010. So the prices of areal units in the same cluster were changing in a similar pattern. This means the clustering shows more similarities in changing patterns of housing prices rather than their absolute values. For property investors, more investment risk will be avoided if they invest into different clusters, such as purchasing some properties from cluster 5 and getting some properties from 16. In addition, investors can also keep an eye on some of the properties in order to predict the future prices of the other properties from the same cluster. The changing pattern clustering will be further explored in Chapters 7 and 9 by using a similar idea to the spatial hierarchical clustering. The clustering in Figure 6.4 of the three dimensional data shows more information about the residents financial information in different areas. The clustering results of these three dimensional data will be further explored in Chapter 8.

Chapter 7

Chameleon Spatial Hierarchical Clustering

In Chapter 4, I introduced Chameleon hierarchical clustering algorithm for grouping non-spatial objects. However, the classical Chameleon hierarchical clustering fails to meet the need of clustering spatial data, where two spatially disconnected areal units cannot be grouped in to the same cluster. In Chapter 6, I introduced spatial hierarchical clustering which can deal with the issue caused by geographical disconnection. Based on this idea, in this chapter, I will propose a new modified clustering technique, Chameleon spatial hierarchical clustering. One of the motivations in applying Chameleon spatial hierarchical clustering is the input data. The CPEP data used in this thesis is one type of similarity data, while most of the available clustering techniques use either dissimilarity or coordinate data as the input data. The transformation between similarity data and dissimilarity or coordinate data could have some influence on the clustering results, and this influence can be reduced if the cluster technique can use similarity data directly as the input data. The goal of this chapter is to develop the Chameleon hierarchical clustering which uses similarity data into Chameleon spatial hierarchical clustering and then apply this clustering technique to the Glasgow housing market to find groups of substitutable areas based on CPEP.

7.1 Chameleon Spatial Hierarchical Clustering Algorithm

Chameleon spatial hierarchical clustering mainly consists of three stages: K-NN graph stage, graph partitioning stage and merging stage. The procedure of the Chameleon spatial hierarchical clustering is shown next. Differences between Chameleon spatial hierarchical clustering and regular Chameleon hierarchical clustering are highlighted in italics.

Algorithm:

1. K-Nearest Neighbours Graph Stage (see Section 7.1.1 for greater detail)
   (a) *Construct a fully complete graph. Remove edges between pairs of vertices if they are geographically disconnected.*
   (b) Construct a K-nearest neighbours graph by identifying and connecting up to the most similar K neighbours to each object, *depending on how many edges are left connected to each object after step 1 (a).*
   Note: Each vertex might have less than K edges.
2. Graph Partitioning Stage
   (a) Set a value for M, which is the number of components that will be formed at the end of the the M-partitioning stage and also chose a value for $C > 1$.
   (b) Coarsening Stage (see Section 4.5.2.2 for greater detail)
      i. Visit each object in a random order and merge the object with an unmerged and connected object which has the maximal similarity. Skip the objects which have been merged in the previous steps.

         Detail:
         A. In each iteration, each object can merge with at most one object.
         B. The similarities (weights) between the newly formed cluster and its adjacent clusters are the sum similarity between the cluster vertices with those adjacent clusters [60].
      ii. As the aim is to form an M-partition graph, so the number of components left should be a function of M. Repeat step 2 (b) i until the remaining number of components is no more than $M \times C$. The formed graph at the last coarsening stage is called the coarsest graph within the components as vertices.

(c) M-partitioning Stage (see Section 7.1.3 for greater detail)

i. The initial bisection is formed based on breadth-first algorithm. It starts with one randomly selected vertex, then grows a region around it until the two parts of the graph have roughly equal size.

ii. Repeat step 2 (c) i until *up to* M components formed and these components subject to the partitioning constraint (see Section 4.5.1.2 for further detail), this process is called the M-way partition algorithm [60].

iii. If M is not large, steps 2 (c) i and 2 (c) ii will be repeated by starting from a different vertex many times until finding all the possible M-partitions, as different starting points may create different M-partitions [58] (We usually only create a small number of partitions for the reason given in Section 4.5.2.3). The optimal M-partitioning will be the one with the minimal total between-components similarities at the coarsest graph stage and this partition will be used in the following stages.

Notation: *1). The geographically isolated components will not merge with any of the components. If the size of the geographically disconnected components is less than* $|V|/(C \times M)$*, then they are allowed to violate the partitioning constraint. 2). M must be greater than the number of the geographical disconnected components. The number of the small sized components for which sizes are less than* $|V|/(C \times M)$ *are excluded from these M components. 3). Bisecting is performed based on the components meeting the partitioning constraint only.*

(d) Uncoarsening Stage (see Section 4.5.2.4 for greater detail)

i. The partition of the coarser graph will be projected back to the next level finer graph successively by splitting the multi-nodes into next level individual nodes.

ii. Apply the hyperedge refinement (HER) algorithm (see Section 4.5.2.4 for further detail) to the bordering vertices. The movement of a bordering vertex depends on the gain (see (4.3) for further detail) and is subject to the partitioning constraint (see Section 4.5.1.2 for further detail).

iii. Repeat steps 2 (d) i and 2 (d) ii until the graph is projected back to the finest graph (the original graph).

3. Merging Stage (see Section 4.5.2.5 for greater detail)

(a) Set a value α_0 in (4.4). In bipartitioning, if the number of areal units in each component is an even number, we set $\alpha = 0.5$, otherwise, we set $\alpha = 0.55$ in order to get an approximately equal bipartitioning components.

(b) Calculate the pairs of clusters' RIRC value by using (4.4), merge the pair of clusters with the maximal RIRC value, then update the RIRC between the newly formed components and its adjacent components.

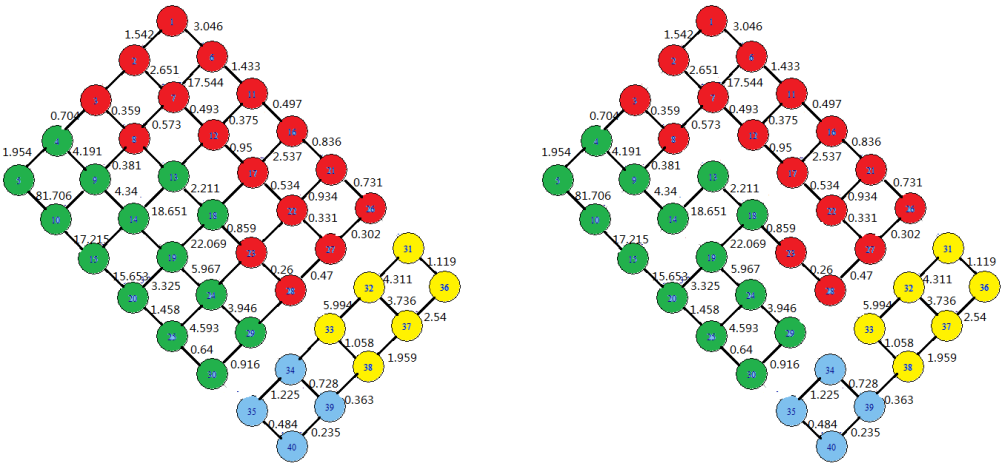
(c) Repeat step 3 (b) until no more merges occur.

(d) *RIRC cannot be calculated for any of the geographically isolated singleton components.*

(e) Use Pearson's version of Hubert's k_τ (PH) (introduced in Section 4.8.5) to choose the optimal number of clusters.

7.1.1 K-Nearest Neighbours Graph Stage

Suppose we have 40 areal units generated from two different cluster distributions, $\mathrm{N}(0, 0.5)$ and $\mathrm{N}(20, 0.5)$ and the true classification differentiated by colours is shown in Figure 7.13. More specifically, all red and blue areal units are generated from $\mathrm{N}(0, 0.5)$, all green and yellow areal units are generated from $\mathrm{N}(20, 0.5)$. Although all red and blue areal units are generated from the same distribution, as they are geographically disconnected clusters, so they are labeled as different clusters, so are all green and yellow areal units. Figures on the edges denote similarities between pairs of areal units. In this example, we set $K = 2$ and assume all similarities between pairs of areal units are measurable, Figure 7.1(a) only displays the geographical connections between areas and Figure 7.1(b) displays the the K-NN graph connections based on the geographical connections. In Glasgow housing market data, the K nearest neighbours will be determined by CPEP data.



(a) Geographical Connections

(b) K-NN connections Based on Geographical Connections, $K = 2$

FIGURE 7.1: Geographical and K-NN Connections

7.1.2 Coarsening Stage

The coarsening stage in Chameleon spatial hierarchical clustering is the same as the coarsening stage in regular Chameleon hierarchical clustering.

In this example, we set $M = 4$ and $C = 2$, so we will repeat coarsening until no more than eight components left. Figure 7.2 shows the clustering at the first coarsening stage. We visit all 40 vertices in a random order, 27 16 6 40 11 21 25 4 29 5 35 28 24 12 1 18 23 33 36 8 32 14 37 30 17 7 10 22 9 31 15 38 19 39 20 2 26 13 34 3. The first visited vertex is 27, its most similar and connected vertex is 28, so we group these two areal units into a new component and name it as 27. The interconnection between areal units 27 and 28 is ignored, the updated similarities between it and its adjacent clusters are the sum of similarities between areal units 27 and 28 to their adjacent clusters. In particular, when we visit the areal unit 23, its most similar and connected adjacent areal unit is 24, which has merged with areal unit 25, then we check the next most similar and connected adjacent areal unit, but all of its adjacent areal units have merged with other areal units, so we will leave the areal unit 23 as an isolated singleton component.

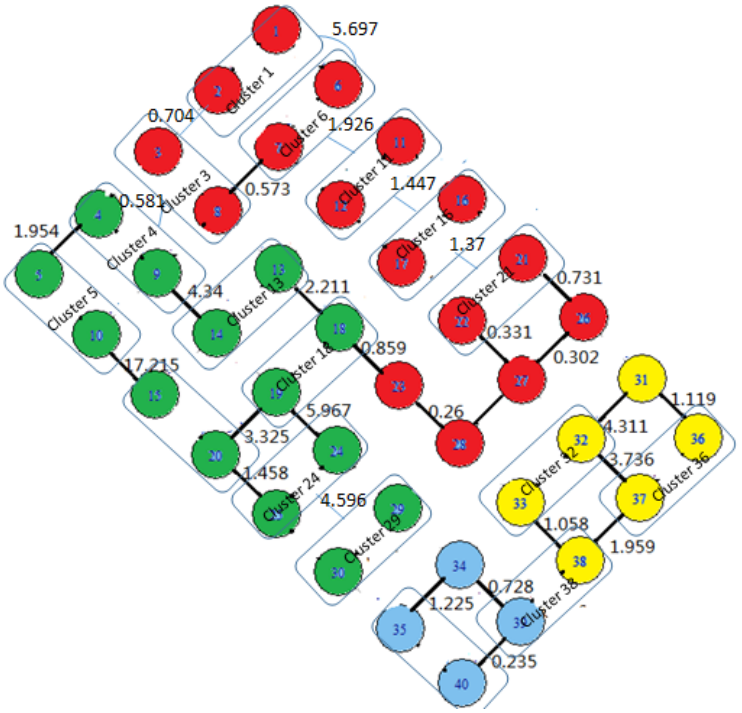


FIGURE 7.2: First Coarsened Graph

We then randomly visit the newly formed 22 components in an order of 15 6 23 4 32 35 5 11 18 1 31 13 38 3 24 21 27 26 29 34 36 16. The coarser graph is shown in Figure 7.3.

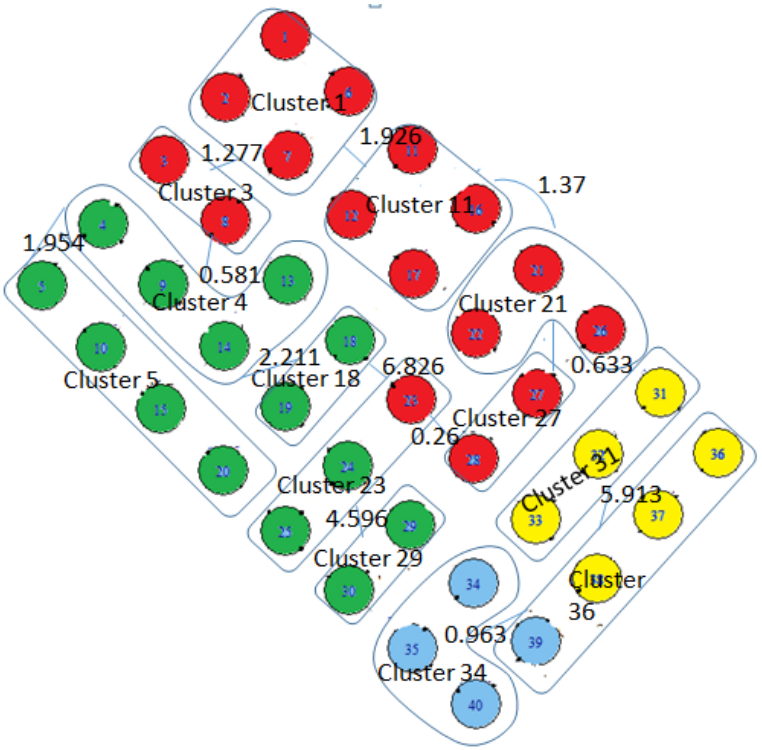


FIGURE 7.3: Second Coarsened Graph

We then randomly visit the newly formed 13 components in an order of 1 27 5 29 11 36 18 21 3 23 31 34 4. The coarser graph is shown in Figure 7.4, we stop coarsening as at this stage as the remaining number of clusters is no more than $M \times C = 8$ components.

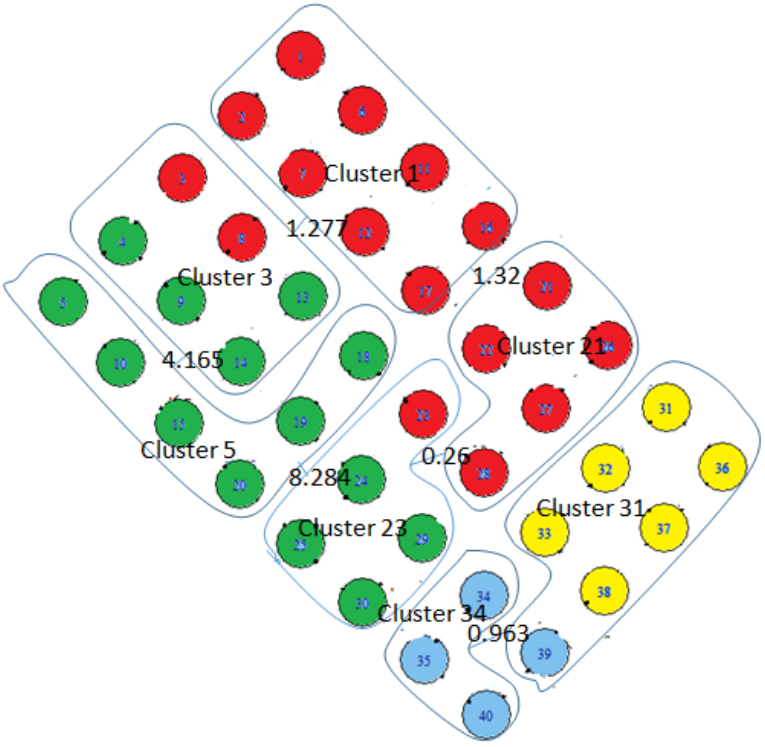


FIGURE 7.4: The Coarsest Graph

The coarsest graph can be simplified to that given in Figure 7.5.

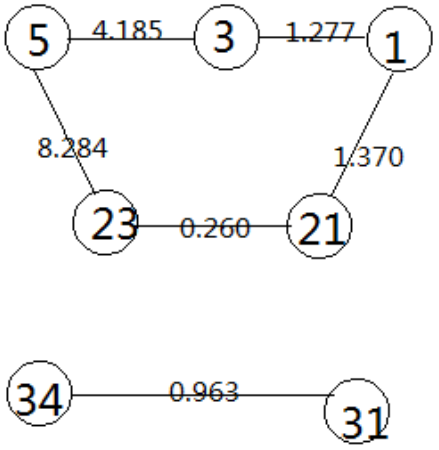


FIGURE 7.5: The Simplified Coarsest Graph of Figure 7.4

7.1.3 M-Partitioning Stage

The M-partitioning stage in Chameleon spatial hierarchical clustering is similar to the M-partitioning stage in regular Chameleon hierarchical clustering. One of the differences is that the geographically isolated components for which sizes are less than $|V|/(C \times M)$ are allowed to violate the partitioning constraint. In addition, M must be greater than the number of the geographical disconnected components.

First, we roughly divide the coarsest graph (G_0) into two different components (A and B) with roughly equal number of vertices if possible. More specifically, suppose there are P_0 components left in G_0. The initial bisected partition is formed by applying the breadth-first algorithm. We randomly select one component as the starting component, then we connect all the connected components to this selected component before moving to the next level neighbours and group these connected components into one larger component (component A), the rest of components are assigned to the other larger component (component B). This procedure will be repeated until up to M components formed if possible while each component subjects to the partitioning constraint. The number of the small sized components for which sizes are less than $|V|/(C \times M)$ are excluded from these M components. Otherwise, there might be the scenarios for which the number of components is M, but the sizes of some components are still greater than $(C \times |V|)/M$.

In Chameleon hierarchical clustering, as K in K-NN graph is always a positive integer, it is unlikely to get a coarsest graph with isolated singleton components. However, in Chameleon spatial hierarchical clustering, edges are formed as the K-NN connections based on the geographical connections, so there might be some geographically isolated components formed at the coarsest graph, which might violate the partitioning constraint (the number of objects is less than $|V|/(C \times M)$). So in the Chameleon spatial hierarchical clustering algorithm, we allow the geographically isolated components (for which size is less than $|V|/(C \times M)$) to violate the partitioning constraint and no more bisections needs to be done on these components.

The partitioning constraint in this example is $\frac{40}{4\times2} = 5 < |V| \leq \frac{40\times2}{4} = 20$. As we can see from the coarsest graph in Figure 7.5, the geographical information automatically created two components A and B, so at the next stage, we need to further partition this partition into four components ($M = 4$). By comparing all different starting points, the optimal starting point to obtain the minimum total between component similarities start from component 5. The most similar and connected components to it are 3 and 23,

so we stop with component $A_1 = \{3, 5, 23\}$ and component $A_2 = \{1, 21\}$. For the next level partition, we only conduct it on component A as if we further divide component B, the newly formed components will violate the partitioning constraint. At the next level partition, the minimum total between component similarities can be obtained by removing component 3 from component A_1. Now we got four components, they are component 1= $\{5, 23\}$, component 2= $\{3\}$ and component 3= $\{1, 21\}$, component 4= $\{31, 34\}$, all of these components satisfy the partitioning constraint, so this is the optimal initial partition of the coarsest graph by comparing different starting points.

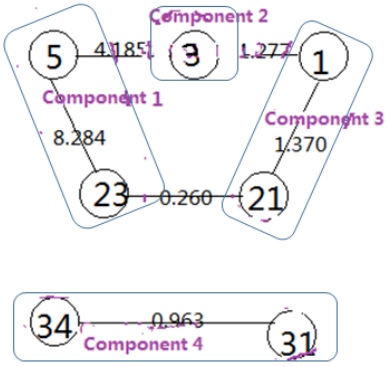


FIGURE 7.6: Initial M Partitioning

7.1.4 Uncoarsening and Refinement Stage

As mentioned in Section 4.5.2.4, the minimal total between component similarities in the coarsest graph might not be the minimal total between component similarities in the finer graphs. So at this stage, we use the hyperedge refinement (HER) introduced in Section 4.5.2.4 to potentially move the bordering vertices to their geographically connected adjacent components in order to reduce the total between component similarities. At this stage, there are no positive gains, so there is no need to make any changes. The next step is to project graph in Figure 7.6 back to its next level finer graph by splitting the multi-nodes into next level individual nodes, which is shown in Figure 7.7.

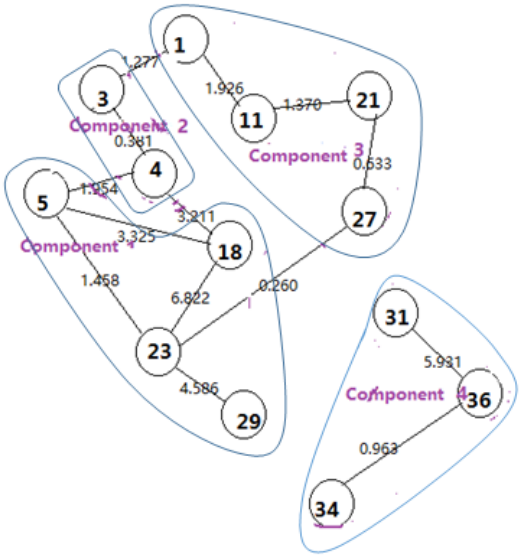


FIGURE 7.7: Finer Graph 1

All geographically connected bordering components are 1, 3, 4, 5, 18, 23, 27. Only components 3 and 4 have positive gains, but if we move them to the other components, it will violate the partitioning constraint, so we do not make any changes at this stage. The next finer graph is shown below,

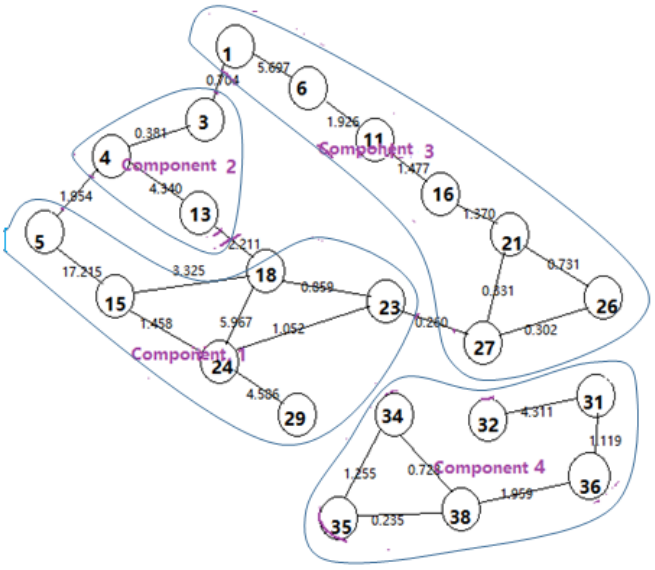


FIGURE 7.8: Finer Graph 2

All geographically connected bordering components are 1, 3, 4, 5, 13, 18, 23, 27. Only component 3 has positive gain, but if we move it to the other partitions, it will violate

the partitioning constraint, so we do not move any components at this stage. The finest graph at this stage is shown in Figure 7.9.

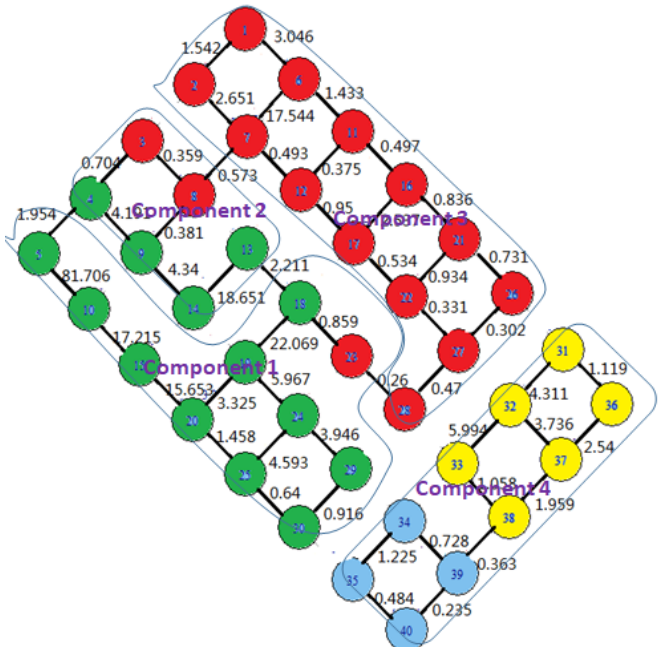


FIGURE 7.9: Finest Graph

7.1.5 Merging Stage

The merging stage in Chameleon spatial hierarchical clustering is as same as the merging stage in regular Chameleon hierarchical clustering.

In this example, we use default values of $\alpha_0 = 1$ for giving the same weight to both the relative connectivity and the relative closeness. In bipartitioning, if the number of areal units in each component is an even number, we set $\alpha = 0.5$, otherwise, we set $\alpha = 0.55$, for which can closely equally bipartition components. The internal connectivity of component 1 is 8.568, the internal connectivity of component 2 is 4.340, the internal connectivity of component 3 is 1.447, the internal connectivity of component 4 is 3.017. The interconnectivity between partitions 1 and 2 is 4.165, the interconnectivity between components 1 and 3 is 0.26, the interconnectivity between components 2 and 3 is 1.280. The internal closeness of component 1 is 12.868, the internal closeness of component 2 is 5.584, the internal closeness of component 3 is 2.071, the internal closeness of component 4 is 1.896. The intercloseness between components 1 and 2 is 2.083, the intercloseness between components 1 and 3 is 0.26, the intercloseness between components 2 and 3 is

0.640. The largest RIRC value is $RIRC_{1,2}$, so we merge components 1 and 2 into one and the updated graph is shown in Figure 7.10.

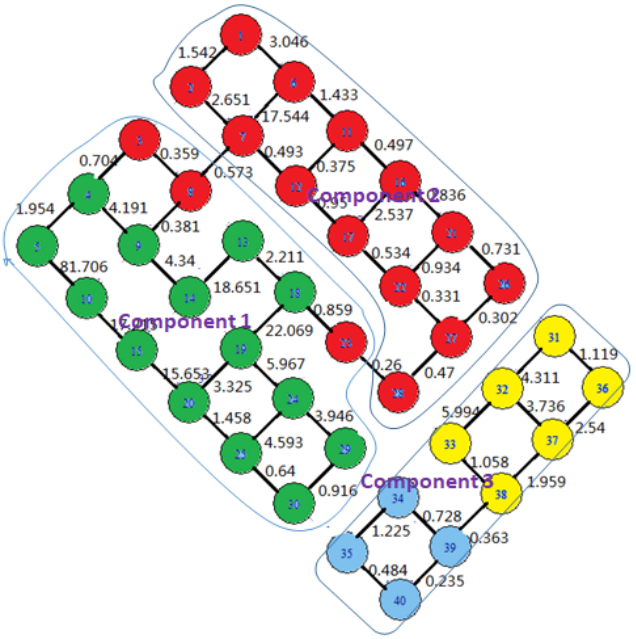


FIGURE 7.10: Merged Graph

The internal connectivity of the new component 1 is 9.132, the internal closeness of the new component 1 is 9.816, then $RIRC_{1,3} = 0.022$ and we merge components 1 and 3 into one. After obtaining the hierarchy of clusterings, clustering 1 is the one shown in Figure 7.9, for which PH is 0.315; Clustering 2 is the one shown in Figure 7.10, for which PH is 0.341; Clustering 3 is the one only has two components, one of them contains both components 1 and 2 in Figure 7.10, the other component only contains the component 3 in Figure 7.10, for which PH is -0.015. By comparing all these three PH values, the clustering with the maximal PH is the one shown in Figure 7.10.

Although the expected number of clusters is 4, the final simulated clustering only has three components, but it still captured most of the characteristics of the data, the adjusted Rand index comparing to the true classification is 0.688, which is relatively high.

7.2 Parameter Setting

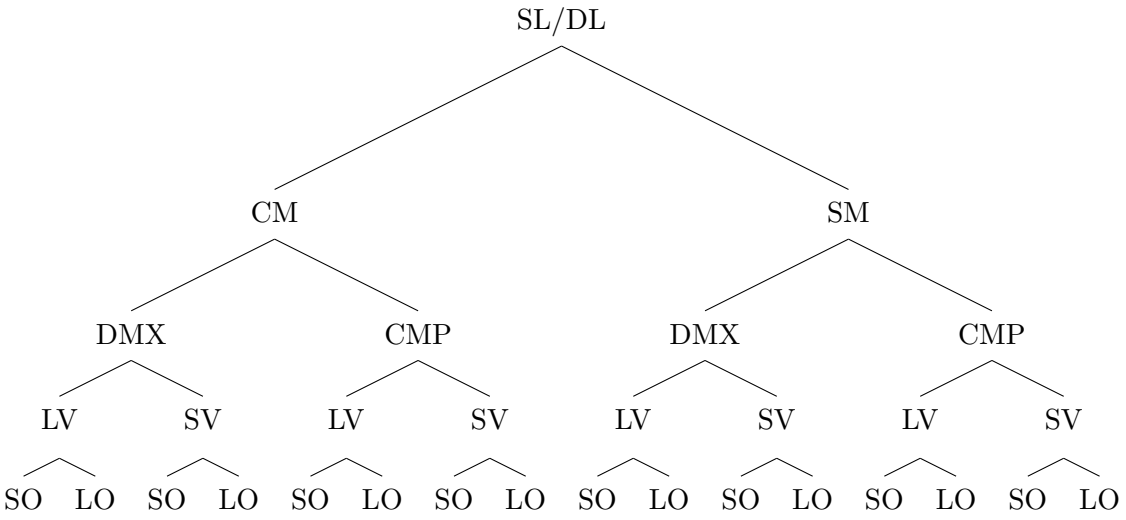
To cluster a dataset using Chameleon spatial hierarchical clustering, there are a number of parameters that need to be set. According to the qualitative comparisons of K in K-nearest neighbor graph stage, the reduced size (no more than $M \times C$) at the coarsening stage and the choice of scheme for combining relative inter-connectivity and relative closeness (α) in the published paper of Karypis et al. [59] (in their experiments, the data size N are from 1,000 to 10,000, $K = \{5, 10, 15, 20\}$, $M \times C = \{20\% \times N, 30\% \times N, 40\% \times N\}$ and $\alpha = \{0.1, 0.5, 1, 1.5, 2, 2.5, 3\}$), they concluded that Chameleon hierarchical clustering is not very sensitive to the above choice of parameters, and it was able to discover the correct clusters for all these combinations for K, $M \times C$ and α. More simulations about the parameters will be given in Sections 7.3.3 to 7.3.6. In order to make this technique more usable, I will give some default values in Section 7.3.9 in order to help users.

7.3 Simulations

In this section, I will use simulations generated from some different scenarios to compare the performance of Chameleon spatial hierarchical clustering and spatial hierarchical clustering. In order to make this technique applicable for most of users, I will set some default parameters and the

7.3.1 Simulation Framework-Factorial Design

In this section, the factorial design will involve several factors, the levels of the distribution means, different mixing proportions, different levels of the variances, geographical locations and different numbers of outliers (32 different combinations in total). For all simulations there are 100 observations generated.



Notation:

- SL: Sparse distribution of areas, where areal units generated from the same distribution are scattered in different parts of the geographical space, i.e. Locations in Figures 7.12 and 7.14.

- DL: Condensed distribution of areas, where most areal units generated from the same distribution are geographically connected, i.e. Locations in Figures 7.11 and 7.13.

- CM: Close levels of means for all distributions, i.e. the mean of one distribution is 0, the mean of the other distribution is 2 (details gives in Section 7.3.2).

- SM: Separate levels of means for all distributions, i.e. the mean of one distribution is 0, the mean of the other distribution is 20 (details gives in Section 7.3.2).

- DMX: Different mixing proportions for all distributions, i.e. approximately 1:9 for areas generated from two distributions, locations in Figures 7.13 and 7.14.

- CMP: Similar mixing proportions for all distributions, i.e. approximately 1:1 for areas generated from two distributions, locations in Figures 7.11 and 7.12.

- LV: Large variances (compared to the distribution mean) for all distributions, i.e. 0.6 or 8 (details gives in Section 7.3.2).

- SV: Small variances (compared to the distribution mean) for all distributions, i.e. 0.25 (details gives in Section 7.3.2).

- LO: Large Number of Outliers, i.e. 6.

- SO: Small Number of Outliers, i.e. 2.

7.3.2 Simulation Scenarios

Further details about the distribution sets mentioned in the tree diagram above are listed below.

1. Multivariate Distributions set 1
   - $\text{MVN}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0.25 & 0\\0 & 0.25\end{pmatrix}\right)$
   - $\text{MVN}\left(\begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}0.25 & 0\\0 & 0.25\end{pmatrix}\right)$

   In multivariate distributions set 1, data are generated from two close levels of means with small variances.

2. Multivariate Distributions set 2
   - $\text{MVN}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0.6 & 0\\0 & 0.6\end{pmatrix}\right)$
   - $\text{MVN}\left(\begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}0.6 & 0\\0 & 0.6\end{pmatrix}\right)$

   In multivariate distributions set 2, data are generated from two close levels of means with large variances.

3. Multivariate Distributions set 3
   - $\text{MVN}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0.25 & 0\\0 & 0.25\end{pmatrix}\right)$

- MVN $\left(\begin{pmatrix}20\\20\end{pmatrix}, \begin{pmatrix}0.25 & 0\\0 & 0.25\end{pmatrix}\right)$

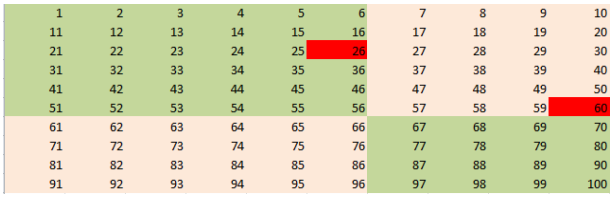
  In multivariate distributions set 3, data are generated from two separate levels of means with small variances.

4. Multivariate Distributions set 4
   - MVN $\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}8 & 0\\0 & 8\end{pmatrix}\right)$
   - MVN $\left(\begin{pmatrix}20\\20\end{pmatrix}, \begin{pmatrix}8 & 0\\0 & 8\end{pmatrix}\right)$

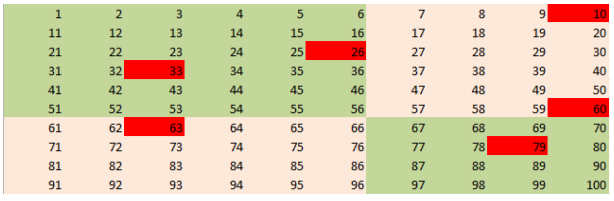
     In multivariate distributions set 4, data are generated from two separate levels of means with large variances.

So in total there are 32 different combinations, for both SL/DL have 16 combinations, each of these combinations has 100 points with 100 replications simulated and all these combinations' simulation results will be examined using the following methods,

- Chameleon spatial hierarchical clustering
- Spatial hierarchical clustering

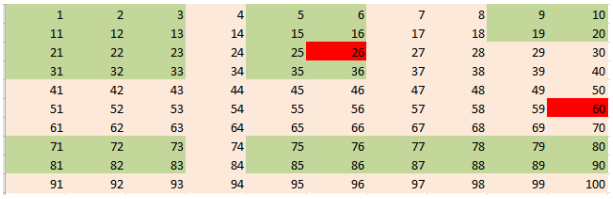


(a) Small Number of Outliers

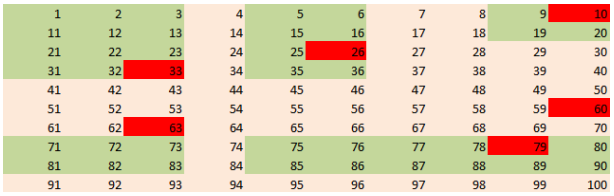


(b) Large Number of Outliers

FIGURE 7.11: Geographical Information with a Condensed Distribution Locations and Similar Mixture Proportions



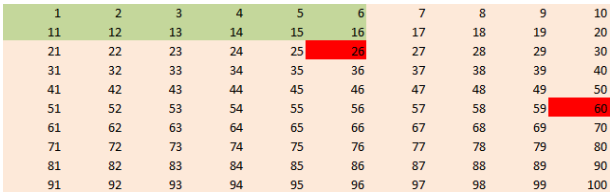
(a) Small Number of Outliers



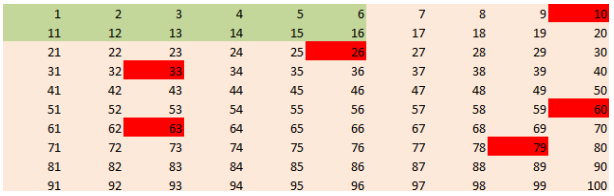
(b) Large Number of Outliers

FIGURE 7.12: Geographical Information with a Sparse Distribution Locations and Similar Mixture Proportions

The location shown in Figure 7.11 is a condensed distribution locations, most areal units generated from the same distribution are geographically connected, while the location shown in Figure 7.12 is a sparse distribution locations, where areal units generated from the same distribution are scattered in different parts of the geographical space. However, in both locations, except for the noise points, all the different distributions have roughly equal numbers of members.

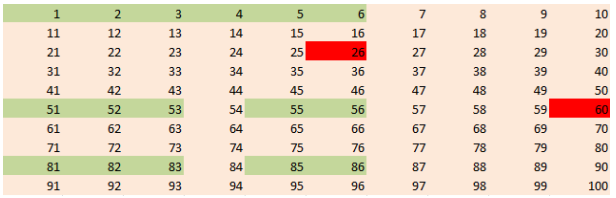


(a) Small Number of Outliers

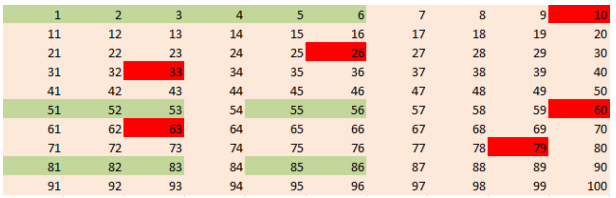


(b) Large Number of Outliers

FIGURE 7.13: Geographical Information with a Condensed Distribution Locations and Different Mixture Proportions



(a) Small Number of Outliers



(b) Large Number of Outliers

FIGURE 7.14: Geographical Information with a Sparse Distribution Locations and Different Mixture Proportions

The location shown in Figure 7.13 is a condensed distribution locations, most areal units generated from the same distribution are geographically connected. However, proportions of data generated from those two distributions are very different, the space is mainly dominated by one of the distributions. The location shown in Figure 7.14 is a sparse distribution locations, where areal units generated from the same distribution are scattered in different parts of the geographical space and the proportions of data generated from the main two distributions are very different.

Data generated from different distributions are labeled in different colours, i.e all yellow objects are generated from the first distribution of Distributions set 1, all green objects are generated from the second distribution of Distributions set 1. In particular, objects in the same colour but geographically disconnected will be labeled as different clusters. The cells in red are the outliers, they are generated from MVN $\left(\begin{pmatrix}10\\10\end{pmatrix}, \begin{pmatrix}0.25 & 0\\0 & 0.25\end{pmatrix}\right)$.

In order to compare the performance of different clustering algorithms, a quantified measurement is required. Adjusted Rand Index (ARI), which has been introduced in Section 4.10.3 will be used. ARI is used to measure the agreement between different clusterings on the same data. In this example, it measures the agreement between the estimated clusterings from different spatial clustering techniques and the corresponding true classification. The higher the ARI is, the more similar the two are and the better the performance of the corresponding method.

7.3.3 Decision about K in K-NN Graph Stage

There are several tuning parameters (pre-determined parameters) influencing the simulated clustering results. In this section, I will select a range of K and investigate the relationship between K and ARI in different scenarios. In our example, we suppose all 100 areal units are generated from simulation set-up 1, the true classifications for four locations are displayed in Figures 7.11, 7.12, 7.13 and 7.14. M = True No. Clusters, $C = 2$, $\alpha_0 = 1$ and $\alpha = 0.5$ (for even number of objects or $\alpha = 0.55$ for odd number of objects in Merging stage). The estimated clustering for each simulation is the one with the maximal PH in the estimated clustering hierarchy. The number of simulations for each scenario is repeated 100 times. All the simulation results are shown in Table 7.1:

TABLE 7.1: Results for Different K for Data of Type Given in Figure 7.11(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$K = 2$	0.659 (0.018)	4.35 (0.23)	6 (4+2)
$K = 3$	0.870 (0.028)	3.80 (0.41)	6 (4+2)
$K = 5$	0.898 (0.025)	4.15 (0.27)	6 (4+2)

The simulation results for 7.11(b), 7.12, 7.13 and 7.14 are shown in Appendix A.3

If K is too small, then the K-NN graph connections based on the spatial connections will be small, neither the average numbers of clusters nor average ARI show good results. When K is too large, the K-NN graph stage will increase computation time. So it is better to choose K neither too large nor too small, e.g. $K = 3$.

7.3.4 Decision about M in M-Partitioning Stage

In paper Karypis et al. [59], they only compared the performance of $M \times C$ over a sequence of values. But in this section and Section 7.3.5, M and C will be compared separately as they have different meanings, where M is the number of components expected to be obtained at M-partitioning stage and the imbalance tolerance C will determine the cluster size. If M is too small, then the groups may not be identifiable, so it is reasonable to set a relatively large value for M, e.g. 10.

7.3.5 Decision about C in M-Partitioning Stage

In this section, I will investigate the impacts of the imbalance tolerance C on different scenarios. For the 100 areal units generated from simulation set-up 1, the true classifications for four types of locations are displayed in Figures 7.11, 7.12, 7.13 and 7.14. M = True No. Clusters, $K = 3$, $\alpha_0 = 1$ and $\alpha = 0.5$ (for even number of objects or $\alpha = 0.55$ for odd number of objects in Merging stage). The estimated clustering for each simulation is the one with the maximal PH in the estimated clustering hierarchy. The number of simulations for each scenario is repeated 100 times. All the simulation results are shown in Table 7.2:

TABLE 7.2: Results for Different C for Data of Type Given in Figure 7.11(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$C = 2$	0.870 (0.028)	3.80 (0.41)	6 (4+2)
$C = 4$	0.802 (0.036)	3.47 (0.33)	6 (4+2)

The simulation results for 7.11(b), 7.12, 7.13 and 7.14 are shown in Appendix A.4

C is an imbalance tolerance parameter. If C is large, it will allow a larger difference between different components and will be more likely to identify the noise points; otherwise, it will allow a smaller difference between different components and will be less likely to identify the noise points. In Table 7.2, the small C is set to be 2, the large C is set to be 4. For the lower bound of the component size, except for data of the types given in Figures 7.11(b), 7.12(b) and 7.14(b) are lower than 1, data of types given in the other locations are still greater than 1, which means they are less likely to identify the noise points given a larger C. From the simulation results we can see that, for the sparse distribution locations, given $C = 4$, only the data of type given in Figure 7.14(b) got a higher ARI. For the condense distribution locations, only when C is small, they got higher ARIs. So I recommend $C = 2$ for the data type of locations is unknown.

7.3.6 Decision about α_0 in Merging Stage

α_0 is a parameter to decide the weight of relative closeness and relative connectivity.

For the 100 areal units are generated from simulation set-up 1, the true classifications for four locations are displayed in Figure 7.11, 7.12, 7.13 and 7.14. M = True No. Clusters, $K = 3$, $C = 2$ ($C = 4$ for data of type given in Figure 7.14(b)) and $\alpha = 0.5$ (for even number of objects or $\alpha = 0.55$ for odd number of objects in Merging stage). The estimated clustering for each simulation is the one with the maximal PH in the estimated clustering hierarchy. The number of simulations for each scenario is repeated 100 times. All the simulation results are shown in Table 7.3:

TABLE 7.3: Results for Different α_0 for Data of Type Given in Figure 7.11(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$\alpha_0 = 0.5$	0.866 (0.020)	3.76 (0.36)	6 (4+2)
$\alpha_0 = 1$	0.870 (0.028)	3.80 (0.41)	6 (4+2)
$\alpha_0 = 2$	0.872 (0.024)	3.08 (0.33)	6 (4+2)

The simulation results for 7.11(b), 7.12, 7.13 and 7.14 are shown in Appendix A.5

α_0 has little impact on the estimated clustering results which is consistent with the conclusion from Karypis et al. [59] for the possible reason that, in general, the relative connectivity and the relative closeness are both less than 1 and have a similar value range (as the total similarities connecting different components are usually smaller than the internal similarities) [59]. So with less prior information, we can set the default value of α_0 being 1.

7.3.7 Decision about α in Merging Stage

In this section, I will investigate the impacts of α in RIRC on different scenarios. α is the parameter used to bipartition the cluster into two roughly equal sized subclusters. This bisection is required in calculating the linkage RIRC. In the equation of RIRC 4.4, it is easy to see that it is the product of relative interconnectivity and relative closeness. Both of these statistics are calculated from internal connectivity and internal closeness which are required to divide a cluster into two roughly equal number of sub clusters, more details are in Section 4.5.1.5. But sometimes, the number of objects in a cluster is not a even number, in this cases, Karypis et al. [59] suggested that α can be set to be 0.55 in order to approximately bipartition into two equal sized subclusters. If clusters are divided unevenly, then it will be more likely to achieve bisected parts having one

larger part with higher weights and a higher number of vertices, while the other part has smaller weights and fewer number of vertices. So the weights of one of the subclusters could be affected more by just one or two vertices. If the cluster is divided more closer to the equal sized subclusters, then the weight difference between two subclusters will be averaged by considering more vertices. In this context, the housing prices in some intermediate zones, e.g. Hillhead, are much higher or lower than the others. In order to avoid the RIRC linkage being affected by only one or a few intermediate zones, so the roughly equal sized bisection is preferred in this thesis.

7.3.8 Speed Comparison with Spatial Hierarchical Clustering

In this section, I will assess the running speed of Chameleon spatial hierarchical clustering. A set of simulations based on different parameter combinations were run using simulated data generated from set-up 1. Each parameter combination was run 100 times, the average time is recorded in seconds.

TABLE 7.4: Speed Comparison in Chameleon Spatial Hierarchical Clustering with Different Parameters for Data of Type Given in Figure 7.11(a)

	$K=2$	$K=3$	$K=5$
$M=8$			
$C=2$	0.36	0.37	0.50
$C=4$	0.18	0.17	0.16
$C=8$	0.22	0.15	0.19
$M=16$			
$C=2$	0.25	0.23	0.27
$C=4$	0.18	0.17	0.16
$C=8$	0.29	0.15	0.32
$M=32$			
$C=2$	0.31	0.51	0.35
$C=4$	0.33	0.31	0.31
$C=8$	0.31	0.30	0.32

$\alpha_0 = 1$ in the above simulations

Spatial hierarchical clustering was also run using the simulated data generated from set-up 1 a hundred times, for which average running time was 1.87 seconds. In Table 7.4,

by comparing with the running speed of spatial hierarchical clustering, it is easy to tell that for all different combinations, Chameleon hierarchical clustering has a much faster running speed, at least five times faster than spatial hierarchical clustering. Generally speaking, when C is smaller (e.g. $C = 2$), it will take longer to run compared with the other values, but this will still not affect the running speed too much.

7.3.9 Default Parameters

To cluster a data set using Chameleon spatial hierarchical clustering, there are a number of parameters that need to be set. In this section, I will suggest default values for all parameters in order to help users to make decisions. The default α_0 will be set as 1 according to the simulation results given in Section 7.3.6 and the published paper Karypis et al. [59]; M will be set by users depending on how many clusters they want to achieve; the decisions in C and K will be determined by the dataset size (number of objects). This technique will be more efficient for large sized data with more than 100 number of objects according to the running results shown in Section 7.3.8. If the number of objects are between 100 to 1,000, then K will be set to 3 and C will be set to be a floor rounding value of $(N \times 20\%)/M$. If the number of objects are between 1,000 to 10,000, then K will be set to 5 and C will be set to be a floor rounding value of $(N \times 30\%)/M$. For more than 10,000, then K will be set to 7 and C will be set to be a floor rounding value of $(N \times 40\%)/M$. These are decided according to the simulated data size ($N = 100$) in Section 7.3 and the simulated data size (N =5,000 to 10,000) in the paper of Karypis et al. [59].

7.3.10 Simulation Results

In this section, I will compare clustering results from both spatial hierarchical clustering and Chameleon spatial hierarchical clustering in all different scenarios (Figures 7.11, 7.12, 7.13 and 7.14). Data are generated from all four different scenarios detailed in Section 7.3.2, with results shown in tables. For all scenarios, tuning parameters required in Chameleon spatial hierarchical clustering will be set as $K = 3$, $M = 10$, $C = 2$ ($C = 4$ for data of type given in Figure 7.14(b)), $\alpha_0 = 1$, $\alpha = 0.5$ (for even number of objects

or $\alpha = 0.55$ for odd number of objects in Merging stage). PH will also be used to select the number of clusters in both spatial clustering techniques. The number of simulations for each scenario is repeated 100 times.

TABLE 7.5: Clustering Results for Data from All Sets of the Type Given in Figure 7.11(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
Distributions generated from simulation set-up 1			
CSHC[a]	0.870 (0.028)	3.80 (0.41)	6 (4+2)
SHC[b]	0.657 (0.075)	8.83 (1.39)	6 (4+2)
Distributions generated from simulation set-up 2			
CSHC	0.732 (0.063)	7.54 (0.37)	6 (4+2)
SHC	0.545 (0.089)	10.87 (1.32)	6 (4+2)
Distributions generated from simulation set-up 3			
CSHC	0.936 (0.023)	4.88 (0.19)	6 (4+2)
SHC	0.683 (0.081)	8.93 (1.42)	6 (4+2)
Distributions generated from simulation set-up 4			
CSHC	0.602 (0.058)	7.65 (0.44)	6 (4+2)
SHC	0.547 (0.085)	10.67 (1.47)	6 (4+2)

[a] CSHC is short for the Chameleon spatial hierarchical clustering.

[b] SHC is short for the spatial hierarchical clustering.

The simulation results for 7.11(b), 7.12, 7.13 and 7.14 are shown in Appendix A.6

The simulation results for locations in Figures 7.11(b), 7.12, 7.13 and 7.14 are shown in Appendix A.6. The tables show that the Chameleon spatial hierarchical clustering ran much faster than the spatial hierarchical clustering across all different scenarios. Chameleon spatial hierarchical clustering does better in a majority of scenarios with high average ARI and the formed numbers of clusters closer to the actual numbers of clusters. Chameleon spatial hierarchical clustering tends to form fewer clusters but each one with larger number of areal units as it has difficulty in identifying noise points.

In Table 7.5 we can see that, when the number of noise points is small, the location is a condensed distribution of areas (i.e. locations in Figures 7.11 and 7.13) and the mixing proportion of different distributions is similar, both spatial hierarchical clustering and Chameleon spatial hierarchical clustering can get good results with higher ARI (comparing to the truth), but Chameleon spatial hierarchical clustering achieves slightly

higher ARI in all four different distributions set. However, in comparison to Table 7.5, estimated clusterings from Chameleon spatial hierarchical clustering shown in Table A.22 are slightly worse with lower ARIs, but are still good in general, for which ARIs are around 0.5 to 0.8. Instead, the behaviors of spatial hierarchical clustering are less affected by the increasing number of noise points, the ARIs are similar in both tables (Tables 7.5 and A.22), the estimated total numbers of clusters are much closer to the true total numbers of clusters. So it is more likely to conclude that Chameleon spatial hierarchical clustering is sensitive to the number of noise points. This conclusion can also be detected by comparing the same location scenario but with different number of noise points (e.g. Tables A.23 and A.24). From Tables A.27 and A.28, we can find that neither of these two spatial clustering techniques are good at clustering the sparse distributions and different mixing proportion areal data. It is interesting to notice that the ARIs in Table A.28 are not worse than the ARI in Table A.27, which means with the sparse distributions (i.e. locations in Figures 7.12 and 7.14) and different mixing proportion areal data, the behaviors of Chameleon spatial hierarchical clustering is less affected by the number of noise points. In addition, the larger variance (i.e. 0.6 and 8) compared with the mean will affect both the numbers of clusters and the ARIs in all of these different scenarios. The estimated clustering results will get slightly worse, with lower ARIs, but still can capture the main clustering structure as the ARIs are still positive. ARI will be zero when areas are randomly assigned to different clusters, which means the estimated clustering is hardly similar to the true classification. If this occurs (ARI is around 0), it will indicate that both spatial hierarchical clustering and Chameleon hierarchical clustering are highly sensitive to the variance.

In the scenarios discussed above it was assumed areal units from the same cluster have independent variables/dimensions. However, dimensions of the areal units from the same cluster can to dependent, i.e. the cluster covariance matrix have non-zero off-diagonal values. So in order to extend the applicable fields of the newly proposed clustering technique, dependent dimensions within clusters in multivariate space will also be used in simulations. They will be added to the scenarios with the best performance (i.e. the number of clusters and average ARI) among all the independent dimensions within cluster scenarios. In addition, I will also simulate the scenarios when different dimensions within cluster have different impact on areal units, i.e. the covariance matrix diagonals have different values.

Comparing different scenarios and distributions, the scenario in Figure 7.11(a) gives better results in both the number of clusters and ARI. In addition, when the mean levels between two groups are more different, the simulation results will be higher in ARI and number of clusters is closer to the actual number of clusters, so I will compare the

performances about the dependent dimensions and different diagonals in these scenario (in Figure 7.11(a), distributions 3 and 4). The off diagonal elements of the simulations will be set as 0.5 (weak correlation) and 0.8 (strong correlation) separately, then the covariance matrices are $\begin{pmatrix} 2 & 0.5 \\ 0.5 & 2 \end{pmatrix}$ (distribution set 5) or $\begin{pmatrix} 2 & 0.8 \\ 0.8 & 2 \end{pmatrix}$ (distribution set 6) and $\begin{pmatrix} 8 & 0.5 \\ 0.5 & 8 \end{pmatrix}$ (distribution set 7) or $\begin{pmatrix} 8 & 0.8 \\ 0.8 & 8 \end{pmatrix}$ (distribution 8) in order to guarantee the determinant to be non-negative and the covariance is symmetric. In the different diagonals simulations, the covariance matrix in distributions 3 will be replaced by $\begin{pmatrix} 0.15 & 0 \\ 0 & 0.35 \end{pmatrix}$ (distribution set 9), the covariance matrix in distribution 4 will be replaced by $\begin{pmatrix} 7 & 0 \\ 0 & 9 \end{pmatrix}$ (distribution set 10).

TABLE 7.6: Clustering Results for Dependent Dimensions Given in Figure 7.11(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
Distributions generated from simulation set-up 5			
CSHC[a]	0.930 (0.028)	4.78 (0.17)	6 (4+2)
SHC[b]	0.675 (0.088)	8.64 (1.27)	6 (4+2)
Distributions generated from simulation set-up 6			
CSHC	0.911 (0.034)	4.63 (0.28)	6 (4+2)
SHC	0.672 (0.074)	8.50 (1.82)	6 (4+2)
Distributions generated from simulation set-up 7			
CSHC	0.589 (0.095)	7.54 (0.78)	6 (4+2)
SHC	0.540 (0.078)	10.48 (1.63)	6 (4+2)
Distributions generated from simulation set-up 8			
CSHC	0.582 (0.058)	7.42 (0.53)	6 (4+2)
SHC	0.536 (0.085)	10.21 (1.89)	6 (4+2)

[a] CSHC is short for the Chameleon spatial hierarchical clustering.

[b] SHC is short for the spatial hierarchical clustering.

TABLE 7.7: Clustering Results for Different Variances Given in Figure 7.11(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
Distributions generated from simulation set-up 9			
CSHC[a]	0.921 (0.083)	4.67 (0.17)	6 (4+2)
SHC[b]	0.676 (0.086)	8.76 (1.31)	6 (4+2)
Distributions generated from simulation set-up 10			
CSHC	0.593 (0.089)	7.51 (0.53)	6 (4+2)
SHC	0.542 (0.072)	10.65 (1.39)	6 (4+2)

[a] CSHC is short for the Chameleon spatial hierarchical clustering.

[b] SHC is short for the spatial hierarchical clustering.

From Table 7.6 we can see that the correlation or dependence between dimensions has an influence on the clustering results. When the between dimension correlation gets stronger, then the average ARI will get slightly lower and the number of clusters will be lower. This happened across all these four scenarios regardless of the variance magnitude. From Table 7.7 we can see that the difference in diagonals has little influence on the average ARIs and the number of clusters, the clustering results do not vary a lot by comparing Tables 7.5 and 7.7, which means that the difference in diagonals does not appear to make the clustering results worse, neither much lower ARI nor very low or high number of clusters.

7.4 Chameleon Spatial Hierarchical Clustering Applied to Glasgow CPEP Data

Here I apply Chameleon spatial hierarchical clustering to the Glasgow data introduced in Section 5.2. In our example, we set $K = 5$ (chosen using the histogram in Figure A.1 in Appendix A.7), $M = 30$, $C = 2$ (according to Section 7.3.9), $\alpha_0 = 1$ and $\alpha = 0.5$ (for even number of objects or $\alpha = 0.55$ for odd number of objects in Merging stage), the number of clusters with the maximal PH (0.417) is 13. In spatial hierarchical clustering, the number of clusters with the maximal PH (0.280) is 30. The clusterings from both spatial clustering techniques are displayed in Figures 7.15 and 7.16.

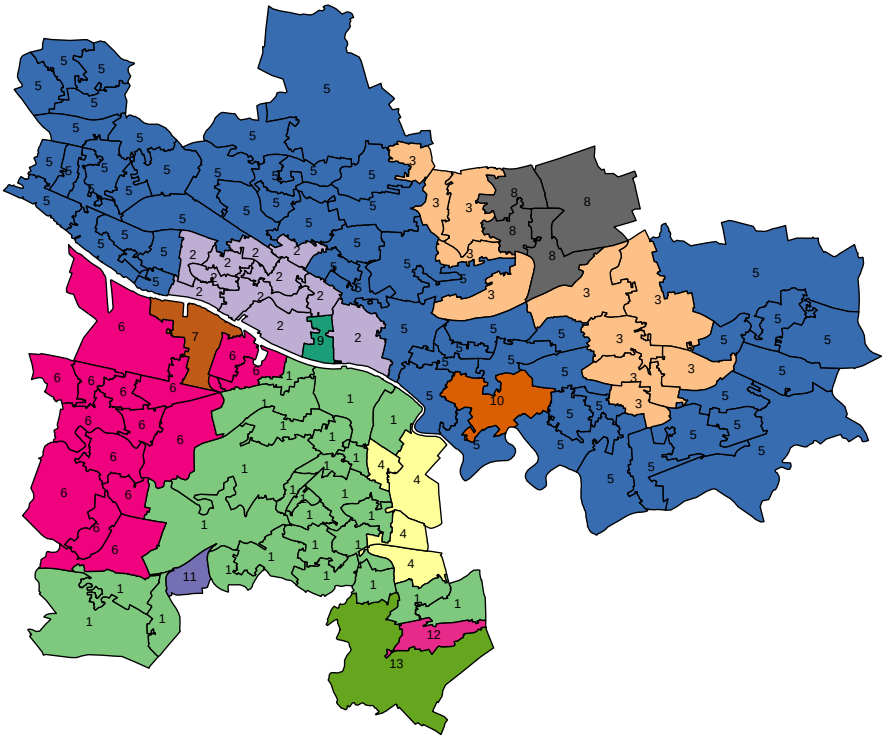


FIGURE 7.15: Clustering Based on Chameleon Spatial Hierarchical Clustering $G = 13$

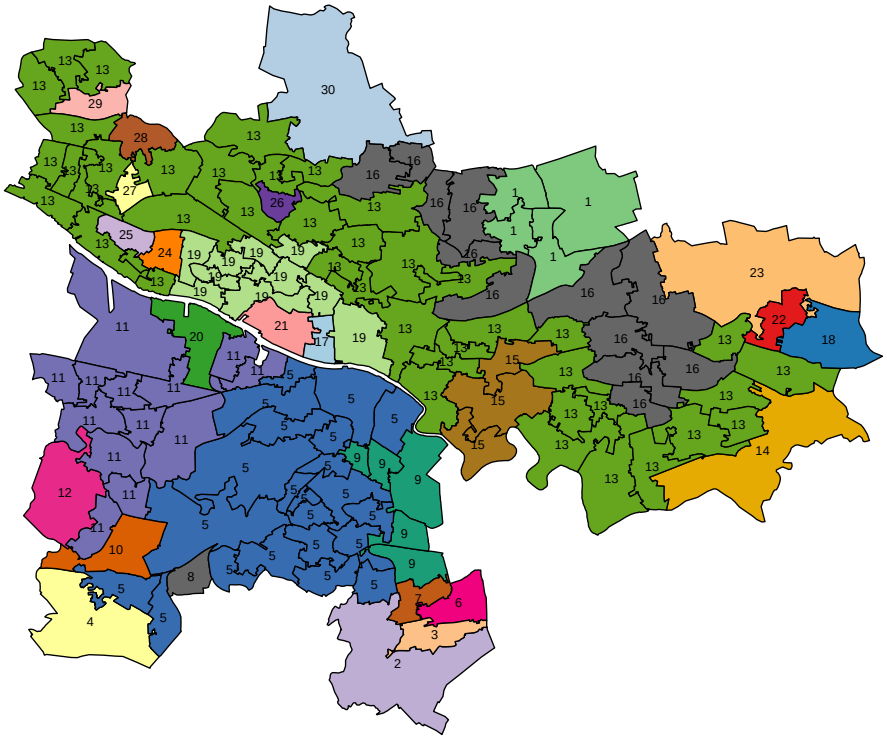
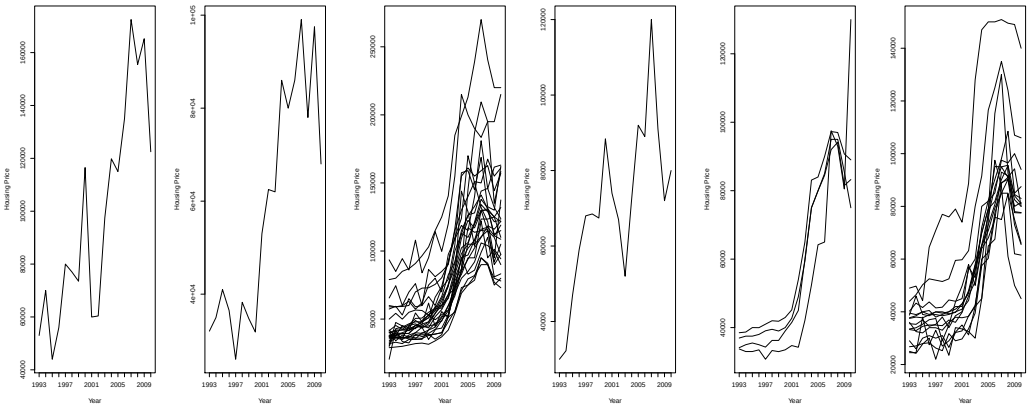


FIGURE 7.16: Clustering Based on Spatial Hierarchical Clustering $G = 30$

In the Chameleon spatial hierarchical clustering, the Glasgow housing market is mainly divided into six non-singleton submarkets with two of them lying on the south side of the Clyde river and the other four lying on the north side of the Clyde river. Spatial hierarchical clustering mainly clustered the Glasgow housing market into similar-sized submarkets. The adjusted Rand index between these two clusterings is 0.652, which

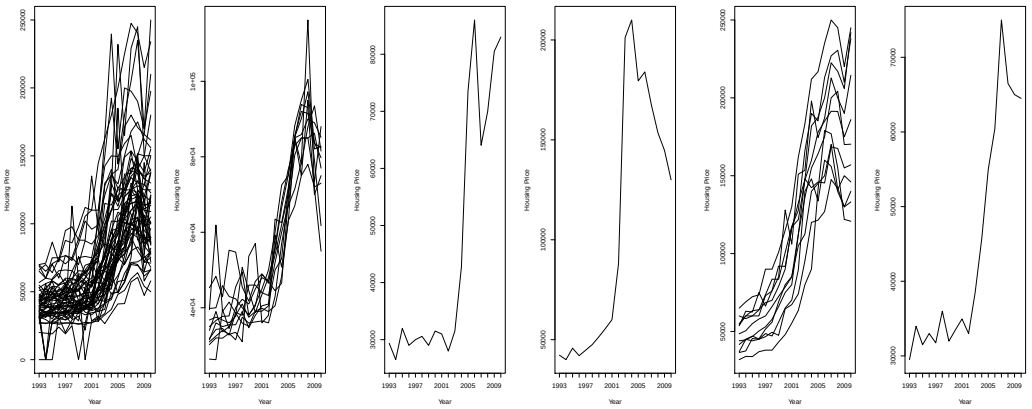
means that the two clusterings are fairly similar. For example, in both clusterings, the housing prices in the West End have been identified as growing differently from the East End. In order to better visually display the house prices of all the intermediate zones over years, their time series plots are shown in Figure 7.17.



(a) Areas in Clusters 13 and 12 in Figure 7.15

(b) Areas in Clusters 1 and 11 in Figure 7.15

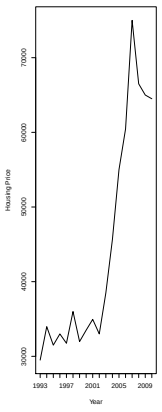
(c) Areas in Clusters 4 and 6 in Figure 7.15



(d) Areas in Clusters 5 and 3 in Figure 7.15

(e) Areas in Clusters 10 and 9 in Figure 7.15

(f) Areas in Clusters 2 and 7 in Figure 7.15



(g) Areas in Clusters 8 in Figure 7.15

FIGURE 7.17: Time Series Based on Chameleon Spatial Hierarchical Clustering

It is clear that these 13 clusters have very different changing pattern across clusters, but

similar patterns within clusters. Specially, in cluster 2 (West End) in Figure 7.17(f), all the areal units from this cluster had a very similar pattern over these years. Cluster 9 (Anderston) is neighbouring to cluster 2, but its housing price changed at a steeper rate than areas in cluster 2. Based on the market information, we know that in Anderston, there are more modern building and properties than the rest of areas in the West End, so it is reasonable to split into different clusters.

7.5 Summary of Chameleon Spatial Hierarchical Clustering

In this chapter, I proposed a new modified clustering technique, Chameleon spatial hierarchical clustering, which is motivated by the novel idea used in spatial hierarchical clustering [13], which means only the geographically connected areal units can be grouped together into clusters. The other motivation in proposing Chameleon spatial hierarchical clustering is the input data, similarity data. One of the applications in this thesis is to use CPEP data to cluster the Glasgow housing market and CPEP data are similarity data which can be used directly in Chameleon spatial hierarchical clustering. The application of Chameleon spatial hierarchical clustering to the CPEP data clustered the Glasgow housing market into 13 groups and shows a strong similarity to the clustering obtained from spatial hierarchical clustering.

Chapter 8

Spatially Constrained Finite Mixture Model with Noise Component

In Chapters 6 and 7, I discussed two spatial clustering methods, spatial hierarchical clustering [13] and Chameleon spatial hierarchical clustering, which group areal data without using any statistical models. In this chapter, I will propose a novel spatial clustering method which models areal data by using an augmented finite mixture model extending the basic finite mixture models introduced in Chapter 4. There are two issues with using the basic finite mixture model in modeling areal data. 1) It ignores the spatial information and will not necessarily produce spatially contiguous clusters; I will address this issue by incorporating information from a neighbourhood matrix (a binary matrix indicating which areas share borders by 1's, otherwise, by 0's) into the finite mixture model. 2) Finite mixture models with Gaussian components fail to fit when there is a singleton cluster or too many anomalous points in the clustering. When a dataset is mainly made up clusters of Gaussian distributions, there might be some noise points which do not follow any of those distributions. We allow for this possibility by extending the finite Gaussian mixture model to a finite Gaussian mixture model with an additional uniform distribution to model unusual objects in order to make the spatially constrained finite mixture models more robust. The nearest-neighbour clutter removal method [21] will then be used to identify the initial singleton clusters before modeling the data points. Objects assigned to the Gaussian components will be considered as clusters, objects assigned to the uniform noise component will be designated as noise or anomalous points. An alternative method to deal with the singleton clusters problem is

by including some prior information in the spatially constrained finite mixture model, which will also be explored. In addition, I will extend the spatially constrained Gaussian mixture model from univariate to multivariate space. At the end of this chapter, I will use the newly proposed two spatial clustering models to cluster Glasgow housing data.

8.1 Spatially Constrained Finite Gaussian Mixture Model

In Chapter 4, I introduced basic finite mixture models. This probability model assumes that data are generated from a set of J component density functions with some unknown vector of parameters $\boldsymbol{\theta}$ (e.g. for Gaussian component, $\boldsymbol{\theta} = (\boldsymbol{\mu}, \mathbf{T})$, where $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_J\}$, $\mathbf{T} = \{\mathbf{T}_1, \cdots, \mathbf{T}_J\}$) and some set of mixing proportions $\boldsymbol{\pi}$, subject to $\sum_{j=1}^{J} \pi^j = 1$, $0 \leq \pi^j \leq 1$. Ignoring the spatial information, we can use (8.1) to model the data. All objects with similar properties are grouped into the same cluster.

$$f\left(\mathbf{X}_i \mid \mathbf{p}, \boldsymbol{\mu}, \mathbf{T}\right) = \sum_{j}^{J} \pi^j \phi_j \left(\mathbf{X}_i \mid \boldsymbol{\mu}_j, \mathbf{T}_j\right). \tag{8.1}$$

However, areal data is different from other data, as they are geographically connected in nature. In other words, areas in the same cluster should not only have similar properties, but also should be geographically connected. Considering this essential characteristic in areal data, S.Sanjay-Gopal and Thomas J. Hebert [46] proposed spatially variant finite mixtures (SVFM), which will be explained below.

Let $\mathbf{p}_i = \left\{p_i^1, \cdots, p_i^J\right\}$ denote the component prior probability vector of the i^{th} areal unit [46], where p_i^j is the prior probability of the i^{th} area belonging to the j^{th} cluster. $\mathbf{Z}_i$ is a discrete latent membership variable of model (8.1), so $\mathrm{P}(Z_i = j) = p_i^j$. The SVFM defines the density function of the i^{th} observation as

$$f\left(\mathbf{X}_i \mid \mathbf{p}, \boldsymbol{\mu}, \mathbf{T}\right) = \sum_{j}^{J} p_i^j \phi_j \left(\mathbf{X}_i \mid \boldsymbol{\mu}_j, \mathbf{T}_j\right). \tag{8.2}$$

A model to incorporate the geographical connections between parameters $\mathbf{p}_i$ is given by the Markov Random Field model defined through a Gibbs density function [41].

If the latent position variable $\mathbf{Z}_i$ in (8.1) can be modeled by Markov Random Field (MRF), then

- $P(Z_i = z_i) > 0$
- $P(Z_i = z_i \mid \mathbf{Z}_{-i} = \mathbf{z}_{-i}) = P(Z_i = z_i \mid \mathbf{Z}_{\mathbf{M}} = \mathbf{z}_{\mathbf{M}},$ where $m \in \mathbf{M}$ if $\mathbf{W}_{i,m} = 1)$,

where $\mathbf{Z}_{-i}$ are the labels of all areas except the i^{th} area and m indexes the spatially contiguous areas to the i^{th} area. It indicates that the assignment of the i^{th} area is conditionally independent of all non-adjacent area assignments given all objects adjacent to the i^{th} object. The joint distribution can be written as the product of conditional distributions,

$$P(\mathbf{Z}) = \prod_{i=1}^{N} P(Z_i \mid \mathbf{Z}_{\mathbf{M}}; m \in \mathbf{M} : \mathbf{W}_{i,m} = 1),$$

which is a mechanism to construct dependence among random variables of an areal process [65]. The Hammersley-Clifford Theorem [49] states that $\mathbf{Z}$ is an MRF in the data space if and only if

$$P(\mathbf{Z}) = \frac{1}{C} \exp\left(-\beta \sum_{i=1}^{N} V(Z_i)\right).$$

This probability is called a Gibbs MRF based prior, where C is a normalizing constant, β is a scalar parameter and $V(Z_i)$ is the neighbourhood function for a neighbourhood of object i. So the Gibbs MRF based prior of the label vector of the i^{th} area, $\mathbf{p}_i$, can be expressed as

$$f(\mathbf{p}) = \frac{1}{C} \exp\left(-\beta \sum_{i=1}^{N} V(\mathbf{p}_i)\right),$$

where $V(\mathbf{p}_i) = \sum_{m \in \mathbf{M}: \mathbf{W}_{i,m}=1} g(u_{i,m})$, which denotes the adjacency information of the i^{th} object, where $g(u_{i,\mathbf{m}})$ is the penalty function. The selected penalty function of $g(u_{i,\mathbf{m}})$ has to be nonnegative and monotonically increasing. So based on these two requirements, we can set

$$g(u_{i,m}) = \left(1 + u_{i,m}^{-1}\right)^{-1},$$

where

$$u_{i,m} = ||\mathbf{p}_i - \mathbf{p}_m||^2 = \sum_{j=1}^{J} \left(p_i^j - p_m^j\right)^2,$$

as the image storage model, which was proposed by Stuart Geman and Donald E. McClure [42]. The joint density should assign higher probability to sets of label vectors $\mathbf{p}_i$ where neighbouring $\mathbf{p}_i$ are similar and lower probability otherwise [46]. In another word, a pair of adjacent objects will be given less penalty, if they are grouped into the same cluster.

Incorporating the spatial information into model (8.1), we extend it to model

$$f\left(\mathbf{X}_i \mid \boldsymbol{\mu}, \mathbf{T}, \mathbf{V}, \mathbf{p}_i\right) \propto \left(\sum_{j=1}^{J} p_i^j \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j, \mathbf{T}_j\right)\right) \times \exp\left(-\beta V\left(\mathbf{p}_i\right)\right). \tag{8.3}$$

β is a scalar parameter which measures the spatial smoothness [26] and is commonly used in image segmentation. The assumption of this Gibbs MRF based prior can be explained if we define $\mathbf{p}_i$ by the mean of its neighbours,

$$\hat{p}_i^j = \frac{1}{|\sum_m I(\mathbf{W}_{i,m} = 1)|} \sum_{m \in \mathbf{M}: \mathbf{W}_{i,m}=1} p_m^j, j = 1, \cdots, J,$$

where $I(\mathbf{W}_{i,m} = 1)$ is an indicator variable which denotes the neighbours of the i^{th} object and $|\sum_m I(\mathbf{W}_{i,m} = 1)|$ is the number of neighbours of the i^{th} area. The prediction error is $\varepsilon_i^j = p_i^j - \hat{p}_i^j$, Gibbs MRF based prior is based on the assumption that $|\sum_m I(\mathbf{W}_{i,m} = 1)|\, \varepsilon_i^j \sim N(0, \beta^2)$ [83]. So β^2 is the variance of the error of $\mathbf{p}_i$ which only considers m neighbours for which $\mathbf{W}_{i,m} = 1$. In this thesis, we use the conclusion achieved by Christophoros Nikou [83] to estimate β^2, which is expressed as,

$$\beta^2 = \frac{1}{N}\frac{1}{J}\sum_{i=1}^{N}\sum_{j=1}^{J}\left(\sum_{m \in \mathbf{M}: \mathbf{W}_{i,m}=1} (p_i^j - p_m^j)\right)^2.$$

According to the published papers of Gopal and Hebert [46], Blekas et al. [18] and Nikou et al. [83], etc, the estimated β is fixed in the estimating procedure and is not treated as a parameter but fixed in their studies and will be estimated by using the clustering obtained from spatial hierarchical clustering. However, in Christophoros Nikou's paper [83], they also improved the estimation of β by using different β_j for different clusters and also updated β_j and $\mathbf{p}_i$ at the same time across iterations. The augmented estimation of $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_J)$ can be expressed as

$$\beta^{2,(t)} = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{m \in \mathbf{M}: \mathbf{W}_{i,m}=1} (p_i^{j,(t)} - p_m^{j,(t)})\right)^2, j = 1 \cdots, J.$$

In this thesis, β will be fixed across the estimating procedure to simplify the model and will not be treated as an additional parameter.

8.2 Generalized EM Algorithm

In Section 4.6.1, I introduced the EM algorithm to estimate parameters in finite mixture models. The EM algorithm contains two steps, an E step and an M step. The E step calculates the expectation of the complete data log likelihood function. If the complete data is inaccessible, then it will be replaced with its conditional expectation given the observed data. In the M step, we look for the parameter values which can maximize the expectation function from the E step. The E step and M step will be repeated in turn until all parameters converge to stable values. Convergence criteria have been discussed in Section 2.3.1.

EM procedure:

For a given initial $\boldsymbol{\theta}^{(0)}$, we start with $t = 1$.
The estimation at the t^{th} iteration is as follows:

1. E Step: Calculate $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)}) = \mathrm{E}\left(\log f(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta}) \mid \boldsymbol{\theta}^{(t-1)}, \mathbf{Z}\right)$ at the t^{th} iteration to estimate $\mathbf{Z}$.

2. M Step: Choose $\boldsymbol{\theta}^{(t)}$ which can maximize $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)})$,

$$Q(\boldsymbol{\theta}^{(t)}; \boldsymbol{\theta}^{(t-1)}) \geq Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)}).$$

3. Increase t by 1.

Convergence: Repeat steps 1 to 3 many times until all parameters converge. More detail about convergence has been given in section 2.3.1.

However, in some scenarios, it is impossible to directly find the maximum value in the M steps. Specifically, in this chapter, as we incorporate the spatial information into the basic finite mixture model (8.1), the probability vector $\mathbf{p} = \left(p^1, \cdots, p^J\right)$ in (8.1) will be extended to matrix $\mathbf{p} = (\mathbf{p}_1, \cdots, \mathbf{p}_N)$, where $\mathbf{p}_i = \left(p_i^1, \cdots, p_i^J\right)$ in (8.3) and subject to

$$0 \leq p_i^j \leq 1,$$

$$\sum_{j=1}^{J} p_i^j = 1.$$

We can see that (8.3) is the same form as (8.1) except for this extended $\mathbf{p}$ and including an additional term used to explain the spatial information. This additional term precludes locating a solution to $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)})$ function in closed form. So instead of finding a value which can globally maximize the Q function, we just need to find a value which increases the Q function over the previous step. There are several generalized expectation maximization (GEM) [80] algorithms that can do this, such as GEMs based on Newton's method, GEM based on ordinary gradient descent and GEMs based on gradient projection. Gradient descent is a first-order method, while Newton's method is a second-order method, as it uses both the first derivative and the second derivative (Hessian). So gradient descent requires less computation and Newton's method requires much more computation time in every iteration [95]. Gradient descent minimizes a function by moving in the negative gradient direction at each step. There is no constraint on the variable. On the other hand, projected gradient descent minimizes a function subject to a constraint. At each step we move in the direction of the negative gradient, and then project onto a feasible set [96]. So in this thesis, I will use a GEM based on the gradient projection algorithm. However, one the limitations of GEM based on the gradient projection algorithm is it usually takes more iterations to reach the optimal point [5]. The differences between the EM algorithm and the GEM based on gradient projection in the following are written in italics.

GEM based on gradient projection procedure:

For a given initial $\boldsymbol{\theta}^{(0)}$, we start with $t = 1$.
The estimation at the t^{th} iteration is as follows:

1. E Step: Calculate $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)}) = \mathrm{E}\left(\log f(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta}) \mid \boldsymbol{\theta}^{(t-1)}, \mathbf{Z}\right)$ at the t^{th} iteration to estimate $\mathbf{Z}$.

2. Generalized M Step: Choose $\boldsymbol{\theta}^{(t)}$ which can *increase* $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)})$ *over the previous iteration*,
$$Q(\boldsymbol{\theta}^{(t)}; \boldsymbol{\theta}^{(t-1)}) \geq Q(\boldsymbol{\theta}^{(t-1)}; \boldsymbol{\theta}^{(t-1)}).$$
More specifically, the parameter space is obtained from the gradient via a projection matrix.

3. Increase t by 1.

Convergence: Repeat steps 1 to 3 many times until all parameters converge. More detail about convergence is given in section 2.3.1.

Details information about implementing this GEM based on the gradient projection is given in Sections 8.3.3 and 8.4.1.

8.3 Spatially Constrained Finite Mixture Model with Noise Component

One issue that challenges the applicability of this spatially constrained mixture model is the existence of singleton clusters - clusters made up of a single point. These singleton clusters can cause the fitting of the spatially constrained finite mixture model to fail. The proposed spatially constrained finite mixture model (8.3) assumes all points can be modeled by one of the Gaussian component distributions. Gaussian distributions have nice properties (a symmetric distribution with most observations located in the middle with high probability, while a small proportion of data located on the rim of the distribution with low probability). So as long as there are not too many outliers and the data is not skewed, we can use Gaussian distributions to model the data. When a dataset is mainly made of clusters of those Gaussian distributions, there might be some noise points which do not follow any of those distributions. We allow for this possibility by extending the finite Gaussian mixture model to a finite Gaussian mixture model with an additional uniform distribution to model unusual objects. I will address the model fitting issue by incorporating a uniform term to model the noise points in order to make the spatially constrained finite mixture model more robust.

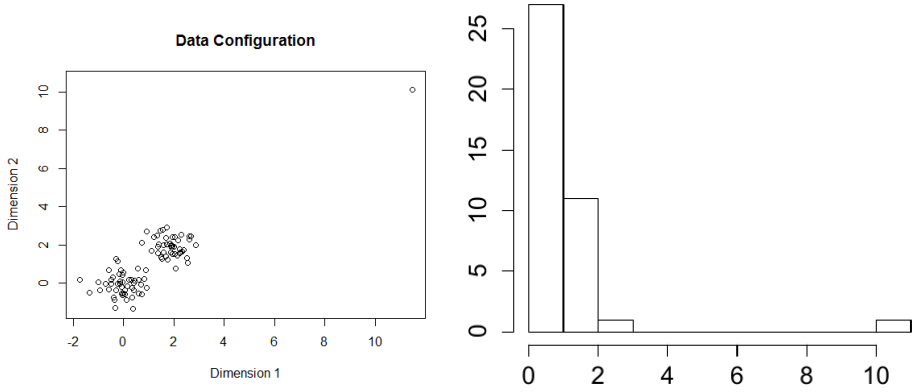
The initialization of identifying noise points has been explored by many researchers. The commonly used model-free method to identify noise points is trimmed K-means [27], which was proposed by Albertos Cuesta and Antonio Juan [27]. For a given probability of anomalous data in the data, trimmed K-means groups all objects into two large groups, one group is treated as a noise group, the rest of objects are placed into another group with J subgroups. The procedure in trimmed K-means in grouping non-noise points is similar to the procedure in K-means clustering [76]. K-means clustering groups objects by minimizing the within-cluster sum of squares (WCSS) (see SS(A) in (4.2) for greater details), but it is mainly used to search for clusters with equal-sized and spherically scattered groups, which is not required in all scenarios. Another way to identify noise points proposed by Denis Allard and Chris Fraley [12] was a model-based approach to identify the potential noise points with the basic model being a mixture of two uniform point processes. However, this is inconsistent with the components in the

spatially constrained finite mixture model used in this thesis, as we assume the finite mixture components are Gaussian distributions.

So in this thesis, I will use nearest-neighbour clutter removal [21] to identify the initial noise points. One of the advantages of nearest-neighbour clutter removal over the other methods is that it is a model-based estimation method, which can estimate the probability of an object belonging to a noise group. In addition, compared with trimmed K-means which requires us to know the proportion of noise points in the data, nearest-neighbour clutter removal does not require this.

8.3.1 Nearest-Neighbour Clutter Removal

The intuition of nearest-neighbour clutter removal method was inspired by K nearest neighbours distance (K-NN distance), which can be illustrated by using Figure 8.1(b).



(a) Data Configuration (b) Distance to the 5th Nearest Neighbour

FIGURE 8.1: Intuition for K Nearest Neighbour Clutter

Assume the given data configuration follows the pattern in Figure 8.1(a), where most objects are located on the lower left corner of the space, so they are more likely to have smaller K-NN distances; while there is one point lying on the top right corner, which is far away from the rest of objects, so this point's K-NN distance will be larger than the other objects'. So there is a higher probability for the single object being a singleton cluster. If we calculate the K-NN distance ($K = 5$) for all objects, then we will get the histogram in Figure 8.1(b), which has two peaks. Intuitively, we will name the component with shorter K-NN distances as a feature component, the one with larger K-NN distances as a clutter component.

The gamma distribution is used to describe the process of waiting time X until the K^{th} event occurs and the distribution can be expressed as,

$$F(x) = 1 - \sum_{k=1}^{K-1} \frac{(\lambda x)^k e^{\lambda x}}{k!}.$$

Similarly, let $r \in [0, +\infty]$ denote the radius of a circle around a selected point and D_K denotes the K^{th} nearest distance to a selected random point. If the K^{th} nearest distance D_K is greater than r, then there will be $K-1$ objects having distances less than r. So the distribution of the nearest K^{th} neighbours of a selected point can be modeled by a modified Gamma distribution and be expressed as

$$F_{D_K}(r) = 1 - \sum_{k=0}^{K-1} \frac{e^{-\lambda\pi r^2} \left(\lambda\pi r^2\right)^k}{k!}. \tag{8.4}$$

If $r^2 = x$, then we can tell that (8.4) is a modified gamma distribution follows $(D_K)^2 \sim \Gamma(K, \lambda\pi)$, so $D_K \sim \Gamma^{1/2}(K, \lambda\pi)$.

From the histogram shown in Figure 8.1(b), we can see that D_K can be modeled by a bimodal distribution with two components, so we can model D_K by

$$D_K \sim p\Gamma^{1/2}(K, \lambda_1\pi) + (1-p)\,\Gamma^{1/2}(K, \lambda_2\pi),$$

where p denotes the proportion of the population in a feature component with a parameter $\lambda_1\pi$, while $1-p$ denotes the proportion of population in a clutter component with a parameter $\lambda_2\pi$. The latent variables $\mathbf{Z} = (Z_1, Z_2, \cdots, Z_N)$ are distributed according to a binomial distribution and takes values 1 or 0. If $I(Z_i = 1)$ is an indicator variable which indicates that the i^{th} object belongs to the feature component, otherwise, the clutter component, then the log likelihood of the complete data is expressed as

$$\begin{aligned} & l(\boldsymbol{\lambda} \mid \mathbf{D}, \boldsymbol{\lambda}, \mathbf{Z}) \\ & = \log \prod_{i=1}^{N} \left[p\Gamma^{1/2}(K, \lambda_1\pi)^{I(Z_i=1)} \cdot (1-p)\,\Gamma^{1/2}(K, \lambda_2\pi)^{I(Z_i=0)} \right]. \end{aligned}$$

We use the EM algorithm to estimate p, λ_1 and λ_2.

The expectation of the complete data log-likelihood function at the t^{th} iteration of the

EM algorithm takes the form of

$$
\begin{aligned}
&Q\left(\boldsymbol{\lambda} \mid \mathbf{D}, \boldsymbol{\lambda}^{(t-1)}\right) \\
&\quad = \sum_{i=1}^{N} \left[E\left(Z_i = 1 \mid \mathbf{D}, \boldsymbol{\lambda}^{(t-1)}\right) \left\{ \log p + \log\left(\Gamma^{1/2}\left(K, \lambda_1^{(t-1)}\pi\right)\right)\right\} \right. \\
&\qquad \left. + E\left(Z_i = 0 \mid \mathbf{D}, \boldsymbol{\lambda}^{(t-1)}\right) \left\{ \log(1-p) + \log\left(\Gamma^{1/2}\left(K, \lambda_2^{(t-1)}\pi\right)\right)\right\}\right],
\end{aligned}
$$

- E Step:

$$
\omega_i^{(t)} = \mathrm{E}\left(Z_i = 1 \mid \mathbf{D}, \boldsymbol{\lambda}^{(t-1)}\right) = \frac{p^{(t-1)}\Gamma^{1/2}\left(K, \lambda_1^{(t-1)}\pi\right)}{p^{(t-1)}\Gamma^{1/2}\left(K, \lambda_1^{(t-1)}\pi\right) + \left(1-p^{(t-1)}\right)\Gamma^{1/2}\left(K, \lambda_2^{(t-1)}\pi\right)}.
$$

- M Step:

$$
\lambda_1^{(t)} = \frac{K\sum_{i=1}^{N}\omega_i^{(t)}}{\pi\sum_{i=1}^{N} d_i^2\omega_i^{(t)}},
$$

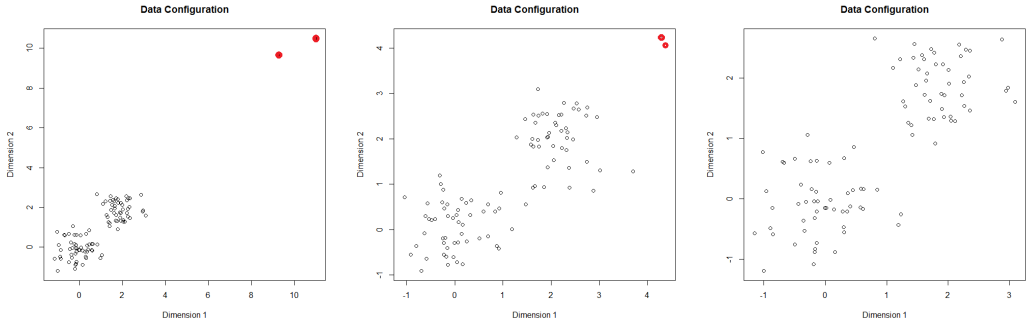
$$
\lambda_2^{(t)} = \frac{K\sum_{i=1}^{N}\left(1-\omega_i^{(t)}\right)}{\pi\sum_{i=1}^{N} d_i^2\left(1-\omega_i^{(t)}\right)},
$$

$$
p^{(t)} = \frac{\sum_{i=1}^{N}\omega_i^{(t)}}{N}.
$$

- The E and M steps will be repeated many times until all the parameters converge. More details about assessing convergence have been given in Section 2.3.1.

After convergence, $\boldsymbol{\omega}$ at the last iteration will be used to cluster objects, if $\omega_i > 1-\omega_i$, then the object will be grouped to the feature component, otherwise, the clutter component. From the empirical research point of view and the published paper [38], the noise points identified by nearest-neighbour clutter removal will be used in the spatially constrained finite mixture model with noise point. The clustering algorithms can be roughly divided into unsupervised clustering and the supervised clustering algorithms. Unsupervised clustering uses no additional information to cluster data, e.g. hierarchical clustering. Supervised clustering uses the additional information, e.g. K-means which classifies data into pre-determined clusters [10]. So the spatially finite mixture model with noise points uses the idea of supervised clustering, which use the prior information from the entire data to cluster some of the data into anomalous points and then apply the spatially constrained finite mixture model on the entire data.

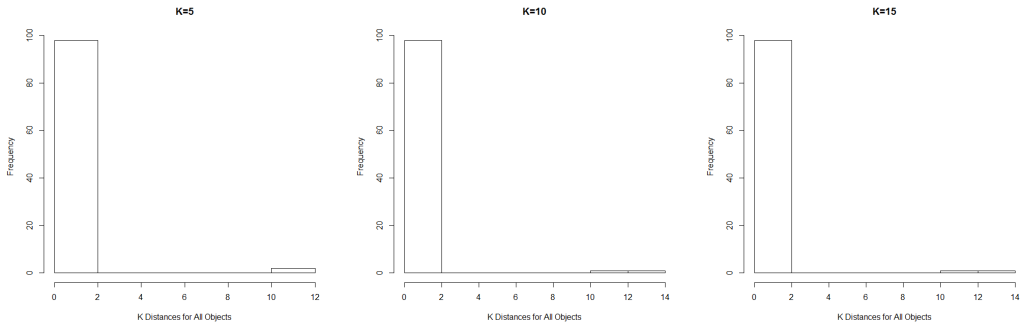
K is a user specific value. Here, we give out two data configurations for K selections.



(a) Data Configuration for Case 1 (b) Data Configuration for Case 2 (c) Data Configuration for Case 3

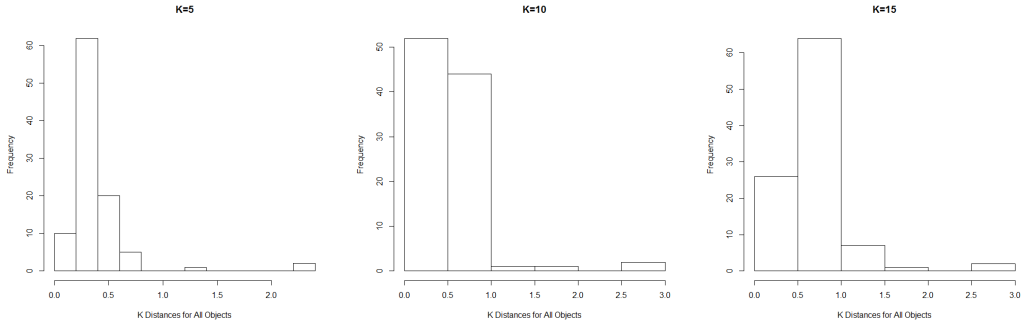
FIGURE 8.2: Intuition for K

Figure 8.2 gives three configurations about the anomalous points and non-anomalous points. In Figure 8.2(a), the red points are far away from the rest of data (black points), then the red points will be treated as the anomalous points. In Figure 8.2(b), the red points are less different from the rest of points, but they can still be told from the rest of points. While, Figure 8.2(c) shows the configuration without the anomalous points. The difference between the anomalous points and non-anomalous points can also be identified by using the K-NN histogram in Figure 8.3. The K nearest distance of a range of K for both datasets are shown in Figures 8.3 to 8.5



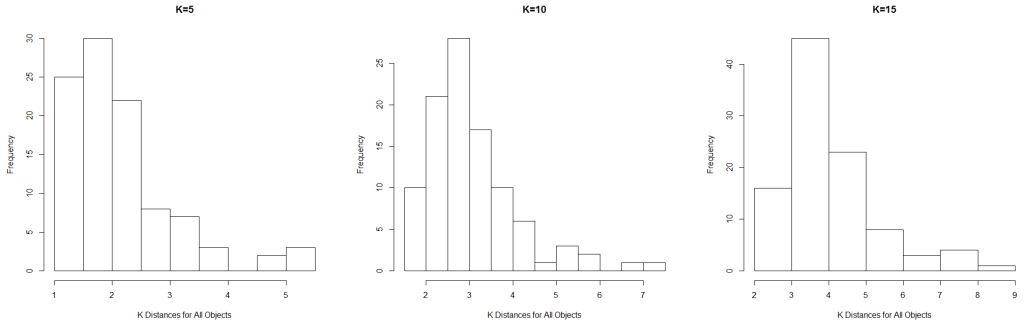
(a) K=5 for Case 1 (b) K=10 for Case 1 (c) K=15 for Case 1

FIGURE 8.3: Histograms of K in Case 1



(a) K=5 for Case 2 (b) K=10 for Case 2 (c) K=15 for Case 2

FIGURE 8.4: Histograms of K in Case 2



(a) K=5 for Case 3 (b) K=10 for Case 3 (c) K=15 for Case 3

FIGURE 8.5: Histograms of K in Case 3

From the histograms shown in Figure 8.3, it is easy to tell that the selection of K in this case has little influence on identifying the feature component and the clutter component as all these selected K can tell the feature component from the clutter component. For the less different mixture data sets, we can see from Figure 8.4, when K increases, the difference between the featured component and the clutter component in this case will be blurred. Indeed, in Figure 8.5, when K is small, it might mistakenly treat some non-anomalous points as anomalous points. So this requires us to seek for a more reliable technique to select the optimal K to avoid this mistake. In the absence of any extra information in selecting K, we can use an entropy-type measure of separation to choose K [21]. Given a selected number of K's, for each K, we calculate

$$S = \sum_{i=1}^{N} \omega_i \log(\omega_i),$$

where ω_i is the probability of the i^{th} object belongs to the feature component. We can also detect the change of S visually by using a plot with K's against separations, then

the optimal K will be the one when S levels off [21].

For the previous example in Figure 8.1, Figure 8.6 is the separation against a selection of K, from $K = 1$ to $K = 20$.

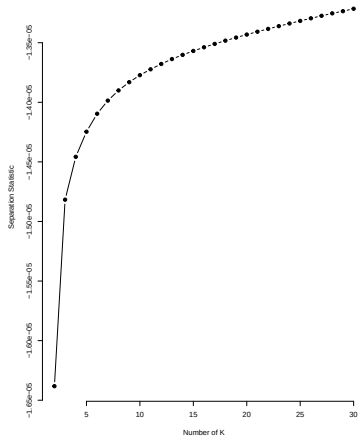


FIGURE 8.6: K Selection in Nearest-Neighbour Clutter Removal Method

We can see that K is always increasing until $K = 10$ it starts to level off, so $K = 10$ is chosen for the nearest-neighbour clutter removal method to identify the noise points.

8.3.2 An Extension of the Spatially Constrained Finite Mixture Model

In the Banfield and Raftery [15] paper, they proposed that all noise points were generated from a Poisson distribution with certain parameters. Poisson distribution is used to describe a process of $K(K \geq 1)$ events happened during a given time interval (i.e. 0 - t) and all individual events uniformly occur in the interval. If one and only one event ($K = 1$) occurs in the interval, then this process can be modeled by a uniform distribution. As all the noise points are singleton clusters, so in this thesis, I assume all noise points can be adequately modeled by a uniform distribution (which can be tested using the variable's histogram and checking its shape at the ends matches with a uniform) where they have equal probability of lying at any point in the data space, so we can extend the spatially constrained finite model to a spatially constrained finite model with a uniform distribution make the model (8.3) more robust, where the uniform distribution is used to model noise points.

Given the data configuration $\mathbf{X}_i$ and the number of non-anomalous groups J, if we

assume that all non-anomalous objects are from a finite mixture of J multivariate normal distributions with certain mixing proportions and the anomalous objects are generated from a uniform distribution which encloses the data region, then the general model of this can be expressed as (8.5). Jeffrey D. Banfield and Adrian E. Raftery [15] has proposed to use uniform distributions to model the anomalous points in one of their papers [15]. In addition, the anomalous points in the real data of the Glasgow housing market are also all intermediate zones with roughly equal number of residents, so we can assume the areas are clustered uniformly along and tightly.

$$f\left(\mathbf{X}_i \mid \boldsymbol{\mu}, \mathbf{T}, \mathbf{V}, \mathbf{p}_i\right) = \left\{ \sum_{j=1}^{J} p_i^j \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j, \mathbf{T}_j\right) + p_i^{J+1} U_{J+1}\left(\mathbf{X}_i \mid \mathbf{V}\right) \right\} \exp\left(-\beta V\left(\mathbf{p}_i\right)\right), \tag{8.5}$$

where $\sum_{j=1}^{J+1} p_i^j = 1$, $0 \leq p_i^j \leq 1$. In model (8.5), $\phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j, \mathbf{T}_j\right)$ represents a multivariate normal distribution, where $\boldsymbol{\mu}_j$ is the mean vector of the j^{th} cluster, $\mathbf{T}_j$ is the covariance matrix of the j^{th} cluster, $U_{J+1}(\cdot|\mathbf{V})$ is a uniform distribution with $\mathbf{V}$ as the volume of space enclosing the data. The sensitivity of the volume of space of the uniform distribution will be compared with both when the range is from the minimum to the maximum and the range is wider, i.e. less than the minimum or greater than the maximum or further out at both ends.

I use the data simulated from the distribution sets introduced in Section 7.3.2 in Figure 7.11(a). Compared with all these scenarios in Table 8.1, it is easy to tell when the range of the uniform distribution covers a wider space, there are fewer anomalous points detected than a narrower uniform space (range 1 in Table 8.1), which increase the average TDR. This can be explained as if the range of the uniform distribution is wider, then the probability of an area being an anomalous point will be lower, the area will be more likely in one of the clusters. However, some of the actual anomalous points cannot be identified, which results in a lower ARI and TPR (comparing ranges 2, 3, 4 with range 1 in Table 8.1). Comparing ranges 2,3 with range 4, we can see that ARI, TPR and TDR do not affected by increasing the maximum or decreasing the minimum, but are affected by changing the range of the uniform distribution. Similar results can also be seen from Table A.29. For the remainder of the thesis, I set the volume of space of the uniform distribution is from the minimum to the maximum.

8.3.3 Parameter Estimation for the Spatially Constrained Finite Mixture Model with Noise Component

TABLE 8.1: Uniform Distribution Sensitivity Comparison

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Range 1	0.876 (0.028)	4.100 (0.403)	3.700 (0.822)	7.800	6 (4+2)	1.000 (0.000)	0.652 (0.263)
Range 2	0.753 (0.033)	4.720 (0.873)	3.240 (0.963)	7.960	6 (4+2)	0.860 (0.142)	0.715 (0.218)
Range 3	0.766 (0.025)	4.640 (0.795)	3.400 (0.972)	8.040	6 (4+2)	0.875 (0.062	0.732 (0.225)
Range 4	0.744 (0.054)	4.670 (0.593)	3.380 (0.941)	8.050	6 (4+2)	0.862 (0.084)	0.736 (0.214)

[a] Range 1: the minimum to the maximum
[b] Range 2: from the minimum-20%$\times$ (the maximum $-$ the minimum) to the maximum
[c] Range 3: from the minimum to the maximum+20%$\times$ (the maximum $-$ the minimum)
[d] Range 4: from the minimum-10%$\times$ (the maximum $-$ the minimum) to the maximum+10%$\times$ (the maximum $-$ the minimum)
The simulation results for Distribution sets 2,3,4 are shown in Appendix A.8

Gradient projection is a technique for constrained optimization, it seeks the direction along the subspace tangent to the active constraints [84]. So the basic assumption of gradient projection is that the parameter lies in the subspace tangent to the active constraints. In the estimation of spatial finite mixture models with a uniform distribution, we need to increase the conditional expectation function of the complete data log likelihood given the observed data over the previous step, but constrained to

$$\begin{aligned} &0 \leq p_i^j \leq 1 \text{ for all } j = 1, \cdots, J+1, \\ &\sum_{j=1}^{J+1} p_i^j = 1. \end{aligned} \tag{8.6}$$

So I will use the GEM gradient projection algorithm to estimate the parameters.

If $I(Z_i = j)$ is an indicator variable, then it indicates whether the i^{th} object belongs to the j^{th} component. The complete data log-likelihood function at the t^{th} iteration is in the following form:

$$\begin{aligned} & l\left(\boldsymbol{\mu}, \mathbf{T} \mid \mathbf{X}, \boldsymbol{\mu}^{(t-1)}, \mathbf{T}^{(t-1)}, \mathbf{V}, \mathbf{Z}, \mathbf{p}\right) \\ \propto & \log\left(\prod_{i=1}^{N}\left(\prod_{j=1}^{J}\left[p_i^{j,(t-1)}\phi_j\left(\mathbf{X}_i; \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)}\right)\right]^{I(Z_i=j)}\right)\left[p_i^{J+1,(t-1)}U\left(\mathbf{X}_i; \mathbf{V}\right)\right]^{I(Z_i=J+1)}\right) \\ & \qquad - \beta\sum_{i=1}^{N} V(\mathbf{p}_i) \\ = & \sum_{i=1}^{N}\left[\sum_{j=1}^{J} I(Z_i = j)\left\{\log(p_i^{j,(t-1)}) + \log\left(\phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)}\right)\right)\right\}\right. \\ & \left. + I(Z_i = J+1)\left\{\log(p_i^{J+1,(t-1)}) + \log\left(\mathbf{U}_{J+1}\left(\mathbf{X}_i \mid V\right)\right)\right\}\right] - \beta\sum_{i=1}^{N} V(\mathbf{p}_i). \end{aligned}$$

The conditional expectation of the complete data log likelihood given the observed data is

$$\begin{aligned} & Q(\boldsymbol{\mu}, \mathbf{T} \mid \boldsymbol{\mu}^{(t-1)}, \mathbf{T}^{(t-1)}, \mathbf{V}, \mathbf{X}, \mathbf{Z}, \mathbf{p}) \\ = & \sum_{i=1}^{N}\left[\sum_{j=1}^{J} E(I(Z_i = j) \mid \mathbf{X}_i, \boldsymbol{\mu}^{(t-1)}, \mathbf{T}^{(t-1)}) \cdot \left\{\log(p_i^{j,(t-1)}) + \log\left(\phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)}\right)\right)\right\}\right. \\ & \left. + E(I(Z_i = J+1) \mid \mathbf{X}_i, \mathbf{V}) \cdot \left\{\log(p_i^{J+1,(t-1)}) + \log\left(\mathbf{U}\left(\mathbf{X}_i \mid \mathbf{V}\right)\right)\right\}\right] \\ & \qquad - \beta\sum_{i=1}^{N} V(\mathbf{p}_i). \end{aligned}$$

We denote $\mathbf{q}_{i,((J+1)\times 1)}$ as the $(J+1)$ dimensional vector of first derivative of $Q(\boldsymbol{\mu}, \mathbf{T} \mid \boldsymbol{\mu}^{(t-1)}, \mathbf{T}^{(t-1)}, \mathbf{V}, \mathbf{X}, \mathbf{Z})$ with respect to the $(J+1)$ dimensional vector $\mathbf{p}_{i,((J+1)\times 1)}$,

$$q_i^j \equiv \frac{\partial Q(\boldsymbol{\mu}, \mathbf{T} \mid \boldsymbol{\mu}^{(t-1)}, \mathbf{T}^{(t-1)}, \mathbf{V}, \mathbf{X}, \mathbf{Z}, \mathbf{p})}{\partial p_i^j} \text{ for all } j = 1, \cdots, J+1.$$

In order to increase $Q(\boldsymbol{\mu}, \mathbf{T} \mid \boldsymbol{\mu}^{(t-1)}, \mathbf{T}^{(t-1)}, \mathbf{V}, \mathbf{X}, \mathbf{Z}, \mathbf{p})$ over the previous step, we seek a feasible direction $(J+1)$ dimensional vector $\mathbf{d}_{i,((J+1)\times 1)}$ (later the subscript in the bracket will be used to denote the matrix dimensions), satisfying $\mathbf{d}_i^T \mathbf{q}_i > 0$, then the movement in the direction $\mathbf{d}_i$ will cause an increase in the function $Q(\boldsymbol{\mu}, \mathbf{T} \mid \boldsymbol{\mu}^{(t-1)}, \mathbf{T}^{(t-1)}, \mathbf{V}, \mathbf{X}, \mathbf{Z}, \mathbf{p})$. These directions $\mathbf{d}_i$ consist of the tangent plane of the active constraints and all directions satisfy

$$(\mathbf{A}_{i,((J+1)\times K_0)})^T \mathbf{d}_{i,((J+1)\times 1)} = \mathbf{0}_{(K_0 \times 1)}, \tag{8.7}$$

where $\mathbf{A}_{i,((J+1)\times K_0)}$ is the gradient of constraints, K_0 is the number of equality constraints (the number of active constraints).

So seeking for a new $\mathbf{p}_i$ along the tangent plane of the active constrains can be expressed as

$$\mathbf{p}_i^{(t)} = \mathbf{p}_i^{(t-1)} + \alpha \mathbf{d}_i,$$

and both $\mathbf{p}_i^{(t)}$ and $\mathbf{p}_i^{(t-1)}$ satisfy (8.6). As the steepest increment direction is required to satisfy (8.7), so we can pose the problem as

$$\begin{aligned} &\min \ \mathbf{d}_i^T \mathbf{q}_i \\ &\text{s.t. } \mathbf{A}_i^T \mathbf{d}_i = \mathbf{0}_{(K_0 \times 1)} \\ &\quad \mathbf{d}_i^T \mathbf{d}_i = 1, \end{aligned} \tag{8.8}$$

which means that we want to find the direction with the most negative directional derivative which satisfies (8.7). If we use Lagrange multipliers $\boldsymbol{\lambda}_{(K_0 \times 1)}$ and μ, then we can re-express (8.8) to

$$\mathcal{L} = \mathbf{d}_i^T \mathbf{q}_i - \mathbf{d}_i^T \mathbf{A}_i \boldsymbol{\lambda} - \mu(\mathbf{d}_i^T \mathbf{d}_i - 1).$$

The condition of $\mathcal{L}$ being stationary is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{d}_i} = \mathbf{q}_i - \mathbf{A}_i \boldsymbol{\lambda} - 2\mu \mathbf{d}_i = \mathbf{0}. \tag{8.9}$$

If we premultiply (8.9) by $\mathbf{A}_i^T$ and take account of (8.7), then we can get

$$\mathbf{A}_i^T \mathbf{q}_i - \mathbf{A}_i^T \mathbf{A}_i \boldsymbol{\lambda} = \mathbf{0},$$

so

$$\boldsymbol{\lambda} = (\mathbf{A}_i^T \mathbf{A}_i)^{-1} \mathbf{A}_i^T \mathbf{q}_i.$$

If we substitute $\boldsymbol{\lambda}$ into (8.9), then we can get

$$\mathbf{d}_i = \frac{1}{2\mu} \left[\mathbf{I} - \mathbf{A}_i (\mathbf{A}_i^T \mathbf{A}_i)^{-1} \mathbf{A}_i^T \right] \mathbf{q}_i$$

So

$$\mathbf{R}_i = \mathbf{I} - \mathbf{A}_i (\mathbf{A}_i^T \mathbf{A}_i)^{-1} \mathbf{A}_i^T.$$

is the projection matrix which can be used to estimate $\mathbf{d}_i$ [73].

The constraint matrix $\mathbf{A}_i$ is created based on constraints (8.6). The constraint, $p_i^1 + p_i^2 + \cdots + p_i^{J+1} = 1$, is always active, the other active constraints depending on the number of zero probabilities. If there are two zero probabilities in $\mathbf{p}_i$, i.e. $p_i^1 = 0$ and $p_i^2 = 0$, then $\mathbf{A}_i^T$ can be expressed as:

$$\mathbf{A}_i^T = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix}_{3 \times (J+1)}.$$

So the general form of $\mathbf{R}_i$ can be expressed as:

$$\mathbf{R}_i \begin{cases} 0 & \text{if } p_i^j = 0 \text{ or } p_i^l = 0, \text{for all } j \neq l \\ \frac{K_0 - 1}{K_0} & \text{if } j = l \text{ and } p_i^j \neq 0 \\ -\frac{1}{K_0} & \text{otherwise} \end{cases}$$

The Generalized EM algorithm for parameters can be summarized as follows:

- E Step: For all objects $i = 1, 2, \cdots, N$.

$$\begin{aligned} &\mathrm{E}\left(I(Z_i = j) \mid \mathbf{X}_i, \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)} \right) \equiv \omega_i^{j,(t)} \\ &= \frac{p_i^{j,(t-1)} \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)} \right)}{\sum_{j=1}^{J} p_i^{j(t-1)} \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)} \right) + p_i^{J+1,(t-1)} \mathbf{U}\left(\mathbf{X}_i \mid \mathbf{V} \right)}, \\ &\text{for } j = 1, \cdots, J. \end{aligned}$$

$$\begin{aligned} &\mathrm{E}\left(I(Z_i = J+1) \mid \mathbf{X}_i, \mathbf{V} \right) \equiv \omega_i^{J+1,(t)} \\ &= \frac{p_i^{J+1,(t-1)} \mathbf{U}\left(\mathbf{X}_i \mid \mathbf{V} \right)}{\sum_{j=1}^{J} p_i^{j,(t-1)} \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)} \right) + p_i^{J+1,(t-1)} \mathbf{U}\left(\mathbf{X}_i \mid \mathbf{V} \right)}. \end{aligned}$$

- Generalized M Step 1: For each object i, do Step a to Step e.
  - Step a: For $j = 1, \cdots, J$

$$c_1 = \sum_{j=1}^{J+1} \omega_i^{j,(t)} \log p_i^{j,(t-1)} - \beta V(\mathbf{p}_i),$$

$$q_i^{j,(t)} = \frac{\phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)}\right)}{\sum_{j=1}^{J} p_i^{j,(t-1)} \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)}\right) + p_i^{J+1,(t-1)} \mathbf{U}_{J+1}\left(\mathbf{X}_i \mid \mathbf{V}\right)} - \beta \frac{\partial V(\mathbf{p}_i)}{\partial p_i^{j,(t-1)}}.$$

$$q_i^{J+1,(t)} = \frac{U_{J+1}\left(\mathbf{X}_i \mid \mathbf{V}\right)}{\sum_{j=1}^{J} p_i^{j,(t-1)} \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)}\right) + p_i^{J+1,(t-1)} \mathbf{U}_{J+1}\left(\mathbf{X}_i \mid \mathbf{V}\right)} - \beta \frac{\partial V(\mathbf{p}_i)}{\partial p_i^{J+1,(t-1)}},$$

  - Step b: Taking the constraints of $\mathbf{p}_i^{(t-1)}$ into consideration,

$$\mathbf{d}_i^{(t)} = \mathbf{R}_i^{(t)} \mathbf{q}_i^{(t)}.$$

    The projection matrix $\mathbf{R}_i^{(t)}$ is defined as:

$$\mathbf{R}_i^{(t)} \begin{cases} 0 & \text{if } p_i^{j,(t-1)} = 0 \text{ or } p_i^{l,(t-1)} = 0, \text{for all } j \neq l \\ \frac{K_0 - 1}{K_0} & \text{if } j = l \text{ and } p_i^{j,(t-1)} \neq 0 \\ -\frac{1}{K_0} & \text{otherwise} \end{cases},$$

    where K_0 is the number of active constraints.
  - Step c: Set $\alpha_1 = 1$. Compute $\mathbf{p}_i^{(t)} = \mathbf{p}_i^{(t-1)} + \alpha_1 \mathbf{d}_i^{(t)}$.
  - Step d: Compute

$$c_2 = \sum_{j=1}^{J+1} \omega_i^{j,(t)} \log p_i^{j,(t)} - \beta V(\mathbf{p}_i).$$

  - Step e: If $c_2 < c_1$, updating α_1 to $0.5\alpha_1$ and go back to Step c.
- Generalized M Step 2:

$$\boldsymbol{\mu}_j^{(t)} = \frac{1}{\sum_{i=1}^{N} \omega_i^{j,(t)}} \sum_{i=1}^{N} \omega_i^{j,(t)} \mathbf{X}_i \ , \ j = 1, \cdots, J,$$

$$\mathbf{T}_j^{(t)} = \frac{1}{\sum_{i=1}^{N} \omega_i^{j,(t)}} \sum_{i=1}^{N} \omega_i^{j,(t)} \left[\mathbf{X}_i - \boldsymbol{\mu}_j^{(t)}\right] \left[\mathbf{X}_i - \boldsymbol{\mu}_j^{(t)}\right]^T \ , j = 1, \cdots, J.$$

- Check convergence, if the parameter converged, stop the procedure; otherwise repeat E Step and Generalized M Step. Details of assessing convergence have been given in Section 2.3.1.

The conditional expectation at the last iteration will then be used to cluster objects, each object will be grouped into the component with the largest ω_i^j, $1 \leq j \leq J+1$.

8.3.4 Spatially Constrained Finite Mixture Model with Noise Component Summary

So in the spatially constrained finite mixture model with noise points, firstly, I will use the nearest neighbour clutter removal to obtain an initial estimate of noise points; Secondly, I will use the spatial hierarchical clustering to group the non-anomalous points; Lastly, GEM with gradient projection will be applied to model (8.5) with component parameters initialized by the spatial hierarchical clustering and nearest-neighbour clutter removal.

8.4 Spatially Constrained Mixture Model with Prior Terms

An alternative way to deal with the problem caused by singular covariance matrices due to outliers in fitting spatially constrained finite mixture models is adding prior terms to the model to regularize it. We call this type of model the spatially constrained mixture model with prior terms. Specifically, the prior terms used in this thesis are set as the same prior distributions used in model-based clustering with dissimilarities in a Bayesian approach [39] (further details have been given in Section 4.7), and parameters will be estimated by using the GEM gradient projection algorithm.

Assuming all objects are generated from a model containing different multivariate normal distribution components with some set of mixing proportions and we incorporate spatial information into this model, then the probability density function of the i^{th} object with

prior spatial information can be expressed as:

$$f\left(\mathbf{X}_i \mid \boldsymbol{\mu}, \mathbf{T}, \mathbf{p}_i\right) = \left\{\sum_{j=1}^{J} p_i^j \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j, \mathbf{T}_j\right)\right\} \exp\left\{-\beta V(\mathbf{p}_i)\right\},$$

where p_i^j subjects to

$$\sum_{j=1}^{J} p_i^j = 1,$$
$$0 \leq p_i^j \leq 1, j = 1, \cdots, J.$$

The incorporation of prior distributions for $\boldsymbol{\mu}$ and $\mathbf{T}$ gives the full posterior distribution of the i^{th} object:

$$f\left(\mathbf{X}_i \mid \boldsymbol{\mu}, \mathbf{T}, \mathbf{p}_i\right) \propto \left\{\sum_{j=1}^{J} p_i^j \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j, \mathbf{T}_j\right)\right\} \prod_{j=1}^{J} \left\{f(\boldsymbol{\mu}_j \mid \boldsymbol{\mu}_{j0}, \mathbf{T}_j) f(\mathbf{T}_j \mid \alpha, \mathbf{B}_j)\right\} \times \exp\left\{-\beta V(\mathbf{p}_i)\right\}. \tag{8.10}$$

$I(Z_i = j)$ is an indicator variable which indicates whether the i^{th} object belongs to the j^{th} component. The complete data log posterior distribution is expressed as:

$$\begin{aligned}
&\log\left(P\left(\boldsymbol{\mu}, \mathbf{T} \mid \mathbf{X}, \boldsymbol{\mu}, \mathbf{T}, \mathbf{Z}, \mathbf{p}, \boldsymbol{\mu}_{j0}, \alpha, \mathbf{B}\right)\right) \\
&\propto \sum_{i=1}^{N} \left\{\log\left(\prod_{j=1}^{J} \left[p_i^j \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j, \mathbf{T}_j\right)\right]^{I(Z_i=j)} \prod_{j=1}^{J} \left[\phi\left(\boldsymbol{\mu}_j \mid \boldsymbol{\mu}_{j0}, \mathbf{T}_j\right) f\left(\mathbf{T}_j \mid \alpha, \mathbf{B}_j\right)\right]\right)\right\} - \sum_{i=1}^{N} V(\mathbf{p}_i) \\
&= \sum_{i=1}^{N} \left[\sum_{j=1}^{J} I(Z_i = j) \left\{\log(p_i^j) + \log\left(\phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j, \mathbf{T}_j\right)\right)\right\}\right] \\
&+ \sum_{j=1}^{J} \log f\left(\boldsymbol{\mu}_j \mid \boldsymbol{\mu}_{j0}, \mathbf{T}_j\right) + \sum_{j=1}^{J} \log f\left(\mathbf{T}_j \mid \alpha, \mathbf{B}_j\right) - \beta \sum_{i=1}^{N} V(\mathbf{p}_i).
\end{aligned}$$

We assume the prior distributions of $\boldsymbol{\mu}_j$ and $\mathbf{T}_j$ are

$$f\left(\boldsymbol{\mu}_j \mid \boldsymbol{\mu}_{j0}, \mathbf{T}_j\right) \sim \phi\left(\boldsymbol{\mu}_{j0}, \mathbf{T}_j\right),$$

$$f\left(\mathbf{T}_j \mid \alpha, \mathbf{B}_j\right) \sim \mathrm{IW}\left(\alpha, \mathbf{B}_j\right),$$

where $\mathrm{IW}(\cdot)$ represents inverse Wishart distribution, $\alpha = P + 4$ and $\mathbf{B}_j = (\alpha - P - 1)\mathbf{S}_j$ for the reasons given in Section 4.7, where P is the number of dimensions and $\mathbf{S}_j$ is the covariance matrix of the initial j^{th} cluster [87]. $\phi(\cdot)$ denotes the multivariate normal distribution, $\boldsymbol{\mu}_{j0}$ is the group mean vector of the initial j^{th} cluster, which can be estimated by using spatial hierarchical clustering [13].

8.4.1 Parameter Estimation for the Spatially Constrained Finite Mixture Model with Prior Terms

The conditional expectation of the complete data log posterior distribution given the observed data is

$$\begin{aligned}
&Q\left(\boldsymbol{\mu}, \mathbf{T} \mid \boldsymbol{\mu}, \mathbf{T}, \boldsymbol{\mu}_{j0}, \alpha, \mathbf{B}, \mathbf{X}, \mathbf{Z}, \mathbf{p}\right) \\
&= \sum_{i=1}^{N} \left\{ \sum_{j=1}^{J} \mathrm{E}\left[Z_i = j \mid \mathbf{X}_i, \boldsymbol{\mu}_j, \mathbf{T}_j\right] \left[\log(p_i^j) + \log \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j, \mathbf{T}_j\right)\right] \right\} \\
&+ \sum_{j=1}^{J} \left(\log f\left(\boldsymbol{\mu}_j \mid \boldsymbol{\mu}_{j0}, \mathbf{T}_j\right)\right) + \sum_{j=1}^{J} \left(\log f\left(\mathbf{T}_j \mid \alpha, \mathbf{B}_j\right)\right) - \beta \sum_{i=1}^{N} V(\mathbf{p}_i).
\end{aligned}$$

- E Step: For all objects $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, J$.

$$\mathrm{E}\left(I(Z_i = j) \mid \mathbf{X}_i, \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)}\right) \equiv \omega_i^{j,(t)} = \frac{p_i^{j,(t-1)} \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)}\right)}{\sum_{k=1}^{J} p_i^{k,(t-1)} \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_k^{(t-1)}, \mathbf{T}_k^{(t-1)}\right)}. \tag{8.11}$$

- Generalized M Step 1: For each object i, do Step a to Step e.
  - Step a: For all points

$$c_1 = \sum_{j=1}^{J} \omega_i^{j,(t)} \log p_i^{j,(t-1)} - \beta V(\mathbf{p}_i), \tag{8.12}$$

$$\begin{aligned}
q_i^{j,(t)} &= \frac{\partial Q\left(\boldsymbol{\mu}, \mathbf{T} \mid \mathbf{X}_i, \boldsymbol{\mu}^{(t-1)}, \mathbf{T}^{(t-1)}, \boldsymbol{\mu}_{j0}, \alpha, \mathbf{B}_j, \mathbf{Z}, \mathbf{p}\right)}{\partial p_i^{j,(t-1)}} \\
&= \frac{\phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)}\right)}{\sum_{j=1}^{J} \phi\left(\mathbf{X}_i \mid \boldsymbol{\mu}_j^{(t-1)}, \mathbf{T}_j^{(t-1)}\right)} - \beta \frac{\partial V(\mathbf{p}_i)}{\partial p_i^{j,(t-1)}}.
\end{aligned} \tag{8.13}$$

  - Step b: Taking the constraints of $\mathbf{p}_i^{(t)}$ into consideration,

$$\mathbf{d}_i^{(t)} = \mathbf{R}_i^{(t)} \mathbf{q}_i^{(t)}.$$

The definition for $\mathbf{R}_i^{(t)}$ is

$$\mathbf{R}_i^{(t)} \begin{cases} 0 & \text{if } p_i^{j,(t-1)} = 0 \text{ or } p_i^{l,(t-1)} = 0, \text{for all } j \neq l \\ \frac{K_0-1}{K_0} & \text{if } j = l \text{ and } p_i^{j,(t-1)} \neq 0 \\ -\frac{1}{K_0} & \text{otherwise} \end{cases},$$

where K_0 is the number of active constraints.

- Step c: Set $\alpha_1 = 1$. Compute

$$\mathbf{p}_i^{(t)} = \mathbf{p}_i^{(t-1)} + \alpha_1 \mathbf{d}_i^{(t)}. \tag{8.14}$$

- Step d: Compute

$$c_2 = \sum_{j=1}^{J} \omega_i^{j,(t)} \log p_i^{j,(t)} - \beta V(\mathbf{p}_i). \tag{8.15}$$

- Step e: If $c_2 < c_1$, updating α_1 to $0.5\alpha_1$ and go back to Step 2c.

- Generalized M Step 2: Compute

$$\boldsymbol{\mu}_j^{(t)} = \frac{\sum_{i=1}^{N} \omega_i^{j,(t)} \mathbf{X}_i + \boldsymbol{\mu}_{j0}}{1 + \sum_i^N \omega_i^{j,(t)}}, \tag{8.16}$$

$$\mathbf{T}_j^{(t)} = \frac{\sum_{i=1}^{N} \omega_i^{j,(t)} \left(\mathbf{X}_i - \boldsymbol{\mu}_j\right)\left(\mathbf{X}_i - \boldsymbol{\mu}_j\right)^T + \left(\boldsymbol{\mu}_j - \boldsymbol{\mu}_{j0}\right)\left(\boldsymbol{\mu}_j - \boldsymbol{\mu}_{j0}\right)^T + \mathbf{B}_j}{\sum_i^N \omega_i^{j,(t)} + \alpha + P + 2}. \tag{8.17}$$

- Check convergence, if the parameter converged, stop the procedure; otherwise repeat E Step and Generalized M Steps. Details of assessing convergence have been given in section 2.3.1.

The conditional expectation at the last iteration will then be used to cluster objects, each object will be grouped into the component with the largest ω_i^j, $1 \leq j \leq J$.

8.5 Spatially Constrained Finite Mixture Model with Prior Terms Summary

So in the spatially constrained finite mixture model with prior terms, firstly, I will use spatial hierarchical clustering to group the data; then, GEM with gradient projection

will be applied to model (8.10) with component parameters initialized by the spatial hierarchical clustering.

8.6 Simulations

In this section, I will use simulations generated from different scenarios to compare the performance of different spatial clustering techniques. The factorial design will be the same one as used in Chapter 7.

8.6.1 Spatially Constrained Mixture Model Examples

All different scenarios will be repeated 100 times. Decision on K in the spatially constrained finite mixture model with noise points can be obtained from Figure 8.7.

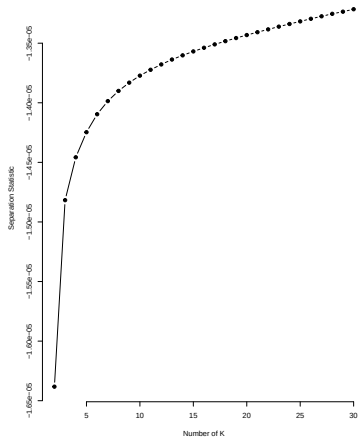


FIGURE 8.7: K Selection in Nearest-Neighbour Clutter Removal Method in Simulated Data Set 1

We can see that K is always increasing until $K = 10$ it starts to level off, so $K = 10$ is used in the nearest-neighbour clutter removal method to identify the noise points. The initial parameter values will be estimated from the spatial hierarchical clustering results. The number of main clusters in the spatially constrained finite mixture model with noise points and the number of clusters in the spatially constrained finite mixture model with prior terms will decided by BIC (introduced in Section 4.9.2). In each scenario, spatially

constrained finite mixture models with different number of clusters will be compared, the one with the minimal BIC will be used to model the data.

There are two statistics will be used to measure the identification of noise points. True Positive Rate (TPR) is the ratio of the number of the estimated noise points that are true noise points to the total number of true noise points.

$$\text{TPR} = \frac{|\text{estimated outliers} \cap \text{actual outliers}|}{|\text{actual outliers}|}. \tag{8.18}$$

If all true noise points are detected, then TPR will be 1. True Discovery Rate (TDR) is the ratio of the number of the estimated noise points that are true noise points to the total number of estimated noise points.

$$\text{TDR} = \frac{|\text{estimated outliers} \cap \text{actual outliers}|}{|\text{estimated outliers}|}. \tag{8.19}$$

If the estimated noise points are all true outliers, then TDR will be 1. Comparing TPR and TDR allow us to assess whether models overestimate the number of outliers or not. For example, if TPR is closer to 1, but TDR is relatively small, e.g. less than 0.5, which means there are more estimated outliers than the actual number of outliers. If TPR is relatively small, but TDR is large, which means the method underestimates the number of outliers.

From the simulation results in Tables 8.2 to 8.5, we can see that, in the sparse distribution of areas or scenarios with more outliers, the spatially constrained finite mixture model with noise distribution obtained higher average ARI in grouping the areal units. Specifically, when variances are large or the difference between two distributions is not very different, spatially constrained finite mixture model with noise distribution did better than the other two methods. However, the spatially constrained finite mixture models with noise distribution tend to form more clusters than the other two methods do. From Table 8.5, we can see that spatially constrained finite mixture model detected more than 18 noise points, which means for data with large variations will get more number of noise points. It has therefore high average TPRs and low average TDRs. Comparing all these four clustering methods we can see that the both spatially constrained finite mixture models have similar performance across all these scenarios and are less affected by the variance (comparing distributions 1 and 2 or distributions 3 and 4). On the other hand, Chameleon spatial hierarchical clustering is more sensitive to the variance in comparing the average ARI in Tables 8.4 and 8.5.

On the other hand, the spatially constrained finite mixture model with prior distribution obtained slightly higher average ARI in scenarios with dense locations or blurred variances

TABLE 8.2: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 1 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.876 (0.028)	4.100 (0.403)	3.700 (0.822)	7.800	6 (4+2)	1.000 (0.000)	0.652 (0.263)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.857 (0.043)	-	-	6.280 (0.361)	6 (4+2)	-	-
Spatial Hierarchical Clustering	0.657 (0.075)	-	-	8.830 (1.390)	6 (4+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.777 (0.051)	-	-	4.150 (0.230)	6 (4+2)	-	-

[a] The simulation results for locations in Figures 7.11(b), 7.12, 7.13 and 7.14 are shown in Appendix A.9

TABLE 8.3: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 2 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.735 (0.083)	5.360 (0.337)	2.700 (0.988)	8.060	6 (4+2)	1.000 (0.000)	0.821 (0.232)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.717 (0.095)	-	-	6.540 (1.072)	6 (4+2)	-	-
Spatial Hierarchical Clustering	0.686 (0.118)	-	-	10.870 (1.320)	6 (4+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.632 (0.063)	-	-	7.540 (0.370)	6 (4+2)	-	-

[a] The simulation results for locations in Figures 7.11(b), 7.12, 7.13 and 7.14 are shown in Appendix A.9

TABLE 8.4: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 3 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.897 (0.005)	4.000 (0.000)	2.230 (0.436)	6.230	6 (4+2)	1.000 (0.000)	0.922 (0.143)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.863 (0.007)	-	-	6.340 (0.170)	6 (4+2)	-	-
Spatial Hierarchical Clustering	0.683 (0.081)	-	-	8.930 (1.420)	6 (4+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.936 (0.023)	-	-	4.880 (0.190)	6 (4+2)	-	-

[a] The simulation results for locations in Figures 7.11(b), 7.12, 7.13 and 7.14 are shown in Appendix A.9

TABLE 8.5: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 4 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.641 (0.081)	5.260 (0.672)	2.660 (0.779)	7.920	6(4+2)	1.000 (0.000)	0.131 (0.108)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.698 (0.108)	-	-	8.130 (1.178)	6 (4+2)	-	-
Spatial Hierarchical Clustering	0.587 (0.085)	-	-	10.670 (1.470)	6 (4+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.602 (0.058)	-	-	7.850 (0.440)	6 (4+2)	-	-

[a] The simulation results for locations in Figures 7.11(b), 7.12, 7.13 and 7.14 are shown in Appendix A.9

(e.g Figure 7.13(a)).

In the scenarios discussed above it was assumed it is possible for areal units from the same cluster have independent variables/dimensions. However, dimensions of the areal units from the same cluster to be dependent, which means the off-diagonals of the covariance matrix would be non-zero. I will also simulate from scenarios where the variances have non-equal values. Comparing different scenarios and distributions, the scenario in Figure 7.11(a) gives better results in both the number of clusters and ARI. In addition, when the mean levels between two groups are more different, the simulation results will be higher in ARI and number of clusters is closer to the actual number of clusters, so I will compare the performances about the dependent dimensions and different diagonals in these scenario (in Figure 7.11(a), distributions 3 and 4). The simulation results of this scenario will be simulated from distributions 5, 6, 7, 8, 9, 10 introduced in Section 7.3.10.

Simulation results displayed in Tables 8.6 to 8.9 show the results in the same where within group dimensions are simulated as dependent. We can see that when the dimensions are dependent, particularly, when the dependence gets stronger, the clustering result will be worse with lower ARI due to the mismatch between the true simulated distribution and the modelling assumption. Comparing across all these four clustering methods, we can see that the two spatially finite mixture models have similar ARI and Chameleon spatial hierarchical clustering generally gives better clustering results with higher ARI except for the results in Tables 8.9 and 8.8. When the simulated data are variables/dimensions dependent, the finite mixture models will form more noise points and less clusters, so the both the average TPR and TDR are lower. The finite mixture model with prior will form more clusters when the variables/dimensions are dependent. Tables 8.10 to 8.11 show the simulation results when the variances are simulated to be different. It is easy to see that the different variances do not appear to affect the clustering results very much, neither the number of clusters nor the average ARI changed a lot. This conclusion is consist with the covariance setting in the spatially constrained mixture models.

In conclusion, the spatially constrained finite mixture model with noise distribution performs well in sparse distribution of areas or multiplex environment (more clusters, heterogeneous objects), while the spatially constrained finite mixture model with prior distribution and the spatial hierarchical clustering are good at dealing with monotonous environment (fewer clusters and homogeneous objects).

TABLE 8.6: Summary of ARI, $G(H+J)$, TPR and TDR of Dependent Data from Distribution Set 5 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.885 (0.013)	3.870 (0.120)	2.350 (0.556)	6.220	6(4+2)	0.962 (0.061)	0.851 (0.108)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.829 (0.008)	-	-	6.550 (0.235)	6 (4+2)	-	-
Spatial Hierarchical Clustering	0.675 (0.088)	-	-	8.640 (1.270)	6 (4+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.930 (0.028)	-	-	4.780 (0.170)	6 (4+2)	-	-

TABLE 8.7: Summary of ARI, $G(H+J)$, TPR and TDR of Dependent Data from Distribution Set 6 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.878 (0.019)	3.800 (0.155)	2.440 (0.556)	6.240	6(4+2)	0.925 (0.085)	0.835 (0.122)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.806 (0.014)	-	-	6.640 (0.220)	6 (4+2)	-	-
Spatial Hierarchical Clustering	0.672 (0.074)	-	-	8.500 (1.820)	6 (4+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.911 (0.034)	-	-	4.630 (0.280)	6 (4+2)	-	-

TABLE 8.8: Summary of ARI, $G(H+J)$, TPR and TDR of Dependent Data from Distribution Set 7 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.625 (0.075)	5.180 (0.830)	2.810 (1.856)	7.990	6(4+2)	0.844 (0.075)	0.123 (0.104)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.675 (0.086)	-	-	8.330 (1.195)	6 (4+2)	-	-
Spatial Hierarchical Clustering	0.540 (0.078)	-	-	10.480 (1.630)	6 (4+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.589 (0.095)	-	-	7.540 (0.780)	6 (4+2)	-	-

TABLE 8.9: Summary of ARI, $G(H+J)$, TPR and TDR of Dependent Data from Distribution Set 8 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.610 (0.084)	5.040 (0.820)	3.050 (2.460)	8.090	6(4+2)	0.835 (0.038)	0.114 (0.136)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.658 (0.106)	-	-	8.560 (1.205)	6 (4+2)	-	-
Spatial Hierarchical Clustering	0.536 (0.085)	-	-	10.210 (1.890)	6 (4+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.582 (0.058)	-	-	7.420 (0.530)	6 (4+2)	-	-

TABLE 8.10: Summary of ARI, $G(H+J)$, TPR and TDR of Different Variances Data from Distribution Set 9 Located in Figure 7.11(a)

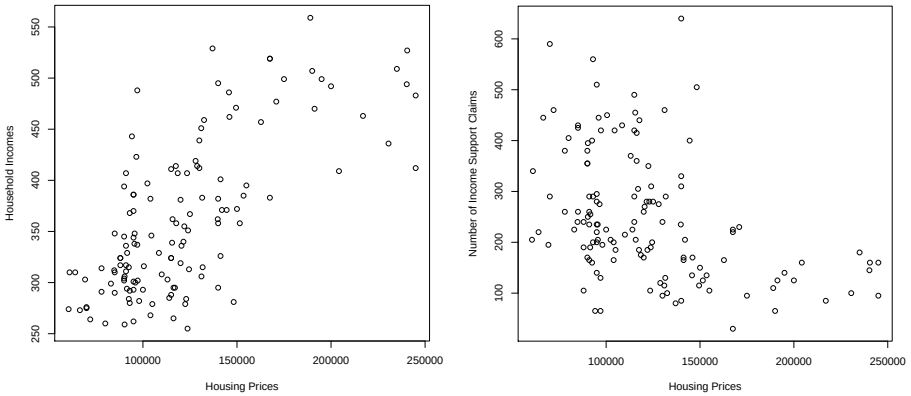
Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.893 (0.007)	4.000 (0.000)	2.180 (0.409)	6.180	6 (4+2)	1.000 (0.000)	0.925 (0.084)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.869 (0.012)	-	-	6.265 (0.200)	6 (4+2)	-	-
Spatial Hierarchical Clustering	0.676 (0.086)	-	-	8.76 (1.31)	6 (4+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.921 (0.083)	-	-	4.67 (0.17)	6 (4+2)	-	-

TABLE 8.11: Summary of ARI, $G(H+J)$, TPR and TDR of Different Variances Data from Distribution Set 10 Located in Figure 7.11(a)

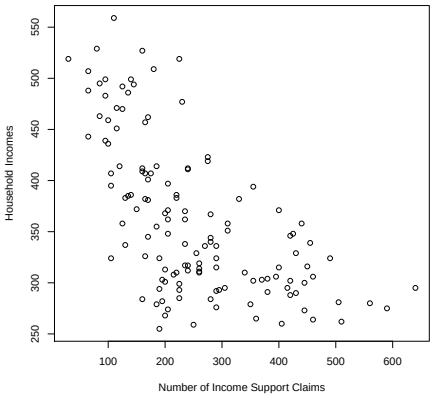
Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.637 (0.079)	5.280 (0.780)	2.710 (2.410)	7.990	6 (4+2)	1.000 (0.000)	0.917 (0.045)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.703 (0.085)	-	-	8.250 (1.210)	6 (4+2)	-	-
Spatial Hierarchical Clustering	0.542 (0.072)	-	-	10.65 (1.39)	6 (4+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.593 (0.089)	-	-	7.51 (0.53)	6 (4+2)	-	-

8.7 Spatially Constrained Finite Mixture Models Applied to Glasgow Housing Data

The variations in household gross income are strongly related to the areal housing prices. The rich with higher income are more likely to purchase properties in areas with higher house prices, while, the poor with lower income are less likely to afford the properties in higher price areas. The aim of this application is to find the areas with similar household incomes and house prices. In addition, a certain percentage of households inherited their properties, so they may not have high incomes but are living in richer areas. The data have been introduced in Chapter 5, which is a three-dimensional data set, with one dimension representing the median house prices in all intermediate zones in 2010, one dimension representing the weekly median gross household income in all intermediate zones in 2010 (Source: http://statistics.gov.scot), the third dimension representing the number of over 60 people income support claims. All data are measured in GBP (£).



(a) The First and the Second Dimensions (b) The First and the Third Dimensions



(c) The Second and the Third Dimensions

FIGURE 8.8: Data Configuration

From Figure 8.8(a) we can tell that most of the objects are lying close to a diagonal line, which indicates that there is a positive relationship between household gross income and the areas' house prices. When the household gross income increases, they tend to choose properties in the areas with higher house prices. There are a large proportion of observations lying at the left bottom corner of the space and they are closer to the diagonal line, while there are only a small proportion of points scattered at the right top corner. Specifically, fewer households with lower gross income can afford properties in the areas with median house prices greater than 150,000, but people with higher gross household income are more flexible in choosing their living areas. So it is sensible that the city have two main clusters with one of them containing households with low gross income and living in low house price areas, the other cluster involves households with high gross income and living in high house price areas. Both Figures 8.8(b) and 8.8(c) shows the negative relationships between variables. However, the negative relationship

(the pearson correlation coefficient is -0.627) between household incomes and the number of income supports in Figure 8.8(c) is slightly stronger than the relationship between the number of income supports and the house prices (the pearson correlation coefficient is -0.413) in Figure 8.8(b).

In the spatially constrained finite mixture model with noise points, the decision on K is made using the plot in Figure 8.9,

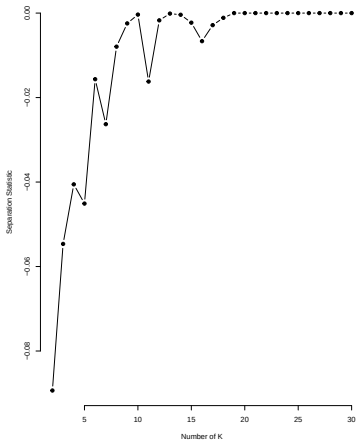


FIGURE 8.9: K Selection in Nearest-Neighbour Clutter Removal Method in Glasgow Housing Market Data

We can see that K is always increasing until $K = 9$ it starts to level off, so $K = 9$ is chosen in nearest-neighbour clutter removal method to identify the noise points. The initial parameter values will be obtained from the spatial hierarchical clustering results according to the empirical research [87].

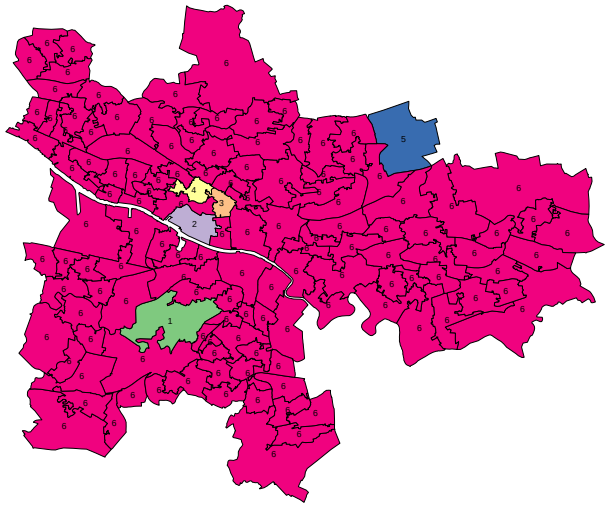


FIGURE 8.10: Noise Points of Household Income and Over 60 Claims in 2010 Glasgow Intermediate Zones

According to the results of nearest neighbour clutter, there are 5 areas (29,78,82,89,112) identified as noise points, which are labeled as different colours in Figure 8.10. All the non-anomalous points are labeled in pink in Figure 8.10. The possible number of non-anomalous groups will be at least two, as Glasgow is mainly divided by the River Clyde into two parts if no manual connections, such as bridges and boats, etc, are taken into consideration. In addition, the optimal number of clusters from the elbow plot in Figure 6.4) by using spatially hierarchical clustering is close to 35. So the range of number of clusters will be chosen from 20 (the non-anomalous groups), for which BIC plot is shown in Figure 8.11.

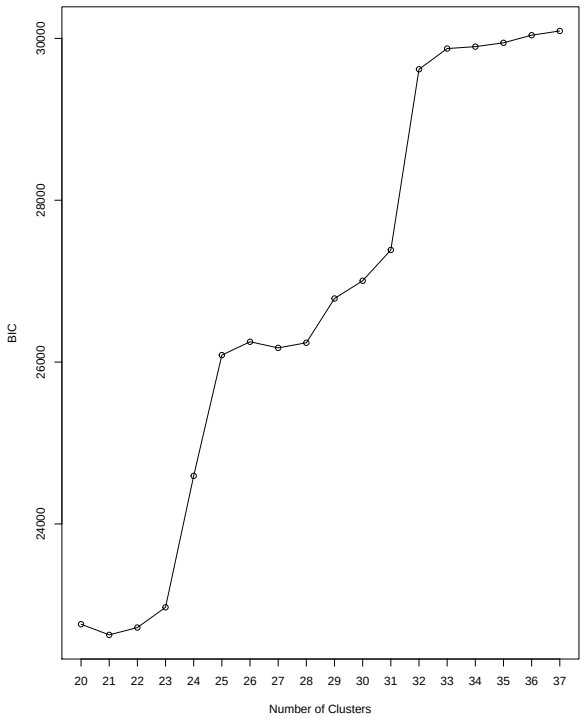


FIGURE 8.11: BIC of Finite Mixture Noise Clustering, Household Income and Over 60 Claims in 2010 Glasgow Intermediate Zones

Figure 8.11 shows that when the number of clusters is 21, the clustering will have the minimal BIC and the corresponding clustering is shown in Figure 8.12. BIC keeps increasing after 27, so there are no more BIC comparison after $G = 37$.

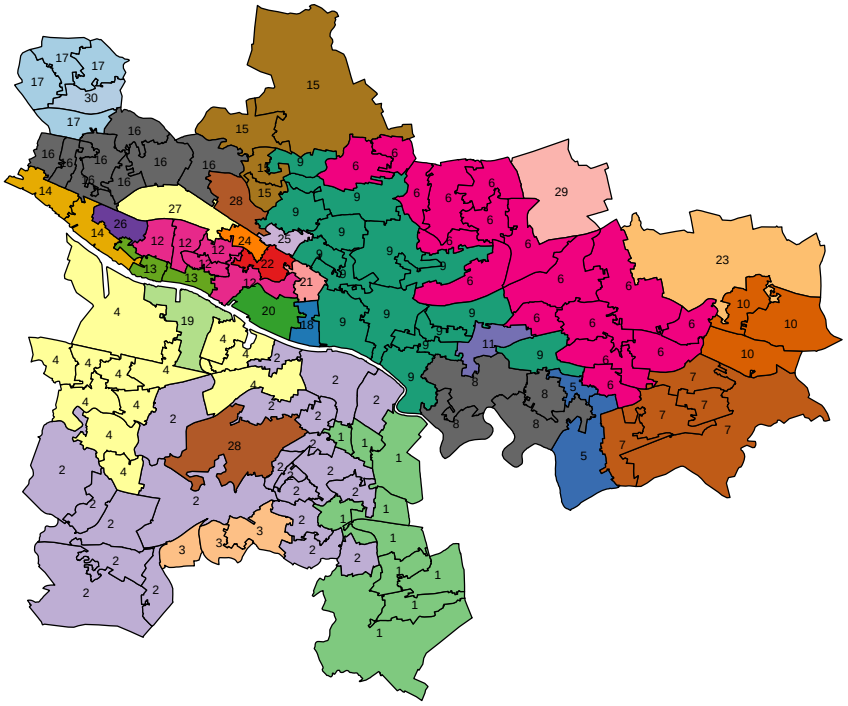


FIGURE 8.12: Finite Mixture Noise Clustering of Housing Price, Household Income and Over 60 Claims in 2010 Glasgow Intermediate Zones

The number of non-anomalous clusters is set as 21, the initial anomalous points are identified by nearest neighbour clutter, the formed clustering is shown in Figure 8.12. The clustering has 30 clusters with more singleton clusters than the initial number of anomalous points. In addition, all the initial anomalous points are also treated as singleton clusters in Figure 8.12. It is also interesting to see that the West End is divided into more clusters than the East End.

TABLE 8.12: Spatially Constrained Finite Mixture Model with Noise Term Summary for Glasgow Housing Data

	Average Median House Price (£)	Average Gross Household Income Per Week (£)	No. Claimts Over 60
1	108249.50	357.75	247.50
2	94683.33	355.11	259.44
3	124600.45	397.45	200.23
4	125333.33	385.00	190.00
5	93538.00	339.33	281.25
6	149450.00	448.00	252.50
7	237750.00	518.00	170.00
8	125475.80	420.20	179.00
9	84517.00	285.25	411.25
10	94029.69	302.56	344.69
11	125464.67	318.00	343.33
12	87166.67	371.00	168.33
13	70000.00	276.00	290.00
14	153500.00	385.00	135.00
15	66500.00	273.00	445.00
16	141250.00	401.00	170.00
17	178150.00	444.60	148.00
18	204225.00	409.00	160.00
19	155000.00	395.00	105.00
20	230564.00	436.00	100.00
21	130424.50	349.60	258.00
22	151500.00	358.00	125.00
23	154747.50	447.50	190.00
24	245000.00	483.00	95.00
25	217000.00	463.00	85.00
26	96000.00	319.00	165.00
27	245000.00	412.00	160.00
28	240250.00	494.00	145.00
29	96500.00	317.75	177.50
30	69250.00	303.00	195.00

Table 8.12 further explains the housing prices, household incomes and the over 60 low income claims situations in different clusters. Clusters 24 (Dowanhill) and 27 (Kelvinside and Jordanhill) both have very similar high housing prices. However, household income

in Kelvinside and Jordanhill is much lower than those in Dowanhill, the same group of areas seem to be unaffordable for households in Kelvinside and Jordanhill as people with a similar income are more likely to live in cluster 18 (Anderston) for which housing price is around 50,000 cheaper than where they are living. It is interesting to see that there are more people over 60 claim income support in Kelvinside and Jordanhill, which can be explained by there are more people in Kelvinside and Jordanhill living in properties inherited from families. Generally speaking, areas with lower housing prices are more likely to have higher number of income support claims. The areas with the lowest housing price (cluster 9) have almost three times number of income support claims than those in the highest housing price areas (clusters 24 and 27).

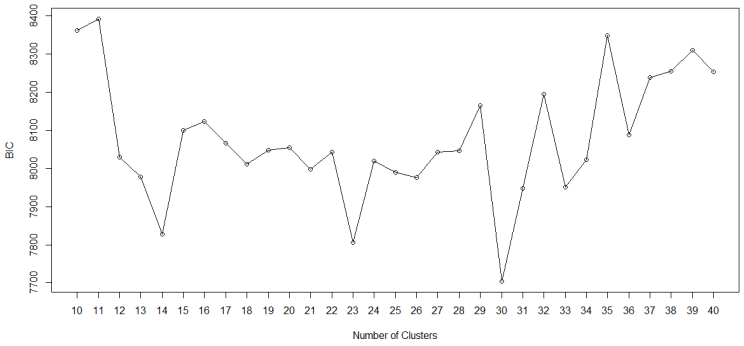


FIGURE 8.13: BIC of Finite Mixture Prior Clustering, Household Income and Over 60 Claimts in 2010 Glasgow Intermediate Zones

From the BIC plot in Figure 8.13, we can see that BIC gets slightly smaller between $G = 11$ to $G = 34$ and increases after then. The minimal BIC is achieved when the number of clusters is 30, for which clustering is shown in Figure 8.14.

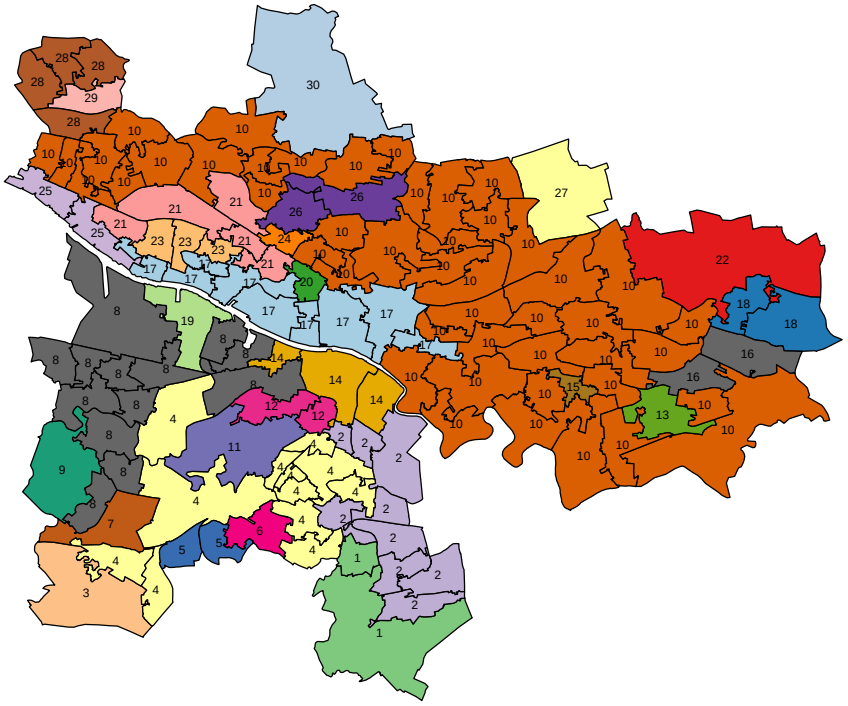


FIGURE 8.14: Finite Mixture Prior Clustering of Housing Price ($G = 30$), Household Income and Over 60 Claims in 2010 Glasgow Intermediate Zones

Figure 8.14 shows the clustering of Glasgow based on the housing Price, household Income and over 60 claims in 2010 Glasgow intermediate zones. The West End has fewer clusters than the clustering in Figure 8.12. However, on the south side of River Clyde, there are more clusters identified by spatially constrained finite mixture model with priors. Further details of each cluster can be found in Table 8.13.

TABLE 8.13: Spatially Constrained Finite Mixture Model with Prior Term Summary for Glasgow Housing Data

	Average Median House Price (£)	Average Gross Household Income Per Week (£)	No. Claimts Over 60
1	167500.00	519.00	127.50
2	94683.33	355.11	259.44
3	88000.00	324.00	105.00
4	126148.75	402.83	186.67
5	90500.00	328.00	215.00
6	195000.00	499.00	140.00
7	61000.00	310.00	340.00
8	92881.23	337.08	279.62
9	149500.00	471.00	115.00
10	105337.34	319.19	316.49
11	240500.00	527.00	160.00
12	154375.00	459.50	167.50
13	175000.00	499.00	95.00
14	122558.33	352.33	326.67
15	170900.00	477.00	230.00
16	138500.00	512.00	82.50
17	143919.44	357.78	244.44
18	62250.00	292.00	212.50
19	66500.00	273.00	445.00
20	204225.00	409.00	160.00
21	239162.80	466.80	136.00
22	151500.00	358.00	125.00
23	193750.00	489.67	105.00
24	217000.00	463.00	85.00
25	96000.00	319.00	165.00
26	144125.00	319.50	407.50
27	189000.00	559.00	110.00
28	96500.00	317.75	177.50
29	69250.00	303.00	195.00
30	150000.00	372.00	150.00

Generally speaking, the higher the housing price is, the higher the household income and the lower the number of income support claimants will be. By comparing Tables 8.12 and 8.13, we can see that the average median house prices in Table 8.13 shows

more difference among clusters, there are hardly any pairs of clusters with a very similar housing prices, but there are similarities in average median housing prices between pairs of areas can be seen in Table 8.12, such as clusters 24 and 27, clusters 26 and 29, clusters 5 and 10. It is also interesting to see that cluster 13 (Mount Vernon North and Sandyhills) in Figure 8.14 has a lower average housing price which is around £175,000, but it does not have many income support claims, almost as few as the claims in richer areas, such as cluster 24 (North Kelvin). In addition, the household incomes in both Mount Vernon North and Sandyhills and North Kelvin are much higher than the other areas, so this difference may be explained by people's preference and it fails to be identified in Table 8.12.

TABLE 8.14: ARI for Pairs of Spatial Clusterings for Two-dimensional Glasgow Housing Market

	Spatial Hierarchical Clustering	Spatially Constrained Finite Mixture Model with Noise Points	Spatially Constrained Finite Mixture Model with Prior Terms
Spatial Hierarchical Clustering	1.000	0.5071	0.6307
Spatially Constrained Finite Mixture Model with Noise Points	-	1.000	0.3321
Spatially Constrained Finite Mixture Model with Prior Terms	-	-	1.000

From Table 8.14, we can see that spatial hierarchical clustering has a higher similarity with both spatially constrained finite mixture models, but the similarity between two spatially constrained finite mixture models is smaller. The difference in ARI can be caused by the different clustering in the West End, the northeast and southeast parts as the spatially constrained finite mixture model with prior generated more clusters in the West End and the northeast, but fewer clusters in the southeast part than the spatially constrained finite mixture model with prior terms.

8.8 Summary of Spatially Constrained Mixture Model

Generally speaking, the spatially constrained finite mixture model with noise points tends to form clusterings with a higher number of clusters and more areas are likely to be treated as anomalous points. From the simulation results obtained in Chapter 8, we can see that the clustering results achieved from the spatially constrained finite mixture model with noise points are less affected by the number of noise points and variances than the spatially constrained finite mixture model with prior terms. The spatially constrained finite mixture model with prior terms is good at clustering data with condensed distributions of areas (scenarios in Figure 7.11) and small variances. However, for the same dense distribution of areas and small variances, the spatial hierarchical clustering does slightly better than the spatially constrained finite mixture model with priors. In the application, the spatially constrained finite mixture model with noise points identifies more difference caused by low income support claims or the source of their properties. This is because from Table 8.12 we can see that there are some households with lower incomes but living in a relatively unaffordable area compared to their incomes. However, in this application, the spatially constrained finite mixture model with prior terms identifies more clusters than with noise points. Table 8.13 shows some households with higher incomes but still living in a relatively cheaper housing price areas.

Chapter 9

Spatially Constrained Bayesian Model-based Clustering with Dissimilarities

In Chapter 8, I introduced the GEM algorithm using gradient projection to estimate the parameters in spatially constrained finite mixture models. In this Chapter, I will explore, given dissimilarity data, how to use a Bayesian approach to estimate parameters in spatially constrained finite mixture models. Given a likelihood with latent variables, GEM explores the parameter values by increasing the expected complete data likelihood function over the previous step. For the same starting value(s), the GEM algorithm often takes less time to converge than the corresponding Bayesian approach does. However, GEM can only provide point estimates of the parameters of interest and does not quantify the full posterior distributions. McMC is a simulation method, given a likelihood, with or without latent variables, and prior information, which produces samples from the posterior distribution generated using either Gibbs sampling or Metropolis Hastings.

The model-based clustering with dissimilarities model using a Bayesian approach proposed by Man-Suk Oh and Adrian E. Raftery [87] has been introduced in Section 4.7. However, this model fails to enforce spatial contiguity between areal units within clusters. So in this chapter, I will extend the Bayesian model-based clustering with dissimilarities model to the spatially constrained Bayesian model-based clustering with dissimilarities model by incorporating spatial information. The spatial term will be in the same form as the one used in Chapter 8.

9.1 Spatially Constrained Bayesian Model-based Clustering with Dissimilarities

In Section 9.1.1, I will review the Bayesian model-based clustering with dissimilarities model. The introduction of a spatially constrained Bayesian model-based clustering with dissimilarities and the related parameter estimation problem will be discussed in Section 9.1.2, decisions about the hyperparameters and proposal distributions will be covered in Section 9.1.3.

9.1.1 Bayesian Model-based Clustering with Dissimilarities Review

Bayesian model-based clustering with dissimilarities was proposed for grouping objects on the basis of dissimilarity data in a Bayesian way. It combines two ideas, it uses dissimilarities to estimate the latent object positions in a Euclidean space and also assumes the positions are generated from a mixture of multivariate Gaussian distributions and each Gaussian distribution corresponds to one cluster component.

Let δ_{ik} denote the calculated dissimilarity between objects i and k in a P dimensional Euclidean space, $\delta_{ik} = \sqrt{\sum_{p=1}^{P} \left(X_{ip} - X_{kp}\right)^2}$, where objects i and k are $\mathbf{X}_i = (X_{i1}, \cdots, X_{iP})$ and $\mathbf{X}_k = (X_{k1}, \cdots, X_{kP})$ respectively. The relationship between the observed dissimilarities $(\mathbf{D}(d_{ik}))$ and calculated dissimilarities can be expressed as
$d_{ik} = \delta_{ik} + \varepsilon_{ik}$.

All distances have non-negative values. Thus, given the calculated dissimilarity δ_{ik}, the observed dissimilarity d_{ik} is assumed to follow a truncated Gaussian distribution with mean equal to the calculated dissimilarity δ_{ik}.

$$d_{ik} \sim \mathrm{N}\left(\delta_{ik}, \sigma^2\right) \mathrm{I}\left(d_{ik} > 0\right) \text{ for all } i \neq k,\ i, k = 1, \cdots, N.$$

For a given number of clusters J, we assume that the i^{th} object's configuration $\mathbf{X}_i$ is generated from a J multivariate Gaussian components mixture model,

$$\mathbf{X}_i \sim \sum_{j=1}^{J} p^j \phi\left(\boldsymbol{\mu}_j, \mathbf{T}_j\right), i = 1, \cdots, N,$$

where p^j denotes the proportion of the population in the j^{th} component. The prior distributions [87] for the model parameters are listed as follows:

$$\begin{aligned}
\sigma^2 &\sim \mathrm{IG}\left(a, b\right), \\
\left(p^1, \cdots, p^J\right) &\sim \mathrm{Dirichlet}\left(1, \cdots, 1\right), \\
\boldsymbol{\mu}_j &\sim \phi\left(\boldsymbol{\mu}_{j0}, \mathbf{T}_j\right), j = 1, \cdots, J, \\
\mathbf{T}_j &\sim \mathrm{IW}\left(\alpha, \mathbf{B}_j\right), j = 1, \cdots, J.
\end{aligned} \tag{9.1}$$

The setting about a, b, $\boldsymbol{\mu}_{j0}$, α and $\mathbf{B}_j$ have been discussed in Section 4.7. However, for areal data, objects assigned to the same cluster have to be geographically connected or their membership groups geographically connected. So in order to meet this requirement of grouping areal data, I will combine the idea of incorporating spatial information introduced in Chapter 8 with the model-based clustering with dissimilarities model to deal with the spatial contiguity in clustering issue.

9.1.2 Parameter Estimation for the Spatially Constrained Bayesian Model-based Clustering with Dissimilarities

The likelihood function of the positions given the dissimilarities can be expressed as

$$\mathcal{L}\left(\mathbf{X}, \sigma^2; \mathbf{D}\right) \propto \left(\sigma^2\right)^{-m/2} \exp\left\{-\frac{\mathrm{SSR}}{2\sigma^2} - \sum_{i=1}^{N-1}\sum_{k=i+1}^{N} \log\left(\Phi\left(\frac{\delta_{ik}}{\sigma}\right)\right)\right\},$$

where $m = N\left(N-1\right)/2$ and $\mathrm{SSR} = \sum\limits_{i=1}^{N-1}\sum\limits_{k=i+1}^{N}\left(d_{ik} - \delta_{ik}\right)^2$.

We assume objects' configurations $\mathbf{X}$ are generated from a J multivariate Gaussian components mixture model and $I(Z_i = j)$ denotes an indicator variable which indicates whether the i^{th} object belongs to the j^{th} component, then the complete data likelihood can be updated to

$$\begin{aligned}
&\mathcal{L}\left(\mathbf{X}, \mathbf{Z}, \sigma^2, \boldsymbol{\mu}, \mathbf{T}, \mathbf{p}; \mathbf{D}\right) f(\mathbf{X}) \\
&\propto \left(\sigma^2\right)^{-m/2} \exp\left\{-\frac{\mathrm{SSR}}{2\sigma^2} - \sum_{i=1}^{N-1}\sum_{k=i+1}^{N} \log\left(\Phi\left(\frac{\delta_{ik}}{\sigma}\right)\right)\right\} \\
&\times \prod_{i=1}^{N}\prod_{j=1}^{J}\left[p^j \left|\mathbf{T}_j\right|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{X}_i - \boldsymbol{\mu}_j)^T \mathbf{T}_j^{-1}\left(\mathbf{X}_i - \boldsymbol{\mu}_j\right)\right\}\right]^{I(Z_i=j)}.
\end{aligned} \tag{9.2}$$

The novelty of the spatially constrained Bayesian model-based clustering with dissimilarities model is that it takes the spatial information into account by incorporating a Gibbs MRF based prior of the probability of the i^{th} object, $\mathbf{p}_i$, into the model (detailed in Section 8.1). The Gibbs MRF based prior of $\mathbf{p}_i$ is expressed as

$$f\left(\mathbf{p}\right) = \frac{1}{C}\exp\left(-\beta\sum_{i=1}^{N}V\left(\mathbf{p}_i\right)\right),$$

where $V\left(\mathbf{p}_i\right) = \sum_{\forall m:\mathbf{W}_{i,m}=1} g\left(u_{i,m}\right)$, which denotes the adjacency information of the i^{th} object. The selected function of $g\left(u_{i,\mathbf{m}}\right)$ is set to be

$$g\left(u_{i,m}\right) = \left(1+u_{i,m}^{-1}\right)^{-1} = \frac{u_{i,m}}{u_{i,m}+1},$$

where

$$u_{i,m} = ||\mathbf{p}_i - \mathbf{p}_m||^2 = \sum_{j=1}^{J}\left(p_i^j - p_m^j\right)^2,$$

is the image storage model. After incorporating spatial information into (9.2), we extend the probability vector $\mathbf{p} = \left(p^1,\cdots,p^J\right)$ in (9.2) to $\mathbf{p} = \left(\mathbf{p}_1,\cdots,\mathbf{p}_N\right)$, where $\mathbf{p}_i = \left(p_i^1,\cdots,p_i^J\right)$, $p_i^j = P(Z_i = j)$, for $j = 1,\cdots,J$. Thus (9.2) is updated to

$$\begin{aligned}\mathcal{L}\left(\mathbf{X},\mathbf{Z},\sigma^2,\boldsymbol{\mu},\mathbf{T},\mathbf{p};\mathbf{D}\right)f(\mathbf{X})f(\mathbf{p}) \propto \left(\sigma^2\right)^{-m/2}\exp\left\{-\frac{\mathrm{SSR}}{2\sigma^2} - \sum_{i=1}^{N-1}\sum_{k=i+1}^{N}\log\left(\Phi\left(\frac{\delta_{ik}}{\sigma}\right)\right)\right\} \\ \times\prod_{i=1}^{N}\prod_{j=1}^{J}\left[p_i^j\left|\mathbf{T}_j\right|^{-1/2}\exp\left\{-\frac{1}{2}\left(\mathbf{X}_i-\boldsymbol{\mu}_j\right)^T\mathbf{T}_j^{-1}\left(\mathbf{X}_i-\boldsymbol{\mu}_j\right)\right\}\right]^{I(Z_i=j)} \\ \times\exp\left(-\beta\sum_{i=1}^{N}V\left(\mathbf{p}_i\right)\right),\end{aligned} \tag{9.3}$$

and then the full posterior distribution is updated to

$$
\begin{aligned}
P\left(\mathbf{X}, \mathbf{Z}, \sigma^2, \boldsymbol{\mu}, \mathbf{T}, \mathbf{p} \mid \mathbf{D}\right) &\propto \mathcal{L}\left(\mathbf{X}, \mathbf{Z}, \sigma^2, \boldsymbol{\mu}, \mathbf{T}, \mathbf{p}; \mathbf{D}\right) f(\mathbf{X}) \prod_{j=1}^{J} \phi\left(\boldsymbol{\mu}_j\right) \prod_{j=1}^{J} f\left(\mathbf{T}_j\right) f\left(\sigma^2\right) f(\mathbf{p}) \\
&\propto \left(\sigma^2\right)^{-m/2} \exp\left\{-\frac{\mathrm{SSR}}{2\sigma^2} - \sum_{i=1}^{N-1} \sum_{k=i+1}^{N} \log\left(\Phi\left(\frac{\delta_{ik}}{\sigma}\right)\right)\right\} \\
&\times \prod_{i=1}^{N} \left\{ \prod_{j=1}^{J} \left[p_i^j \left|\mathbf{T}_j\right|^{-1/2} \exp\left\{-\frac{1}{2}\left(\mathbf{X}_i - \boldsymbol{\mu}_j\right)^T \mathbf{T}_j^{-1} \left(\mathbf{X}_i - \boldsymbol{\mu}_j\right)\right\}\right]^{I(Z_i=j)} \right\} \\
&\times \prod_{j=1}^{J} \left[\left|\mathbf{T}_j\right|^{-1/2} \exp\left\{-\left(\boldsymbol{\mu}_j - \boldsymbol{\mu}_{j0}\right)^T \mathbf{T}_j^{-1} \left(\boldsymbol{\mu}_j - \boldsymbol{\mu}_{j0}\right)\right\}\right] \\
&\times \prod_{j=1}^{J} \left[\frac{\left|\mathbf{B}_j\right|^{\alpha/2}}{\Gamma\left(\alpha/2\right)} \left|\mathbf{T}_j\right|^{-\frac{\alpha+P+1}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{B}_j \mathbf{T}_j^{-1}\right)\right)\right] \\
&\times \frac{b^a}{\Gamma(a)} (\sigma^2)^{-a-1} \exp\left(-\frac{b}{\sigma^2}\right) \\
&\times \exp\left(-\beta \sum_{i=1}^{N} V\left(\mathbf{p}_i\right)\right).
\end{aligned}
$$

The conditional posterior distributions are listed below,

$$
\begin{aligned}
&f\left(\mathbf{X}_i \mid Z_i = j, \text{others}\right) \\
&\quad \propto \exp\left[-1/2 \left(\mathbf{X}_i - \boldsymbol{\mu}_j\right)^T \mathbf{T}_j^{-1} \left(\mathbf{X}_i - \boldsymbol{\mu}_j\right) - \frac{1}{2\sigma^2} \sum_{k \neq i} \left(\delta_{ik} - d_{ik}\right)^2 \right. \\
&\qquad \left. - \sum_{k \neq i} \log\left(\Phi\left(\frac{\delta_{ik}}{\sigma}\right)\right)\right], \\
&\qquad\qquad \text{for } i = 1, \cdots, N \text{ and } j = 1, \cdots, J,
\end{aligned}
$$

$$
f\left(\sigma^2 \mid \text{others}\right) \propto \left(\sigma^2\right)^{-(m/2+a+1)} \exp\left[-\frac{1}{\sigma^2}\left(\mathrm{SSR}/2 + b\right) - \sum_{i=1}^{N-1} \sum_{k=i+1}^{N} \log\left(\Phi\left(\frac{\delta_{ik}}{\sigma}\right)\right)\right],
$$

$$
f(\mathbf{p}_i \mid \text{others}) \propto \left(\prod_{j=1}^{J} p_i^j\right)^{I(Z_i=j)} \exp\left(-\beta \sum_{\forall m: W_{i,m}=1} \frac{\sum_{l=1}^{J} \left(p_i^l - p_m^l\right)^2}{1 + \sum_{l=1}^{J} \left(p_i^l - p_m^l\right)^2}\right),
$$

$$
\text{for } i = 1, \cdots, N,
$$

$$
\boldsymbol{\mu}_j \mid \text{others} \sim \phi\left(\frac{N_j \bar{\mathbf{X}}_j + \boldsymbol{\mu}_{j0}}{N_j + 1}, \frac{\mathbf{T}_j}{N_j + 1}\right), \text{ for } j = 1, \cdots, J, \tag{9.4}
$$

where $N_j = \sum_{i=1}^{N} I(Z_i = j)$ is the number of objects belong to the j^{th} cluster, $\bar{\mathbf{X}}_j = \frac{\sum_{i=1}^{N} I(Z_i=j)\mathbf{X}_i}{N_j}$ is the mean vector of group j.

$$\mathbf{T}_j \mid \text{others} \sim \text{IW}\left(\alpha + N_j/2, \mathbf{B}_j + \mathbf{S}_j/2\right), \text{ for } j = 1, \cdots, J, \tag{9.5}$$

where $\mathbf{S}_j = \sum_{i=1}^{N} \left(\mathbf{X}_i - \boldsymbol{\mu}_j\right)\left(\mathbf{X}_i - \boldsymbol{\mu}_j\right)' \text{I}\left(Z_i = j\right)$.

$$P(Z_i = j \mid \text{others}) = \frac{p_i^j \phi\left(\mathbf{X}_i; \boldsymbol{\mu}_j, \mathbf{T}_j\right)}{\sum_{l=1}^{J} p_i^l \phi\left(\mathbf{X}_i; \boldsymbol{\mu}_l, \mathbf{T}_l\right)}, \text{ for } i = 1, \cdots, N, \tag{9.6}$$

where $\phi\left(\mathbf{X}_i; \boldsymbol{\mu}_j, \mathbf{T}_j\right)$ is the multivariate normal distribution of the j^{th} component. According to a *maximum a posteriori* (MAP) assignment, the i^{th} object is assigned to $\underset{j}{\text{argmax}}$ $P(Z_i = j \mid \text{others})$.

9.1.3 Decisions on Hyperparameters and Proposal Distributions

From the conditional posterior distributions, it is easy to tell that $\mathbf{X}_i$, σ^2 and $\mathbf{p}_i$ do not follow any specific known distributions, so the newly proposed values of these parameters will be generated by using the Metropolis Hastings method, while the newly proposed values of the rest of the parameters will be generated from their conjugate conditional posterior distributions using Gibbs sampling. As the model proposed in this chapter is an augmented model based on Bayesian model-based clustering with dissimilarities proposed by Oh and Raftery [87], so we will use the same hyperparameters (a, b, α and $\mathbf{B}_j$) and proposal distributions for $\mathbf{X}_i$, σ^2 used by Oh and Raftery [87] (more details have been given in Section 4.7).

From the empirical Bayes [24] point of view and the published work of Oh and Raftery [87], the initial clustering is estimated by using spatial hierarchical clustering [13], the initial values of $\mathbf{X}^{(0)}$ and $(\sigma^2)^{(0)}$ are obtained from Bayesian multidimensional scaling introduced in Section 3.2, $(\sigma^2)^{(0)} = SSR^{(0)}/m$. $\boldsymbol{\mu}_0$ are the pre-determined cluster means based on the spatial hierarchical clustering [87]. For the hyperparameters in the prior distribution of σ^2, we set a to be a smaller value so that the inverse gamma distribution can relatively evenly cover a wider range of values, for the reason given in Section 4.7. b will be set to make the prior mean of σ^2 equal to $\text{SSR}^{(0)}/m$ according to the published

paper of Oh and Raftery [87]. The hyperparameters in the prior distribution of $\mathbf{T}_j$ are set to be $\alpha = P+4$ and $\mathbf{B}_j = (\alpha - P - 1)\mathbf{S}_j$ for the reasons given in Section 4.7, where P is the number of dimensions and $\mathbf{S}_j$ is the initial covariance matrix of the j^{th} cluster [87].

The proposal distribution of $\mathbf{X}_i^{(t)}$ is set to be a normal distribution, with mean equal to the configuration $\mathbf{X}_i^{(t-1)}$ obtained from previous iteration, for which variance is set to be proportional to $\sigma^2/(N_j - 1)$, where N_j is the number of objects in the j^{th} component for the reason given in Section 4.7.

The proposal distribution of $(\sigma^2)^{(t)}$ is set to be a truncated normal distribution, with mean $(\sigma^2)^{(t-1)}$ (the value obtained from previous iteration), for which variance is proportional to the variance of $\text{IG}(m/2 + a, \text{SSR}/2 + b)$ [87] (further details have been given in Section 4.7).

The proposal distribution of $\mathbf{p}_i$ would be set to be a Dirichlet distribution,

$$\text{Dir}\,(\alpha_1, \cdots, \alpha_J) \sim \text{Dir}\,(N_1, \cdots, N_J)\,,$$

where N_j is the number of objects in the initial j^{th} cluster if we had no spatial information. The reason for choosing the Dirichlet distribution as the proposal distribution of $\mathbf{p}_i$ is due to its property of guaranteeing that the sum of the generated values always equals to 1 and each value is a non-negative value. The hyperparameters of a Dirichlet distribution will be updated based on the number of objects in each cluster, i.e. N_j. For the spatially constrained Bayesian model-based clustering with dissimilarities model I adjust the setting of the proposal distribution of $\mathbf{p}_i$. For each $\mathbf{p}_i$, all the hyperparameters α_j will be set as 0, except for the hyperparameters relating to the memberships of its current cluster and the clusters of its spatially adjacent objects. For example, if the total number of clusters is 6 and the i^{th} object currently belongs to the second component and its contiguous neighbours' cluster memberships are either the third or the fifth components, then the hyperparameters in the Dirichlet distribution will be $(0, N_2, N_3, 0, N_5, 0)$. The initial hyparameters will be determined by the assignments in the initial clustering.

In addition, the variances in the proposal distributions of $\mathbf{X}_i$, σ^2 and $\mathbf{p}_i$ will be updated every hundred iterations. If the acceptance rate is greater than 0.45, then the variance of the proposal distribution will be doubled. If the acceptance rate is less than 0.2, then it will be reduced by half.

The variance of a Dirichlet distribution $(\alpha_1, \cdots, \alpha_j \cdots, \alpha_J)$ is

$$\text{var}_0 = \frac{\alpha_j (\alpha_0 - \alpha_j)}{\alpha_0^2 (\alpha_0 + 1)}, \tag{9.7}$$

where $\alpha_0 = \sum_{j=1}^{J} \alpha_j$. The variance of $\mathbf{p}_i$ can be updated by multiplying all Dirichlet hyperparameters by a constant H, then (9.7) will be updated to

$$\frac{H\alpha_j (H\alpha_0 - H\alpha_j)}{H^2\alpha_0^2 (H\alpha_0 + 1)} \approx \frac{1}{H} \times \text{var}_0.$$

If H is less than 1, then the variance will increase; otherwise it will decrease. So the proposal distribution of $\mathbf{p}_i$ will be updated every hundred iterations this way. If the acceptance rate is greater than 0.45, then we will multiply the proposal distribution parameters by a small H $(H = 0.5)$ to increase its proposal variance. If the acceptance rate is less than 0.2, then we will multiply the proposal distribution parameters by a large H $(H = 2)$ to reduce its proposal variance.

9.1.4 Spatially Constrained Bayesian Model-based Clustering with Dissimilarities Algorithm

The spatially constrained Bayesian model-based clustering with dissimilarities algorithm can be summarized as the following steps:

1. Select an optimal number of dimensions by using MDSIC (given in Section 3.2.1). Apply Bayesian multidimensional scaling (BMDS) (detailed in Section 3.2) to obtain the initial objects' configurations $\mathbf{X}^{(0)}$.

2. For a given number of clusters, apply spatial hierarchical clustering (further details have been given in Section 6.1) to obtain the initial clustering.

3. Use the objects' configurations and the initial clustering obtained in Steps 1 and 2 to set the $(\sigma^2)^{(0)}$, $\mathbf{B}$, $\boldsymbol{\mu}_0$ and $\mathbf{p}^{(0)}$. Choose values for the hyperparameters a, b and α.

4. Sample the parameters' values by using the conditional posterior distributions listed in Section 9.1.2. For each iteration t, perform the following steps.

   (a) Generate a new $\mathbf{X}_i^{(t)}$ and accept it according to the acceptance ratio in (2.4), for which proposal distribution is given in Section 9.1.3, for $i = 1, \cdots, N$.

(b) To get around the weak identification problem introduced in Section 2.2, we use the Procrustes transformation on each iteration's $\mathbf{X}$ to transform the sample $\mathbf{X}^{(t)}$ $(t > 0)$ to be as close as possible to the initial object configuration $\mathbf{X}^{(0)}$.

(c) Generate a new $\sigma^{2,(t)}$ and accept it according to the acceptance ratio in (2.4), for which proposal distribution is given in Section 9.1.3.

(d) Generate a new value of $\boldsymbol{\mu}_j^{(t)}$ from the conditional posterior distribution (9.4), for $j = 1, \cdots, J$.

(e) Generate a new value of $\mathbf{T}_j^{(t)}$ from the conditional posterior distribution (9.5), for $j = 1, \cdots, J$.

(f) Generate a new value of $\mathbf{p}_i^{(t)}$ and accept it according to the acceptance ratio in (2.4), for which proposal distribution is given in Section 9.1.3, for $i = 1, \cdots, N$.

(g) For each iteration, objects are assigned into different components with probabilities estimated from (9.6), then each object will be assigned to the component with the highest posterior probability.

5. Relabel the outputs to avoid the label switching phenomena.

6. Discard the burn-in period, the point estimates of the parameter values will be the average values of the chains over the convergence periods.

7. With the results from Step 6, each object will be assigned to the component with the highest probability calculated from (9.6).

9.2 Simulations

In this section, I will use simulations generated from different scenarios to compare the performance between spatially constrained Bayesian model-based clustering with dissimilarities model and spatial hierarchical clustering. The factorial design will be the same as the one in Chapter 7. The number of dimensions for simulations in Tables 9.1 to 9.7 and A.59 are set as 2, which is the true number of dimensions of the simulated data.

TABLE 9.1: Summary of ARI and BIC Based on Different Numbers of Clusters Data Using the 4 Distributions Sets Data Given Locations from Figure 7.11(a)

Fitted No.Clusters	Simulation Set-up 1 ARI (BIC)	Simulation Set-up 2 ARI (BIC)	Simulation Set-up 3 ARI (BIC)	Simulation Set-up 4 ARI (BIC)
4	0.210 (84309.8)	0.203 (74606.9)	0.215 (54610.2)	0.363 (38467.2)
5	0.616 (44381.0)	0.518 (52252.2)	0.530 (47225.1)	0.499 (12531.2)
6 ⋆	0.871 (15933.1)	0.883 △ (20390.9 ◯)	0.920 △ (8034.1 ◯)	0.890 (7483.1 ◯)
7	0.917 △ (11971.7 ◯)	0.612 (33630.1)	0.870 (17263.6)	0.917 △ (12111.0)
8	0.895 (16302.4)	0.728 (24534.1)	0.832 (17907.8)	0.885 (20949.7)
9	0.847 (20451.6)	0.766 (43449.9)	0.867 (19323.2)	0.837 (14486.9)
10	0.771 (26084.9)	0.651 (31111.0)	0.861 (20127.7)	0.631 (23757.5)
Dimensionality	2	2	2	2
CSHC ARI [a]	0.777 (0.051)	0.632 (0.063)	0.936 (0.023)	0.602 (0.058)
CSHC No.Cluster [b]	4.15 (0.23)	7.54 (0.37)	4.88 (0.19)	7.85 (0.44)
SHC ARI [c]	0.657 (0.075)	0.545 (0.089)	0.683 (0.081)	0.587 (0.085)
SHC No.Cluster [d]	8.83 (1.39)	10.87 (1.32)	8.93 (1.42)	10.67 (1.47)

The simulation results for locations in Figures 7.11(b), 7.12, 7.13 and 7.14 are shown in Appendix A.10

The data design is shown in Section 7.3.2

⋆ denotes the true number of clusters

△ denotes the clustering with the best ARI

◯ denotes the clustering with the best BIC

[a] CSHC ARI is short for the average adjusted Rand Index of Chameleon spatial hierarchical clustering

[b] CSHC No.Cluster is short for the average number of clusters of Chameleon spatial hierarchical clustering

[c] SHC ARI is short for the average adjusted Rand Index of is short for the spatial hierarchical

[d] SHC No.Cluster is short for the average number of clusters of is short for the spatial hierarchical clustering

Tables 9.1 shows the adjusted Rand Index compared to the true classification and BIC for the model of fit to different data in the location of Figure 7.11(a) over a range of possible number of clusters. From Table 9.1 we can see that as the number of clusters is approximately close to the true number of clusters, the Rand Index will get close to 1. In all simulations, BIC will be used to decide the optimal number of clusters. We can see that the estimated number of clusters by using the minimal BIC is close to the true number of clusters, except for the scenario with sparse distributions of areas and very different mixing proportions, i.e. Location in Figure 7.13(a). The ARI values of the clusterings with minimal BIC are very high in scenarios with similar mixing proportions, i.e. locations in Figures 7.11(a), 7.11(b), 7.12(a) and 7.12(b). In addition, we can see that the number of noise points does not seem to largely affect the ARI. However, it is not surprising to see that the variances will affect the ARIs. The larger variance will result in estimated clusterings with slightly smaller ARIs. Compared with the other methods, we can see that the Bayesian method gives clustering results where the average ARI or the number of clusters are less affected by the variance. Specially when the clusters are condensed (scenarios in Figure 7.11), both high and low variance have high ARI (more than 0.8).

Similar to Chapters 7 and 8, within cluster dimension dependence simulations are run, with the simulation results shown in Tables 9.2 to 9.5. We can see that the ARI will decrease when the dependence of the within cluster dimensions gets stronger. Comparing with the other methods, the spatially constrained Bayesian model-based clustering method achieves high ARI across all different scenarios, but when the variance is small (e.g. distribution 5) and the correlation is weak (i.e. 0.5), Chameleon spatial hierarchical clustering can achieve a higher ARI than the spatially constrained Bayesian model-based clustering, e.g. simulation results in Table 9.2. In addition, comparing the dimensionality between the independent and the dependent variables/dimensions, we can see that the dimensionality is not affected by the dependence, dimensionality in the dependence scenarios is not more than in the independent scenarios, which shows that we can assume the dimensions are independent in the prior distribution $\mathbf{X}_i$.

Table 9.6 to 9.7 show the results when the variances are different, the results got from spatially constrained Bayesian model-based clustering are consistent with the results from the other spatial clustering methods, the ARI does not affected very much by the difference in variances.

TABLE 9.2: Summary of ARI, $G(H+J)$, TPR and TDR of Dependent Data from Distribution Set 5 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)	Dimensionality
Spatially Constrained Model-based Clustering with Dissimilarities	0.910	-	-	6	6(4+2)	-	-	2
Spatially Constrained Finite Mixture Model with Noise Distribution	0.885 (0.013)	3.870 (0.120)	2.350 (0.556)	6.220	6(4+2)	0.962 (0.061)	0.851 (0.108)	-
Spatially Constrained Finite Mixture Model with Prior Distribution	0.829 (0.008)	-	-	6.550 (0.235)	6 (4+2)	-	-	-
Spatial Hierarchical Clustering	0.675 (0.088)	-	-	8.640 (1.270)	6 (4+2)	-	-	-
Chameleon Spatial Hierarchical Clustering	0.930 (0.028)	-	-	4.780 (0.170)	6 (4+2)	-	-	-

TABLE 9.3: Summary of ARI, $G(H+J)$, TPR and TDR of Dependent Data from Distribution Set 6 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)	Dimensionality
Spatially Constrained Model-based Clustering with Dissimilarities	0.872	-	-	6	6(4+2)	-	-	2
Spatially Constrained Finite Mixture Model with Noise Distribution	0.878 (0.019)	3.800 (0.155)	2.440 (0.556)	6.240	6(4+2)	0.925 (0.085)	0.835 (0.122)	
Spatially Constrained Finite Mixture Model with Prior Distribution	0.806 (0.014)	-	-	6.640 (0.220)	6 (4+2)	-	-	
Spatial Hierarchical Clustering	0.672 (0.074)	-	-	8.500 (1.820)	6 (4+2)	-	-	
Chameleon Spatial Hierarchical Clustering	0.911 (0.034)	-	-	4.630 (0.280)	6 (4+2)	-	-	

TABLE 9.4: Summary of ARI, $G(H+J)$, TPR and TDR of Dependent Data from Distribution Set 7 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)	Dimensionality
Spatially Constrained Model-based Clustering with Dissimilarities	0.907	-	-	7	6(4+2)	-	-	2
Spatially Constrained Finite Mixture Model with Noise Distribution	0.625 (0.075)	5.180 (0.830)	2.810 (1.856)	7.990	6(4+2)	0.844 (0.075)	0.123 (0.104)	
Spatially Constrained Finite Mixture Model with Prior Distribution	0.675 (0.086)	-	-	8.330 (1.195)	6 (4+2)	-	-	
Spatial Hierarchical Clustering	0.540 (0.078)	-	-	10.480 (1.630)	6 (4+2)	-	-	
Chameleon Spatial Hierarchical Clustering	0.589 (0.095)	-	-	7.540 (0.780)	6 (4+2)	-	-	

TABLE 9.5: Summary of ARI, $G(H+J)$, TPR and TDR of Dependent Data from Distribution Set 8 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)	Dimensionality
Spatially Constrained Model-based Clustering with Dissimilarities	0.883	-	-	6	6(4+2)	-	-	2
Spatially Constrained Finite Mixture Model with Noise Distribution	0.610 (0.084)	5.040 (0.820)	3.050 (2.460)	8.090	6(4+2)	0.835 (0.038)	0.114 (0.136)	
Spatially Constrained Finite Mixture Model with Prior Distribution	0.658 (0.106)	-	-	8.560 (1.205)	6 (4+2)	-	-	
Spatial Hierarchical Clustering	0.536 (0.085)	-	-	10.210 (1.890)	6 (4+2)	-	-	
Chameleon Spatial Hierarchical Clustering	0.582 (0.058)	-	-	7.420 (0.530)	6 (4+2)	-	-	

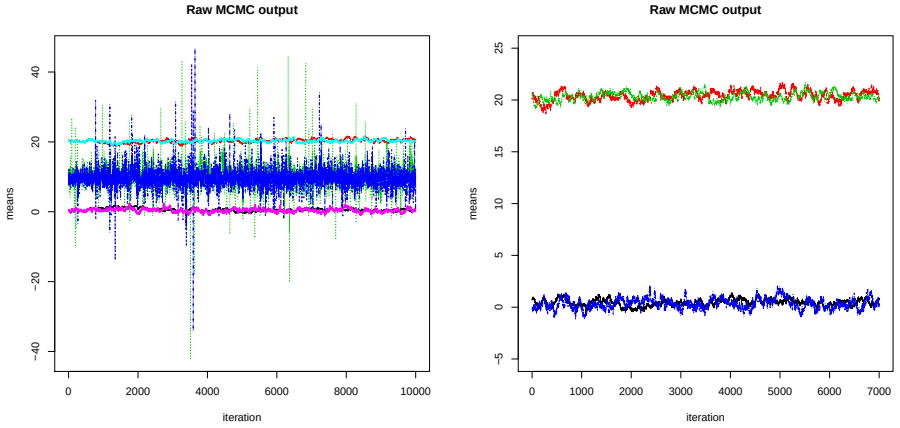
TABLE 9.6: Summary of ARI, $G(H+J)$, TPR and TDR of Different Diagonals Data from Distribution Set 9 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Model-based Clustering with Dissimilarities	0.915	-	-	6	6(4+2)	-	-
Spatially Constrained Finite Mixture Model with Noise Distribution	0.893 (0.007)	4.000 (0.000)	2.180 (0.409)	6.180	6 (4+2)	1.000 (0.000)	0.925 (0.084)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.869 (0.012)	-	-	6.265 (0.200)	6 (4+2)	-	-
Spatial Hierarchical Clustering	0.676 (0.086)	-	-	8.76 (1.31)	6 (4+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.921 (0.083)	-	-	4.67 (0.17)	6 (4+2)	-	-

TABLE 9.7: Summary of ARI, $G(H+J)$, TPR and TDR of Different Diagonals Data from Distribution Set 10 Located in Figure 7.11(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Model-based Clustering with Dissimilarities	0.894	-	-	6	6(4+2)	-	-
Spatially Constrained Finite Mixture Model with Noise Distribution	0.637 (0.079)	5.280 (0.780)	2.710 (2.410)	7.990	6 (4+2)	1.000 (0.000)	0.917 (0.045)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.703 (0.085)	-	-	8.250 (1.210)	6 (4+2)	-	-
Spatial Hierarchical Clustering	0.542 (0.072)	-	-	10.65 (1.39)	6 (4+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.593 (0.089)	-	-	7.51 (0.53)	6 (4+2)	-	-

Figure 9.1 give the trace plots over all iterations in the first dimension for the scenario labeled in red in Table 9.1.



(a) McMC of Means with Noise Groups in Scenario 7.11(a) of Data from Set 3

(b) McMC of Means without Noise Groups in Scenario 7.11(a) of Data from Set 3

FIGURE 9.1: Parameters Convergence Plots in Scenario 7.11(a) of Data from Set 3

Figure 9.1 shows that the trace plots of the components' means in scenario 7.11(a) with data generated from date set 3. The number of iterations is 10,000. The discard period of the McMC chain in Figure 9.1 is the first 3,000 iterations. Although the parameters started to converge from an very early stage, we discarded more iterations to try to ensure the stability of the convergence. From Figure 9.1(a) we can see that the green line and the blue varied in a wider range around the line 10, while the other four lines varied in a relatively narrow range around the lines 0 and 20, which turn out to be the estimated means for the main clusters. In order to better observe the behaviors of the main components' means, the trace plot of the main clusters only is given in Figure 9.1(b).

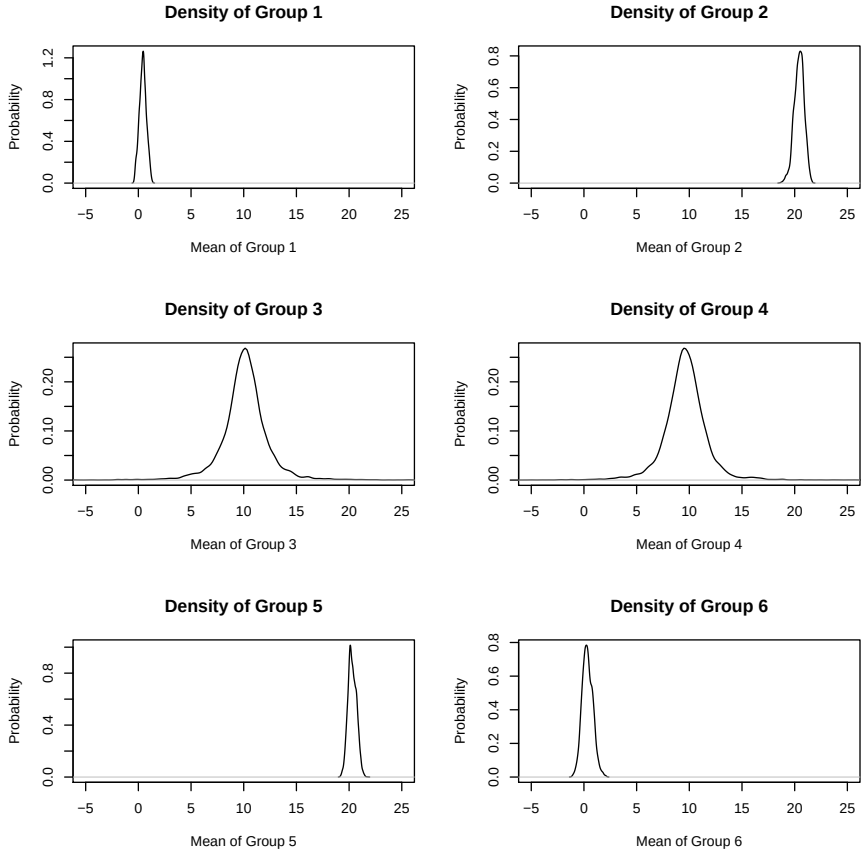


FIGURE 9.2: Probability Density Plots of Cluster Means for Scenario in Figure 7.11(a)

The probability density plots for different components' means are displayed on the same scale in Figure 9.2. The probability density plots in the third and fourth sub-figures are the probability density plots of noise clusters, which have relatively large variance, so the probability density plots cover a wider range of values. The probability density plots in the first and sixth sub-figures are the probability density plots of cluster means for clusters generated from the first dimension of MVN $\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0.25 & 0\\0 & 0.25\end{pmatrix}\right)$, while the probability density plots in the second and fifth sub-figures are the probability density plots of cluster means for clusters generated from the first dimension of MVN $\left(\begin{pmatrix}20\\20\end{pmatrix}, \begin{pmatrix}0.25 & 0\\0 & 0.25\end{pmatrix}\right)$. These distributions cover a relative small range of values.

9.3 Spatially Constrained Bayesian Model-based Clustering with Dissimilarities Applied to Glasgow CPEP Data

Spatially constrained Bayesian model-based clustering with dissimilarities groups areas using dissimilarities data, so we need to transform CPEP from similarity into dissimilarity data. In this case we used the reciprocal of CPEP (all the diagonal elements are set to be 0s). As this technique carries out multidimensional scaling and model-based clustering at the same time, there is an issue in choosing both the optimal number of dimensions and clusters.

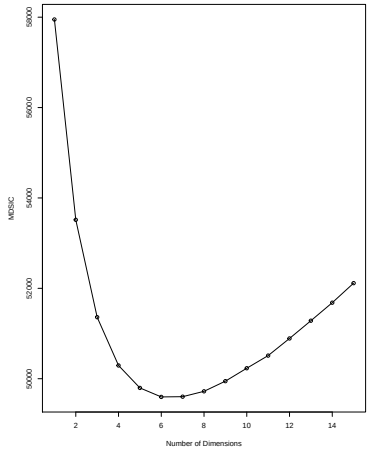
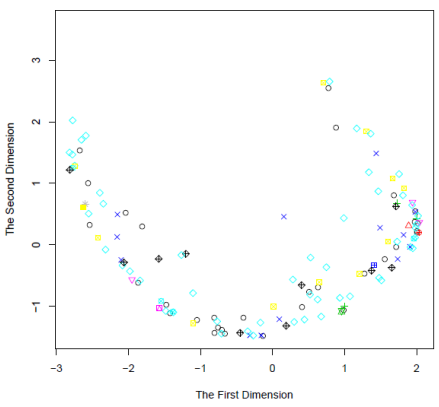
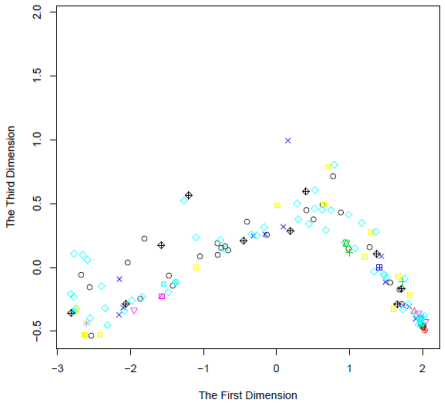


FIGURE 9.3: Number of Dimensions for Glasgow CPEP by Using Spatially Constraint Bayesian Model-based Clustering

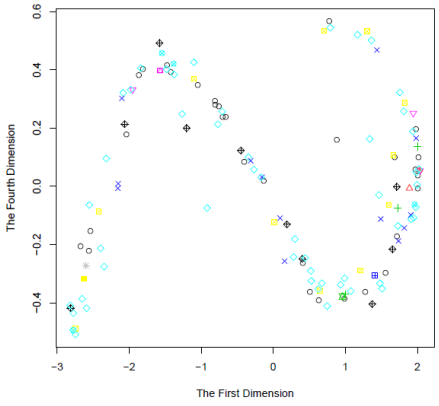
The MDSIC plot in Figure 9.3 illustrates the decision made in the number of dimensions. When $P = 6$, it has the minimum MDSIC (further details have been given in Section 3.2.1), so the number of dimensions will be set as 6 in this application. A few CPEP configurations in a paired dimensional space are displayed in Figure 9.4.



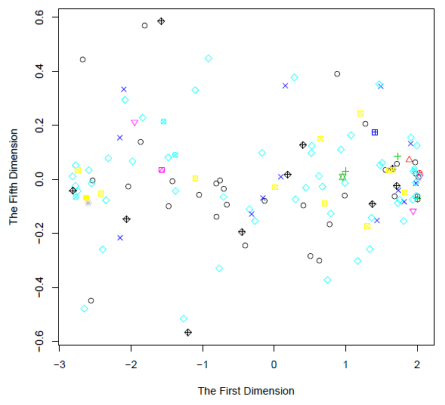
(a) CPEP Configuration In the First and Second Dimensions



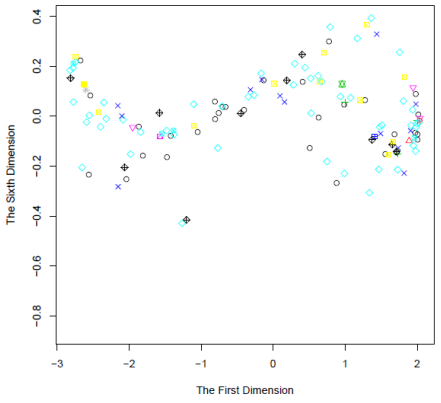
(b) CPEP Configuration In the First and Third Dimensions



(c) CPEP Configuration In the First and Fourth Dimensions



(d) CPEP Configuration In the First and Fifth Dimensions



(e) CPEP Configuration In the First and Sixth Dimensions

FIGURE 9.4: CPEP Configuration in a Six Dimensional Space

Figure 9.4 shows the CPEP configurations between the first and other dimensions. Different colours and shapes represent different clusters in the clustering from the Figure 9.6. It is easy to tell that the clustering cannot be visually found from the CPEP configurations. Specifically in Figure 9.4(d), the data is randomly scattered in these two dimensions. For example, the singleton area 8, Finnieston and Kelvinhaugh, in Figure 9.6 represented by the red circle with cross in the middle right in Figure 9.4(a) or bottom the right corner in Figure 9.4(b), is lying not very far from the others. Further information about the singletons will be explained in the paragraph followed by Figure 9.6.

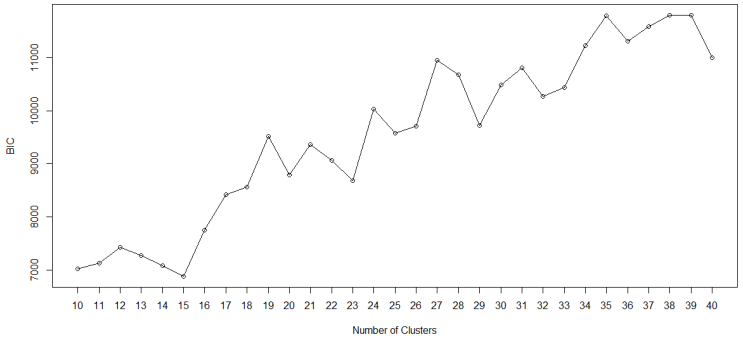


FIGURE 9.5: BIC in a Two Dimensional Space

The BIC for different number of clusters are shown in Figure 9.5. Looking at the clusterings achieved using the other spatial clustering techniques, which have more than 20 clusters, I start the number of clusters from $G = 10$. As BIC keeps increasing after $G = 23$, so Figure 9.5 only shows the numbers of clusters from 10 and 40. The number of clusters with the minimal BIC is when $G = 15$, which is shown in Figure 9.6.

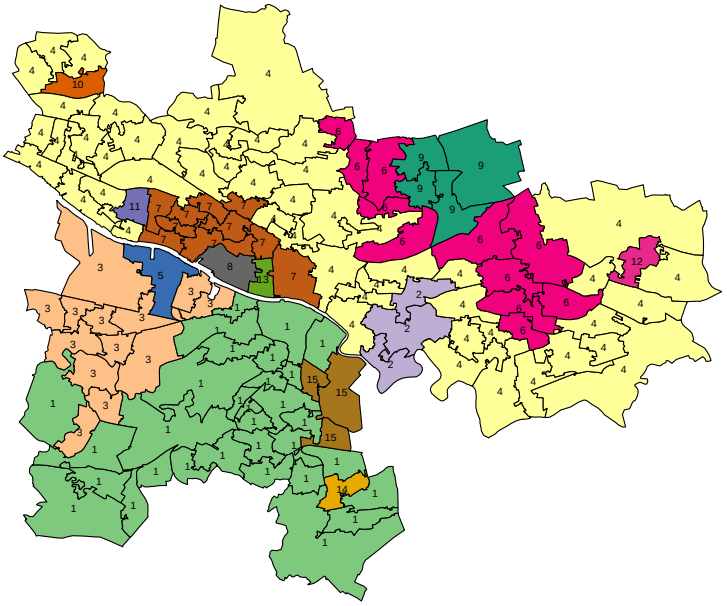
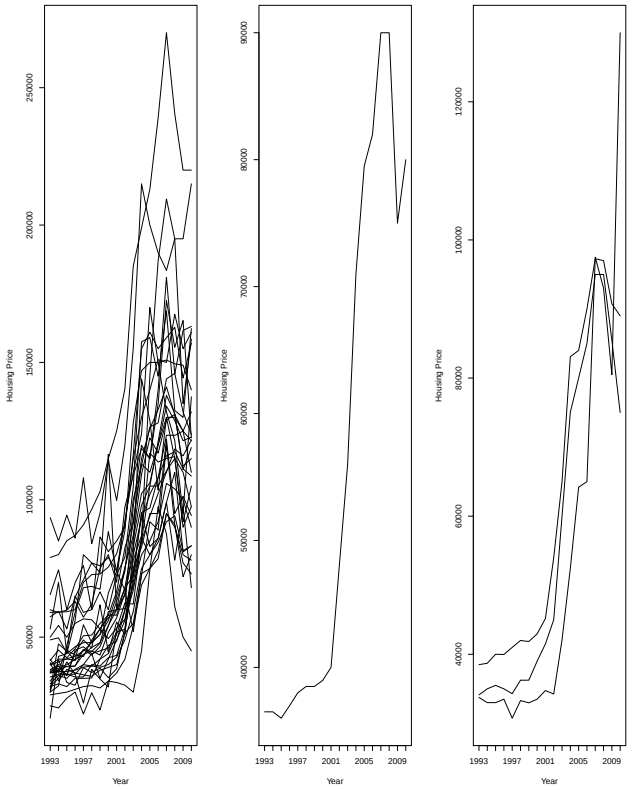
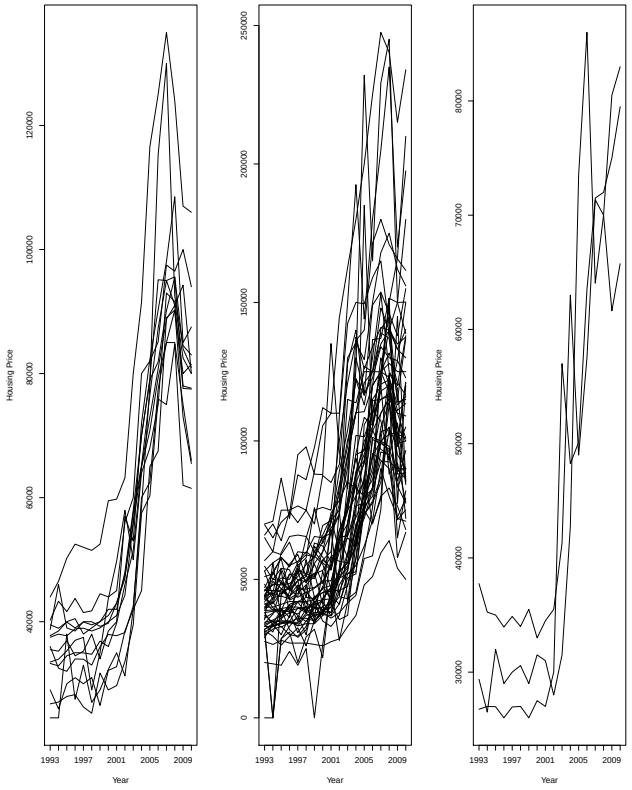


FIGURE 9.6: Bayesian Clustering in Glasgow Housing Market in a Six Dimensional Space

Figure 9.6 shows the clustering with 15 clusters using Bayesian spatially constrained model-based clustering. Compared with the clustering in Figure 7.15, both have very similar patterns and a similar number of clusters, but the clustering in Figure 9.6 finds more singleton clusters than the one in Figure 7.15. The singleton clusters are Drumchapel South (10), Victoria Park (11), Finnieston and Kelvinhaugh (8), Anderston (13), Govan and Linthouse (5), Castlemilk (14). The changing patterns of all these areal units can be viewed from Figures 9.7, 9.8 and 9.9.

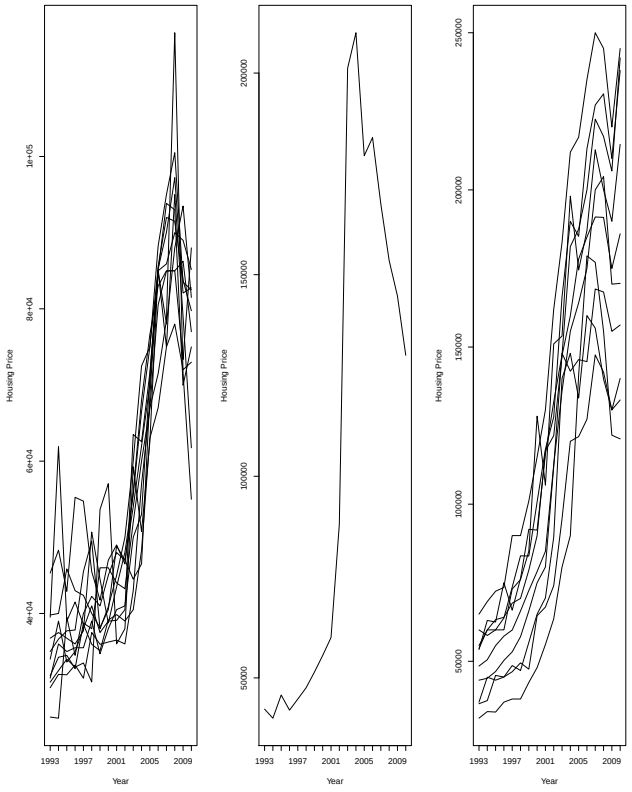


(a) Areas in Clusters 1, 14, 15 in Figure 9.6

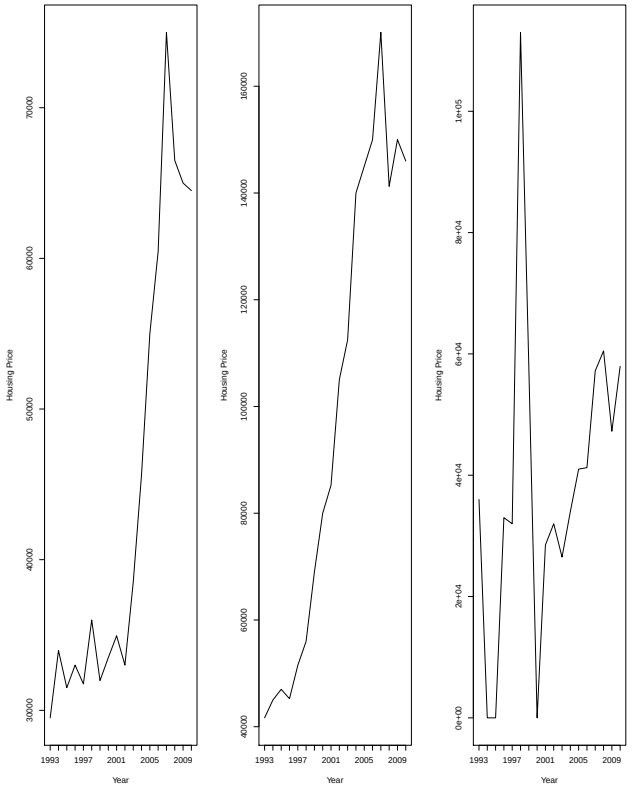


(b) Areas in Clusters 3, 4, 2 in Figure 9.6

FIGURE 9.7: Time Series Based on Spatially Constrained Model-Based Clustering -part1

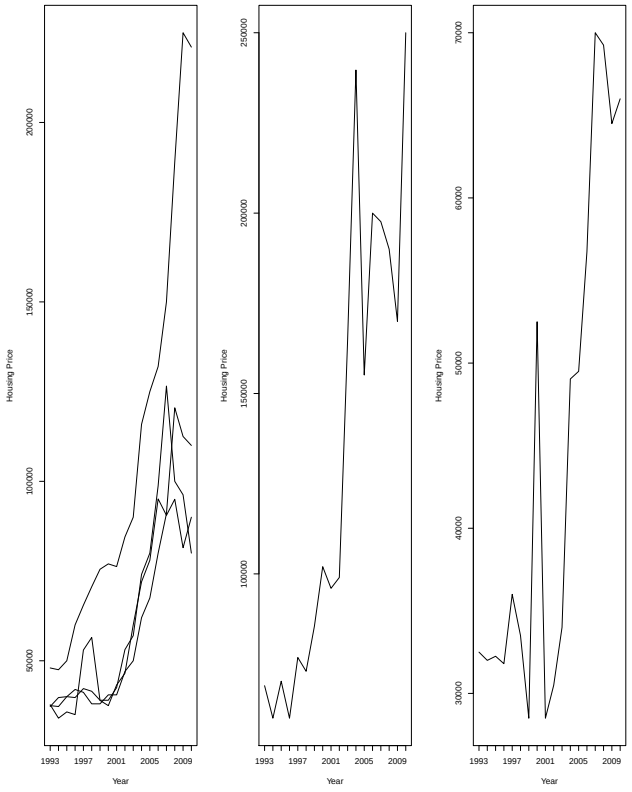


(a) Areas in Clusters 6, 13, 7 in Figure 9.6



(b) Areas in Clusters 5, 8, 12 in Figure 9.6

FIGURE 9.8: Time Series Based on Spatially Constrained Model-Based Clustering -part2



(a) Areas in Clusters 9, 11, 10 in Figure 9.6

FIGURE 9.9: Time Series Based on Spatially Constrained Model-Based Clustering -part3

Figures 9.7, 9.8 and 9.9 show the changing patterns of all 2001 intermediate zones over 1993 to 2010. Generally speaking, most of the areas experienced a sharp price decrease in 2008 (clusters 1 and 4, 15), but there are some areas that changed in different patterns. In cluster 15, all three areas, Toryglen and Oatlands, Govanhill East and Aikenhead, Kingspark North, reached a similar highest price in 2007, but dropped slightly after 2008. It is also interesting to see that cluster 12 (Central Easterhouse) has the most different pattern, it reached the peak housing price in 1997 and sharply decreased and reached the bottom in 2001. It is easy to see that three areas in cluster 2 (Dalmarnock, Carntyne West and Haghill, Parkhead West and Barrowfield) changed in a very similar way over these 18 years, all of them increased slowly before 2001, but increased sharply after 2001.

TABLE 9.8: ARI comparing Spatial Clusterings for the Glasgow Housing Market in a Six-dimensional Space

	Spatial Hierarchical Clustering (Section 6.2)	Chameleon Spatial Hierarchical Clustering (Section 7.4)	Spatially Constrained Bayesian Model-based Clustering with Dissimilarities Model
Spatial Hierarchical Clustering	1.000	0.652	0.698
Chameleon Spatial Hierarchical Clustering	-	1.000	0.841
Spatially Constrained Bayesian Model-based Clustering with Dissimilarities Model	-	-	1.000

From Table 9.8 we can see that there is a high consistency between Bayesian spatially constrained model-based clustering and Chameleon spatial hierarchical clustering. Spatial hierarchical clustering also has a relative high similarities with both Bayesian spatially constrained model-based clustering and Chameleon spatial hierarchical clustering. According to the Glasgow housing market information and the simulation results, compared with all three CPEP clustering results, I would choose the clustering in Figure 9.6 as the best clustering. Firstly, the simulation results in Table 9.1 concluded the Bayesian model-based clustering with dissimilarities is less affected by the variance. Secondly, from the changing patterns or tendencies of these 133 intermediate zones over these 15 years in Figures 9.7 to 9.9, we can observe very similar changing patterns in each cluster, e.g. the changing patterns in cluster 7 in Figure 9.8(a), which supports the closeness of the estimated clustering and the true classification.

9.4 Summary of Spatially Constrained Bayesian BMCD

Spatially constrained Bayesian model-based clustering with dissimilarities introduced in Chapter 9 clusters spatial data based on dissimilarity data and carries out multidimensional scaling and model-based clustering simultaneously. Compared with the other methods, we can see that Bayesian method gives clustering results where average ARI or the number of clusters are less affected by the variance. In the application, compared with

Chameleon spatial hierarchical clustering, the Bayesian spatially constrained model-based clustering generates a clustering with a larger number of clusters, so more details and differences among different areas can be identified. Compare with the methods introduced in the previous chapters, the CPEP clustering results is more reasonable according to the simulation results and the Glasgow housing market information.

Chapter 10

Conclusion

Cluster analysis is a technique to assign objects into groups, where objects put into the same group share similar characteristics or are closer to each other than objects assigned to different clusters. One of the applications in cluster analysis is to group spatial data which are spatially contiguous in nature. However, common clustering techniques which used to group spatial data will not guarantee this contiguity. The aim of the thesis is to develop some novel spatial clustering techniques to group spatial data, and I also applied these newly developed spatial clustering techniques to identify the substitutable submarkets in Glasgow housing market, where the properties grouped in the same clusters are substitutable to customers.

The majority of standard clustering techniques ignore spatial contiguity when grouping spatial data. Based on the need of grouping data, Anderson et al. [13] proposed a spatial hierarchical clustering algorithm to identify the spatial patterns of disease mapping in Glasgow, which results in all the areal units grouped together sharing at least one common boundary. So based on the novelty of the spatial hierarchical clustering, I extended Chameleon hierarchical clustering in a similar manner. I then extended the finite mixture models and Bayesian model-based clustering with dissimilarities models by incorporating a spatial prior term in the regular model to enforce spatial contiguity. In addition, the applications of these novelty spatial clustering algorithms were extended from univariate to multivariate space.

10.1 Chameleon Spatial Hierarchical Clustering

Chameleon spatial hierarchical clustering algorithm was introduced in Chapter 7. Instead of giving all the possible clusterings like spatial hierarchical clustering does, Chameleon spatial hierarchical clustering only gives a reduced hierarchy of clusterings (a small proportion of the complete hierarchy of clusterings).

The advantages of applying Chameleon spatial hierarchical clustering can be summarized as follows. Firstly, it does not require object coordinates, which are not always available in practice. Secondly, it works much faster than spatial hierarchical clustering algorithm across all different scenarios, reducing the time used by the spatial hierarchical clustering algorithm by almost half in my simulations. So it will be more competitive for application to large-sized data than other spatial clustering techniques. From the simulation results we can see that the clustering results from Chameleon spatial hierarchical clustering and spatial hierarchical clustering are very similar. However, Chameleon spatial hierarchical clustering did especially well in scenarios with less noise points.

One of the disadvantages in Chameleon spatial hierarchical clustering is the setting of tuning parameters. Although Chameleon spatial hierarchical clustering does well in many scenarios, it requires specifications of parameters (K, C, M, α, α_0) to achieve a good estimated clustering, which means this requires more preliminary runs or prior information.

10.2 Spatially Constrained Finite Mixture Models

In Chapter 8, I extended the classical finite mixture model to a spatially constrained finite model by including a spatial term and used GEM based on gradient projection algorithm to estimate the model parameters. The improvements for spatially constrained finite mixture models are not only limited to improving the grouping of spatial data, but also making the finite mixture models more robust in dealing with the singular covariance matrices issue by either identifying the noise objects as a separate group or incorporating prior terms into models. Specifically, the spatially constrained finite mixture model with noise points uses the nearest neighbour clutter removal algorithm to initially identify potential noise points before modeling the data, and models those

potential noise points in the finite mixture model with a uniform distribution. Spatially constrained finite mixture model with priors models the data by including prior terms in the spatially constrained finite models to avoid singular covariance matrices. In addition, I also extended the spatially constrained finite mixture models from univariate to multivariate space. From the simulation results obtained in Chapter 8, we can see that the spatially constrained finite mixture model with noise points tends to produce more clusters, and the clustering results are less affected by the number of noise points and variances than the spatially constrained finite mixture model with prior terms. The spatially constrained finite mixture model with prior terms is good at clustering data with dense distribution of areas and small variances. However, for the same dense distribution of areas and small variances, the spatial hierarchical clustering does slightly better than the spatially constrained finite mixture model with priors.

One of the disadvantages of spatially constrained finite mixture models is that all the clusters are modeled by normal distributions. However, for some extremely skewed data or categorical data, these should be modeled by other distributions. In addition, in Chapter 8, I incorporated all components (non-noise components and the noise points) with the spatially constrained membership prior. However, it would also be worth comparing the behaviors of the spatially constrained model with noise points proposed in Chapter 8 with the model if the spatially constrained prior only combines with the non-noise components.

10.3 Spatially Constrained Bayesian Model-based Clustering with Dissimilarities

Although often the spatially constrained finite mixture models with GEM algorithm from Chapter 8 can run much faster than McMC, GEM can only provide a point estimate of the parameter of interest and does not quantify the full posterior distribution. In addition that model requires data configurations rather than dissimilarities. Spatially constrained Bayesian model-based clustering with dissimilarities introduced in Chapter 9 clusters spatial data based on dissimilarity data and carries out multidimensional scaling and model-based clustering simultaneously. In some simulations, when the number of clusters is too small, comparing the estimated clusterings from the model to the true classification, the estimated clusterings are not good, with low ARIs. However, when the estimated number of clusters is approximately around the true number of clusters, the estimated clusterings are good, with high ARIs. The advantage of the

spatially constrained Bayesian model-based clustering with dissimilarities model over the other spatial clustering techniques is that it can obtain parameter distributions, which can estimate the probability of obtaining a parameter value. However, one of the disadvantages of the spatially constrained Bayesian model-based clustering with dissimilarities model is that it is much slower than spatial hierarchical clustering. In addition, the distances used in spatially constrained Bayesian model-based clustering with dissimilarities are assumed to be Euclidean distances. However, there are many other distances that could be used instead of Euclidean distance.

10.4 Simulation Summary

When the given datasets are relationship data (similarity or dissimilarity), from simulation results we can see that spatially constrained BMCD achieved higher ARI (compared with the true classifications) and closer number of clusters compared to the actual number of true groups in most scenarios, e.g. Tables 9.1. Chameleon spatial hierarchical clustering tends to detect fewer clusters compared with the actual number of groups, e.g. Table 7.5. Spatial hierarchical clustering will find more clusters than Chameleon spatial hierarchical clustering and the actual classification, e.g. Table 7.5. When the given datasets are coordinate data, spatially constrained finite mixture model with noise points tends to group data with more clusters, e.g. Tables 8.2 and 8.3. Comparing the zero and non-zero off-digonals in the covariance matrix simulation results, we can see that for the same scenario, when the dependence between within cluster dimensions (the off-diagonals in the covariance matrix are higher) is stronger, then the average ARI will be lower, but the clustering is still identifiable. Comparing all these four methods and difference scenarios, the difference in variances will not affect the simulation results very much.

10.5 Application to Glasgow Data

In each novel spatial clustering technique chapter, the proposed spatial clustering technique was applied to Glasgow housing market data. The study region is the city of Glasgow, including 133 administrative units which are defined as intermediate zones. Intermediate zones are built from groups of data zones and fit within council area boundaries, each intermediate zone contains at least 2,500 residents [4]. In Chapters 7 and 9, I used the

CPEP or transformed CPEP (i.e. the reciprocal CPEP data with all diagonal elements are 0s) as the input data. The optimal number of clusters for spatial hierarchical clustering and Chameleon spatial hierarchical clustering are 30 and 13 respectively. Spatial hierarchical clustering achieves more clusters and created a clustering with a larger number of submarkets. Compared with the clustering achieved from Chameleon spatial hierarchical clustering, it clustered the East End into more submarkets. Although both clustering techniques clustered the West End into a different cluster, spatial hierarchical clustering clustered the West End into more detailed submarkets. The reason spatial hierarchical clustering achieved more clusters is because objects grouped into different clusters in the earlier stage will not be merged with the other groups, but Chameleon spatial hierarchical cluster allows the different clusters to merge at the merging stage if necessary. CPEP data describe the changing patterns of different areas over years. The changing pattern of areas are less varied than the actual housing prices, it shows information about the relative values rather than the absolute values. Based on the aim of identifying the similar changing pattern areas and the available data, Chameleon spatial hierarchical clustering is preferred over spatial hierarchical clustering as the required input data is closer to CPEP, which will reduce the errors in transforming data. In addition, the clustering obtained by spatially constrained BMCD has 15 clusters, its clustering is more similar to the clustering obtained from Chameleon spatial hierarchical clustering, for which ARI is 0.841. More specifically, the clustering of the East End is similar to Chameleon spatial hierarchical clustering, but spatially constrained BMCD grouped the West End into more clusters than Chameleon spatial hierarchical clustering did. Generally speaking, the clusterings achieved by all three techniques are very similar according to ARI comparisons in Table 9.8. According to the simulation results, compared with all three CPEP clustering results, I would choose the clustering in Figure 9.6 as more reasonable. Firstly, the simulation results in Table 9.1 concluded the Bayesian model-based clustering with dissimilarities is less affected by the variance. Secondly, the changing patterns or tendencies of these 133 intermediate zones over these 15 years in Figures 9.7 to 9.9, we can observe very similar changing patterns in each cluster, e.g. the changing patterns in cluster 7 in Figure 9.8(a), which supports the closeness of the estimated clustering and the true classification.

In Chapter 8, I combined the median house price, median gross household income in 2010 with the same year's the number of over 60 income support claims as the input data, and then used the spatially constrained finite mixture model with noise points or with prior terms to group this combined three dimensional data. The clusterings obtained from both techniques having 30 clusters, but the spatially constrained finite mixture model with noise points obtained more singleton clusters than with prior terms did. One of the most different parts between these two techniques is the clustering on the north side of

Clyde river, the cluster 10 in Figure 8.14 was clustered into more submarkets in Figure 8.12. In addition, cluster 17 in Figure 8.14 has been grouped into several submarkets in the clustering shown in Figure 8.12. Comparing the 3D data clustering results in Figures 8.12 and 8.14, the clustering shows in Figure 8.12 is more reasonable. The clustering shown in Figure 8.12 clusters the West End into several submarkets, which is consistent with Glasgow housing market information published by RSS [3]. For example, although Anderston (cluster 18 in Figure 8.12) and Finnieston and Kelvinhaugh (cluster 20 in Figure 8.12) have similar housing prices, the difference in low income claims is large, which indicates these two areal units should be clustered into different clusters.

10.6 Summary

In this thesis, I proposed three novel clustering techniques in grouping spatial data and also extended them from univariate to multivariate space. These newly proposed spatial clustering techniques make it possible to group a region into several spatial contiguous clusters with the areal units grouped together sharing similar attributes.

However, one of the drawbacks of all these spatial clustering methods is the formed number of clusters may not be equal to the initially set number of clusters. The number of clusters in the formed clustering structure may only be close to the initially set number of clusters. For example, in spatially constrained Bayesian model-based clustering with dissimilarities, the formed number of clusters may be less than the set number of clusters as at the classification step, the objects are assigned to the cluster with the highest posterior probability, then some redundant clusters will be emptied if necessary. On the other hand, the formed number of clusters might increase in the spatially constrained finite mixture model with noise points as we only set the number of main clusters, but give different memberships for objects assigned to the uniform distribution. There are no direct parameters in Chameleon spatial hierarchical clustering that determine the final formed number of clusters. It uses the combination of all pre-decided parameters (i.e. K, C, α, α_0) to determine the formed clustering structure. In Chapters 8 and 9, the off-diagonals in covariance matrix Λ of the coordinate data $\mathbf{X}$ can also be set as non-zero values to meet more general scenarios.

In both Chapters 8 and 9 I used the clustering from spatial hierarchical clustering as the initial clustering to estimate the initial cluster parameters, i.e. the cluster means and covariance, different clusters are given different initial values. The same manner of

setting is also used in Oh and Raftery [86] and Oh and Raftery [87]. However, some studies suggest using unsupervised clustering (clustering without the knowledge of any previous rules or clusters) to avoid using the same information or the empirical based results twice. Nishant Arora, Sandeep Jain and Santosh Kumar Verma [14] discussed the performance caused by the priori knowledge requirements of initial clusters. So further studies could be performed to compare the behaviors of the proposed clustering models between the unsupervised and empirical-based clusterings, i.e. the empirical-based clustering is the same initial clustering as the one used in this thesis, the unsupervised clustering is the one clustered without the knowledge of any previous rules or clusters.

Appendix A

A.1 Calculation about Integration of $h(\sigma^2, \mathbf{X})$

$$h(\sigma^2, \mathbf{X}) = (\sigma^2)^{-(m/2+a+1)} \exp\left[-\frac{SSR/2+b}{\sigma^2} - \sum_{i>j} \log \Phi\left(\frac{\delta_{ij}}{\sigma}\right)\right],$$

When N is extremely large, $h(\sigma^2, \mathbf{X})$ is approximately equal to the likelihood function.

$$h(\sigma^2, \mathbf{X}) \approx \mathcal{L}(\mathbf{X}, \sigma^2) = (\sigma^2)^{-m/2} \exp\left[-\frac{SSR/2}{\sigma^2} - \sum_{i>j} \log \Phi\left(\frac{\delta_{ij}}{\sigma}\right)\right], \tag{A.1}$$

As we mentioned in Section 3.2, $-\frac{SSR/2}{\sigma^2}$ is the dominated term of $\mathcal{L}(\mathbf{X}, \sigma^2)$, so we can simplify (A.1) to

$$\mathcal{L}(\mathbf{X}, \sigma^2) \approx (\sigma^2)^{-m/2} \exp\left[-\frac{SSR/2}{\sigma^2}\right],$$

A Laplace approximation [70] to the integral of a function is defined as follows:

$$\begin{aligned} I &= \int_a^b \exp\left(-\lambda g(y)\right) h(y) dy \\ &\approx \exp\left(-\lambda g(y^*)\right) h(y^*) \sqrt{\frac{2\pi}{\lambda g''(y^*)}} \end{aligned}$$

According to a Laplace approximation to the integral of $\mathcal{L}(\mathbf{X}, \sigma^2)$, where $\lambda = SSR/2$, $g(\sigma^2) = 1/\sigma^2$, $h(\sigma^2) = (\sigma^2)^{-1/m}$ and $(\sigma^2)^*$ is the MLE esimator of σ^2, which is equal to SSR/m. So $g''(\sigma^2) = 2/(\sigma^2)^3$ and then we can get

$$\int \mathcal{L}(\mathbf{X}, \sigma^2) d\sigma^2 \approx (2\pi)^{1/2} (1/m)^{1/2} (SSR/m)^{-m/2+1} \exp(-m/2).$$

A.2 Code for Bisecting a Cluster

```
approximateBisection<-function(weightMatrix,mode="matrix",minimumGain=1e-5){
#   minimumGain<-1e-5 # minimum value for gain, setting it to 0 might lead to
#   infinite loop due to numerical inaccuracy

  N<-dim(weightMatrix)[1] # number of elements
  m<-N/2

  # start off with a random partition
  A<-sample(1:N,N/2,replace=FALSE)
  B<-(1:N)[-A]

  maxGain<-Inf
  while(maxGain>minimumGain){
    DA<-rowSums(weightMatrix[A,B])-rowSums(weightMatrix[A,A])+diag(weightMatrix[A,A])
    DB<-rowSums(weightMatrix[B,A])-rowSums(weightMatrix[B,B])+diag(weightMatrix[B,B])
    unmarkedA<-1:m
    unmarkedB<-1:m
    markedA<-rep(0,m)
    markedB<-rep(0,m)
    gains<-rep(0,m)
    for(k in 1:m){
      # find best pair from remainder
  %    # with 2 loops, slow but easy on memory
if(mode=='2loops'){
      bestGain<--Inf
      besti<-0
      bestj<-0
      for(i in unmarkedA)
        for(j in unmarkedB){
          gain<-DA[i]+DB[j]-2*weightMatrix[A[i],B[j]]
          if(gain>bestGain) {bestGain<-gain; besti<-i;bestj<-j}
        }
#           mark the best pair
        unmarkedA<-unmarkedA[-which(unmarkedA==besti)]
        unmarkedB<-unmarkedB[-which(unmarkedB==bestj)]
        markedA[k]<-besti
        markedB[k]<-bestj
}

       gains[k]<-bestGain

        # update D for unmarked indices
        DA[unmarkedA]<-DA[unmarkedA]
+2*weightMatrix[A[unmarkedA],A[besti]]-2*weightMatrix[A[unmarkedA],B[bestj]]
        DB[unmarkedB]<-DB[unmarkedB]
+2*weightMatrix[B[unmarkedB],B[bestj]]-2*weightMatrix[B[unmarkedB],A[besti]]
      }
      gains<-cumsum(gains)
      bestPartition<-which.max(gains)
      maxGain<-gains[bestPartition]
      if(maxGain>minimumGain){
          # swap best pairs
```

```
          A1<-c(A[-markedA[1:bestPartition]],B[markedB[1:bestPartition]])
          B1<-c(B[-markedB[1:bestPartition]],A[markedA[1:bestPartition]])
          A<-A1
          B<-B1
          }
    }
  list(A,B)
}
```

A.3 Expanded Decision about K in K-NN Graph Stage for Chapter 7

TABLE A.1: Results for Different K for Data of Type Given in Figure 7.11(b)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$K = 2$	0.665 (0.032)	5.20 (0.12)	10 (4+6)
$K = 3$	0.739 (0.025)	4.17 (0.06)	10 (4+6)
$K = 5$	0.722 (0.043)	4.55 (0.10)	10 (4+6)

TABLE A.2: Results for Different K for Data of Type Given in Figure 7.12(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$K = 2$	0.448 (0.011)	5.85 (0.18)	8 (6+2)
$K = 3$	0.530 (0.007)	5.70 (0.20)	8 (6+2)
$K = 5$	0.574 (0.014)	6.08 (0.24)	8 (6+2)

TABLE A.3: Results for Different K for Data of Type Given in Figure 7.12(b)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$K = 2$	0.494 (0.065)	6.02 (0.32)	12 (6+6)
$K = 3$	0.509 (0.052)	5.45 (0.26)	12 (6+6)
$K = 5$	0.541 (0.048)	5.85 (0.27)	12 (6+6)

TABLE A.4: Results for Different K for Data of Type Given in Figure 7.13(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$K = 2$	0.734 (0.031)	3.10 (0.17)	4 (2+2)
$K = 3$	0.869 (0.024)	2.05 (0.04)	4 (2+2)
$K = 5$	0.842 (0.029)	1.88 (0.05)	4 (2+2)

TABLE A.5: Results for Different K for Data of Type Given in Figure 7.13(b)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$K = 2$	0.611 (0.041)	4.80 (0.19)	8 (2+6)
$K = 3$	0.772 (0.044)	3.45 (0.18)	8 (2+6)
$K = 5$	0.816 (0.052)	3.90 (0.22)	8 (2+6)

TABLE A.6: Results for Different K for Data of Type Given in Figure 7.14(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$K = 2$	0.488 (0.023)	5.15 (0.29)	8 (6+2)
$K = 3$	0.660 (0.016)	4.70 (0.18)	8 (6+2)
$K = 5$	0.697 (0.015)	5.20 (0.24)	8 (6+2)

TABLE A.7: Results for Different K for Data of Type Given in Figure 7.14(b)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$K = 2$	0.478 (0.026)	5.72 (0.38)	12 (6+6)
$K = 3$	0.573 (0.047)	4.85 (0.25)	12 (6+6)
$K = 5$	0.546 (0.025)	5.25 (0.23)	12 (6+6)

A.4 Expanded Decision about C in M-Partitioning Stage

TABLE A.8: Results for Different C for Data of Type Given in Figure 7.11(b)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$C = 2$	0.739 (0.025)	4.17 (0.06)	10 (4+6)
$C = 4$	0.709 (0.048)	4.25 (0.11)	10 (4+6)

TABLE A.9: Results for Different C for Data of Type Given in Figure 7.12(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$C = 2$	0.530 (0.007)	5.70 (0.20)	8 (6+2)
$C = 4$	0.517 (0.010)	6.05 (0.56)	8 (6+2)

TABLE A.10: Results for Different C for Data of Type Given in Figure 7.12(b)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$C = 2$	0.509 (0.052)	5.45 (0.26)	12 (6+6)
$C = 4$	0.461 (0.038)	5.85 (0.25)	12 (6+6)

TABLE A.11: Results for Different C for Data of Type Given in Figure 7.13(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$C = 2$	0.869 (0.024)	2.05 (0.04)	4 (2+2)
$C = 4$	0.842 (0.033)	1.96 (0.23)	4 (2+2)

TABLE A.12: Results for Different C for Data of Type Given in Figure 7.13(b)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$C = 2$	0.772 (0.044)	3.45 (0.18)	8 (2+6)
$C = 4$	0.712 (0.063)	3.73 (0.26)	8 (2+6)

TABLE A.13: Results for Different C for Data of Type Given in Figure 7.14(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$C = 2$	0.660 (0.016)	4.70 (0.18)	8 (6+2)
$C = 4$	0.715 (0.035)	5.30 (0.17)	8 (6+2)

TABLE A.14: Results for Different C for Data of Type Given in Figure 7.14(b)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
$C = 2$	0.573 (0.047)	4.85 (0.25)	12 (6+6)
$C = 4$	0.723 (0.031)	8.10 (0.20)	12 (6+6)

A.5 Decision about α_0 in Merging Stage for Chapter 7

TABLE A.15: Results for Different α_0 for Data of Type Given in Figure 7.11(b)

	Average ARI (sd)	Average No. Clusters (Main Clusters and Noise Points) (sd)	Total No.Clusters (Main Clusters and Noise Points)
$\alpha_0 = 0.5$	0.742 (0.031)	4.20 (0.11)	10 (4+6)
$\alpha_0 = 1$	0.739 (0.025)	4.17 (0.06)	10 (4+6)
$\alpha_0 = 2$	0.722 (0.024)	4.23 (0.08)	10 (4+6)

TABLE A.16: Results for Different α_0 for Data of Type Given in Figure 7.12(a)

	Average ARI (sd)	Average No. Clusters (Main Clusters and Noise Points) (sd)	Total No.Clusters (Main Clusters and Noise Points)
$\alpha_0 = 0.5$	0.541 (0.009)	5.63 (0.18)	8 (6+2)
$\alpha_0 = 1$	0.530 (0.012)	5.70 (0.20)	8 (6+2)
$\alpha_0 = 2$	0.538 (0.010)	5.66 (0.17)	8 (6+2)

TABLE A.17: Results for Different α_0 for Data of Type Given in Figure 7.12(b)

	Average ARI (sd)	Average No. Clusters (Main Clusters and Noise Points) (sd)	Total No.Clusters (Main Clusters and Noise Points)
$\alpha_0 = 0.5$	0.507 (0.005)	5.40 (0.24)	12 (6+6)
$\alpha_0 = 1$	0.509 (0.007)	5.45 (0.26)	12 (6+6)
$\alpha_0 = 2$	0.511 (0.008)	5.37 (0.27)	12 (6+6)

TABLE A.18: Results for Different α_0 for Data of Type Given in Figure 7.13(a)

	Average ARI (sd)	Average No. Clusters (Main Clusters and Noise Points) (sd)	Total No.Clusters (Main Clusters and Noise Points)
$\alpha_0 = 0.5$	0.934 (0.017)	2.10 (0.17)	4 (2+2)
$\alpha_0 = 1$	0.969 (0.024)	2.05 (0.04)	4 (2+2)
$\alpha_0 = 2$	0.942 (0.019)	2.08 (0.15)	4 (2+2)

TABLE A.19: Results for Different α_0 for Data of Type Given in Figure 7.13(b)

	Average ARI (sd)	Average No. Clusters (Main Clusters and Noise Points) (sd)	Total No.Clusters (Main Clusters and Noise Points)
$\alpha_0 = 0.5$	0.754 (0.031)	4.52 (0.16)	8 (2+6)
$\alpha_0 = 1$	0.772 (0.044)	3.45 (0.18)	8 (2+6)
$\alpha_0 = 2$	0.783 (0.022)	3.49 (0.12)	8 (2+6)

TABLE A.20: Results for Different α_0 for Data of Type Given in Figure 7.14(a)

	Average ARI (sd)	Average No. Clusters (Main Clusters and Noise Points) (sd)	Total No.Clusters (Main Clusters and Noise Points)
$\alpha_0 = 0.5$	0.643 (0.013)	4.63 (0.21)	8 (6+2)
$\alpha_0 = 1$	0.660 (0.016)	4.70 (0.18)	8 (6+2)
$\alpha_0 = 2$	0.661 (0.011)	4.59 (0.14)	8 (6+2)

TABLE A.21: Results for Different α_0 for Data of Type Given in Figure 7.14(b)

	Average ARI (sd)	Average No. Clusters (Main Clusters and Noise Points) (sd)	Total No.Clusters (Main Clusters and Noise Points)
$\alpha_0 = 0.5$	0.674 (0.036)	8.80 (0.18)	12 (6+6)
$\alpha_0 = 1$	0.723 (0.031)	8.10 (0.20)	12 (6+6)
$\alpha_0 = 2$	0.702 (0.052)	8.77 (0.24)	12 (6+6)

A.6 Expanded Simulation Results for Chapter 7

TABLE A.22: Clustering Results for Data from All Sets of the Type Given in Figure 7.11(b)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
Distributions generated from simulation set-up 1			
CSHC	0.783 (0.039)	4.37 (0.11)	10 (4+6)
SHC	0.681 (0.062)	13.47 (0.90)	10 (4+6)
Distributions generated from simulation set-up 2			
CSHC	0.534 (0.024)	7.85 (0.24)	10 (4+6)
SHC	0.541 (0.056)	14.47 (0.94)	10 (4+6)
Distributions generated from simulation set-up 3			
CSHC	0.874 (0.048)	5.50 (0.13)	10 (4+6)
SHC	0.687 (0.043)	13.67 (0.66)	10 (4+6)
Distributions generated from simulation set-up 4			
CSHC	0.368 (0.022)	8.20 (0.27)	10 (4+6)
SHC	0.566 (0.069)	14.73 (0.83)	10 (4+6)

TABLE A.23: Clustering Results for Data from All Sets of the Type Given in Figure 7.12(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
Distributions generated from simulation set-up 1			
CSHC	0.748 (0.021)	5.75 (0.13)	8 (6+2)
SHC	0.678 (0.077)	10.72 (1.90)	8 (6+2)
Distributions generated from simulation set-up 2			
CSHC	0.442 (0.033)	7.50 (0.24)	8 (6+2)
SHC	0.541 (0.089)	13.25 (1.22)	8 (6+2)
Distributions generated from simulation set-up 3			
CSHC	0.591 (0.024)	5.60 (0.17)	8 (6+2)
SHC	0.682 (0.073)	10.80 (1.35)	8 (6+2)
Distributions generated from simulation set-up 4			
CSHC	0.435 (0.040)	7.26 (0.33)	8 (6+2)
SHC	0.556 (0.078)	13.55 (1.18)	8 (6+2)

TABLE A.24: Clustering Results for Data from All Sets of the Type Given in Figure 7.12(b)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
Distributions generated from simulation set-up 1			
CSHC	0.543 (0.023)	5.85 (0.17)	12 (6+6)
SHC	0.692 (0.053)	12.15 (0.88)	12 (6+6)
Distributions generated from simulation set-up 2			
CSHC	0.289 (0.047)	8.95 (0.38)	12 (6+6)
SHC	0.522 (0.040)	14.09 (0.76)	12 (6+6)
Distributions generated from simulation set-up 3			
CSHC	0.509 (0.018)	5.35 (0.15)	12 (6+6)
SHC	0.695 (0.088)	12.38 (1.24)	12 (6+6)
Distributions generated from simulation set-up 4			
CSHC	0.249 (0.039)	9.15 (0.27)	12 (6+6)
SHC	0.508 (0.082)	13.80 (0.94)	12 (6+6)

TABLE A.25: Clustering Results for Data from All Sets of the Type Given in Figure 7.13(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
Distributions generated from simulation set-up 1			
CSHC	0.816 (0.062)	2.90 (0.35)	4 (2+2)
SHC	0.571 (0.085)	8.10 (1.13)	4 (2+2)
Distributions generated from simulation set-up 2			
CSHC	0.544 (0.054)	5.90 (0.36)	4 (2+2)
SHC	0.435 (0.071)	10.55 (1.05)	4 (2+2)
Distributions generated from simulation set-up 3			
CSHC	0.840 (0.044)	3.10 (0.23)	4 (2+2)
SHC	0.589 (0.081)	7.98 (0.89)	4 (2+2)
Distributions generated from simulation set-up 4			
CSHC	0.549 (0.034)	6.50 (0.22)	4 (2+2)
SHC	0.449 (0.092)	10.49 (1.50)	4 (2+2)

TABLE A.26: Clustering Results for Data from All Sets of the Type Given in Figure 7.13(b)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
Distributions generated from simulation set-up 1			
CSHC	0.750 (0.044)	4.78 (0.29)	8 (2+6)
SHC	0.577 (0.096)	11.17 (0.85)	8 (2+6)
Distributions generated from simulation set-up 2			
CSHC	0.441 (0.036)	10.05 (0.17)	8 (2+6)
SHC	0.434 (0.071)	13.85 (1.37)	8 (2+6)
Distributions generated from simulation set-up 3			
CSHC	0.771 (0.049)	4.15 (0.22)	8 (2+6)
SHC	0.574 (0.085)	10.90 (0.45)	8 (2+6)
Distributions generated from simulation set-up 4			
CSHC	0.456 (0.030)	10.45 (0.36)	8 (2+6)
SHC	0.437 (0.069)	13.75 (1.72)	8 (2+6)

TABLE A.27: Clustering Results for Data from All Sets of the Type Given in Figure 7.14(a)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
Distributions generated from simulation set-up 1			
CSHC	0.497 (0.025)	5.00 (0.16)	8 (6+2)
SHC	0.562 (0.062)	10.73 (1.20)	8 (6+2)
Distributions generated from simulation set-up 2			
CSHC	0.481 (0.042)	6.05 (0.32)	8 (6+2)
SHC	0.411 (0.092)	12.16 (1.11)	8 (6+2)
Distributions generated from simulation set-up 3			
CSHC	0.505 (0.034)	5.15 (0.17)	8 (6+2)
SHC	0.573 (0.056)	10.89 (1.26)	8 (6+2)
Distributions generated from simulation set-up 4			
CSHC	0.405 (0.067)	6.20 (0.45)	8 (6+2)
SHC	0.455 (0.074)	11.85 (1.28)	8 (6+2)

TABLE A.28: Clustering Results for Data from All Sets of the Type Given in Figure 7.14(b)

	Average ARI (sd)	Average No. Clusters (sd)	Total No.Clusters (Main Clusters and Noise Points)
Distributions generated from simulation set-up 1			
CSHC	0.723 (0.031)	8.10 (0.20)	12 (6+6)
SHC	0.645 (0.053)	13.18 (0.98)	12 (6+6)
Distributions generated from simulation set-up 2			
CSHC	0.348 (0.011)	10.25 (0.42)	12 (6+6)
SHC	0.445 (0.031)	14.25 (0.47)	12 (6+6)
Distributions generated from simulation set-up 3			
CSHC	0.632 (0.033)	8.55 (0.35)	12 (6+6)
SHC	0.656 (0.060)	13.01 (0.78)	12 (6+6)
Distributions generated from simulation set-up 4			
CSHC	0.465 (0.012)	9.55 (0.46)	12 (6+6)
SHC	0.470 (0.048)	14.31 (0.58)	12 (6+6)

A.7 Histogram of K in Glasgow Housing Market CPEP Data in Chapter 7

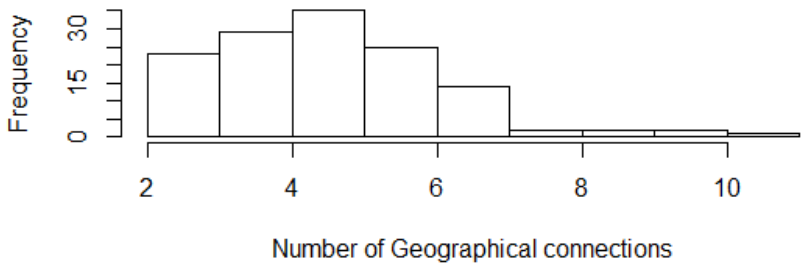


FIGURE A.1: Number of Geographical Connections for Alll Intermediate Zones

From the simulation results of K in Section 7.3.3, it is better to choose a K not too large or small.

A.8 Expanded Simulation Results for Uniform Distribution in Chapter 8

A.9 Expanded Simulation Results for Multivariate Data in Chapter 8

TABLE A.29: Expanded Uniform Distribution Sensitivity Comparison

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Distribution Set 2[a]	0.735 (0.083)	5.360 (0.337)	2.700 (0.988)	8.060	6 (4+2)	1.000 (0.000)	0.821 (0.232)
Distribution Set 2[b]	0.682 (0.069)	5.810 (0.390)	2.130 (0.762)	7.942	6 (4+2)	0.943 (0.060)	0.744 (0.216)
Distribution Set 2[c]	0.641 (0.071)	5.260 (0.362)	2.660 (0.733)	7.920	6(4+2)	1.000 (0.000)	0.835 (0.108)
Distribution Set 2[d]	0.685 (0.080)	5.910 (0.413)	2.170 (0.524)	8.081	6(4+2)	0.948 (0.065)	0.781 (0.154)

[a] Range 1: the minimum to the maximum
[b] Range 2: from the minimum-1 to the maximum
[c] Range 3: from the minimum to the maximum+1
[d] Range 4: from the minimum-1 to the maximum+1

TABLE A.30: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 1 Located in Figure 7.11(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.894 (0.011)	4.000 (0.000)	6.300 (0.596)	10.300	10 (4+6)	1.000 (0.000)	0.960 (0.078)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.861 (0.062)	-	-	9.870 (0.512)	10 (4+6)	-	-
Spatial Hierarchical Clustering	0.681 (0.062)	-	-	13.470 (0.900)	10 (4+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.783 (0.039)	-	-	14.370 (0.110)	10 (4+6)	-	-

TABLE A.31: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 2 Located in Figure 7.11(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.825 (0.090)	4.733 (2.333)	6.733 (1.258)	11.466	10 (4+6)	1.000 (0.000)	0.914 (0.129)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.795 (0.112)	-	-	9.850 (1.035)	10 (4+6)	-	-
Spatial Hierarchical Clustering	0.541 (0.056)	-	-	14.470 (0.094)	10 (4+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.534 (0.024)	-	-	7.850 (0.240)	10 (4+6)	-	-

TABLE A.32: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 3 Located in Figure 7.11(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.897 (0.006)	4.000 (0.000)	6.200 (0.407)	10.200	10 (4+6)	1.000 (0.000)	0.971 (0.058)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.863 (0.009)	-	-	9.241 (0.506)	10 (4+6)	-	-
Spatial Hierarchical Clustering	0.687 (0.043)	-	-	13.670 (0.660)	10 (4+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.874 (0.048)	-	-	5.500 (0.130)	10 (4+6)	-	-

TABLE A.33: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 4 Located in Figure 7.11(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.735 (0.066)	4.433 (0.679)	16.833 (3.983)	21.266	10 (4+6)	1.000 (0.000)	0.971 (0.058)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.686 (0.088)	-	-	12.330 (1.570)	10 (4+6)	-	-
Spatial Hierarchical Clustering	0.566 (0.069)	-	-	14.730 (0.830)	10 (4+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.368 (0.022)	-	-	8.200 (0.270)	10 (4+6)	-	-

TABLE A.34: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 1 Located in Figure 7.12(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.894 (0.022)	6.033 (0.183)	2.167 (0.379)	8.200	8 (6+2)	1.000 (0.000)	0.960 (0.078)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.834 (0.021)	-	-	7.770 (0.566)	8 (6+2)	-	-
Spatial Hierarchical Clustering	0.678 (0.077)	-	-	10.720 (1.900)	8 (6+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.748 (0.021)	-	-	5.750 (0.130)	8 (6+2)	-	-

TABLE A.35: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 2 Located in Figure 7.12(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.776 (0.072)	7.333 (1.807)	2.933 (1.230)	10.266	8 (6+2)	1.000 (0.000)	0.777 (0.246)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.760 (0.068)	-	-	7.780 (1.180)	8 (6+2)	-	-
Spatial Hierarchical Clustering	0.541 (0.089)	-	-	13.250 (1.220)	8 (6+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.442 (0.033)	-	-	7.500 (0.240)	8 (6+2)	-	-

TABLE A.36: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 3 Located in Figure 7.12(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.896 (0.015)	6.033 (0.183)	2.400 (0.814)	8.433	8 (6+2)	1.000 (0.000)	0.893 (0.190)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.848 (0.052)	-	-	7.569 (0.689)	8 (6+2)	-	-
Spatial Hierarchical Clustering	0.682 (0.073)	-	-	10.800 (1.350)	8 (6+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.591 (0.024)	-	-	5.600 (0.170)	8 (6+2)	-	-

TABLE A.37: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 4 Located in Figure 7.12(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.614 (0.209)	9.633 (2.526)	18.533 (7.157)	28.166	8 (6+2)	1.000 (0.000)	0.130 (0.065)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.724 (0.111)	-	-	7.991 (1.315)	8 (6+2)	-	-
Spatial Hierarchical Clustering	0.556 (0.078)	-	-	13.550 (1.180)	8 (6+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.435 (0.040)	-	-	7.260 (0.330)	8 (6+2)	-	-

TABLE A.38: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 1 Located in Figure 7.12(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.869 (0.066)	6.467 (0.819)	6.367 (0.718)	12.834	12 (6+6)	1.000 (0.000)	0.952 (0.088)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.727 (0.110)	-	-	10.710 (1.131)	12 (6+6)	-	-
Spatial Hierarchical Clustering	0.692 (0.053)	-	-	12.150 (0.880)	12 (6+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.543 (0.023)	-	-	5.850 (0.170)	12 (6+6)	-	-

TABLE A.39: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 2 Located in Figure 7.12(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.749 (0.098)	7.233 (1.073)	6.900 (0.960)	14.133	12 (6+6)	1.000 (0.000)	0.885 (0.115)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.650 (0.128)	-	-	10.811 (2.051)	12 (6+6)	-	-
Spatial Hierarchical Clustering	0.522 (0.040)	-	-	14.090 (0.760)	12 (6+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.289 (0.047)	-	-	8.950 (0.380)	12 (6+6)	-	-

TABLE A.40: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 3 Located in Figure 7.12(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.893 (0.007)	6.000 (0.000)	6.233 (0.430)	12.233	12 (6+6)	1.000 (0.000)	0.967 (0.061)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.783 (0.121)	-	-	9.110 (1.514)	12 (6+6)	-	-
Spatial Hierarchical Clustering	0.695 (0.088)	-	-	12.380 (1.240)	12 (6+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.509 (0.018)	-	-	5.350 (0.150)	12 (6+6)	-	-

TABLE A.41: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 4 Located in Figure 7.12(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.721 (0.161)	8.367 (1.752)	17.533 (5.469)	25.900	12 (6+6)	1.000 (0.000)	0.378 (0.131)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.666 (0.170)	-	-	10.729 (2.145)	12 (6+6)	-	-
Spatial Hierarchical Clustering	0.508 (0.082)	-	-	13.800 (0.940)	12 (6+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.249 (0.039)	-	-	9.150 (0.270)	12(6+6)	-	-

TABLE A.42: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 1 Located in Figure 7.13(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.597 (0.042)	3.433 (1.104)	3.967 (2.539)	7.400	4 (2+2)	1.000 (0.000)	0.652 (0.285)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.598 (0.079)	-	-	4.100 (0.53)	4 (2+2)	-	-
Spatial Hierarchical Clustering	0.571 (0.085)	-	-	8.100 (1.130)	4 (2+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.816 (0.062)	-	-	2.900 (0.350)	4 (2+2)	-	-

TABLE A.43: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 2 Located in Figure 7.13(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.598 (0.070)	3.967 (1.377)	6.133 (3.646)	10.100	4 (2+2)	1.000 (0.000)	0.437 (0.245)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.585 (0.091)	-	-	5.741 (0.625)	4 (2+2)	-	-
Spatial Hierarchical Clustering	0.435 (0.071)	-	-	10.550 (1.050)	4 (2+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.544 (0.054)	-	-	5.900 (0.360)	4 (2+2)	-	-

TABLE A.44: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 3 Located in Figure 7.13(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.606 (0.053)	3.367 (1.426)	4.300 (3.725)	7.676	4 (2+2)	1.000 (0.000)	0.667 (0.135)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.593 (0.054)	-	-	3.877 (0.813)	4 (2+2)	-	-
Spatial Hierarchical Clustering	0.589 (0.081)	-	-	7.890 (0.890)	4 (2+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.840 (0.044)	-	-	3.100 (0.230)	4 (2+2)	-	-

TABLE A.45: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 4 Located in Figure 7.13(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.351 (0.061)	2.433 (1.194)	30.600 (4.606)	33.033	4 (2+2)	1.000 (0.000)	0.067 (0.010)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.463 (0.085)	-	-	5.581 (1.155)	4 (2+2)	-	-
Spatial Hierarchical Clustering	0.449 (0.092)	-	-	10.490 (1.500)	4 (2+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.549 (0.034)	-	-	6.500 (0.220)	4 (2+2)	-	-

TABLE A.46: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 1 Located in Figure 7.13(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.710 (0.054)	5.433 (1.977)	2.600 (0.968)	8.033	8 (2+6)	1.000 (0.000)	0.844 (0.219)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.671 (0.077)	-	-	6.275 (1.154)	8 (2+6)	-	-
Spatial Hierarchical Clustering	0.577 (0.096)	-	-	11.170 (0.850)	8 (2+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.750 (0.044)	-	-	4.780 (0.290)	8 (2+6)	-	-

TABLE A.47: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 2 Located in Figure 7.13(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.468 (0.072)	6.067 (1.893)	3.100 (1.522)	9.193	8 (2+6)	1.000 (0.000)	0.733 (0.251)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.403 (0.086)	-	-	7.444 (1.643)	8 (2+6)	-	-
Spatial Hierarchical Clustering	0.434 (0.071)	-	-	13.850 (1.370)	8 (2+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.441 (0.036)	-	-	10.050 (0.170)	8 (2+6)	-	-

TABLE A.48: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 3 Located in Figure 7.13(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.699 (0.037)	5.100 (1.494)	2.400 (0.724)	7.600	8 (2+6)	1.000 (0.000)	0.889 (0192)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.608 (0.055)	-	-	6.521 (1.352)	8 (2+6)	-	-
Spatial Hierarchical Clustering	0.574 (0.015)	-	-	10.900 (0.450)	8 (2+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.771 (0.049)	-	-	4.150 (0.220)	8 (2+6)	-	-

TABLE A.49: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 4 Located in Figure 7.13(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.426 (0.091)	6.767 (2.635)	27.167 (7,746)	33.934	8 (2+6)	1.000 (0.000)	0.079 (0.021)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.474 (0.081)	-	-	10.661 (1.428)	8 (2+6)	-	-
Spatial Hierarchical Clustering	0.437 (0.069)	-	-	13.750 (1.720)	8 (2+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.456 (0.030)	-	-	10.450 (0.360)	8 (2+6)	-	-

TABLE A.50: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 1 Located in Figure 7.14(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.639 (0.040)	2.800 (1.375)	12.300 (5.004)	15.100	8 (6+2)	1.000 (0.000)	0.575 (0.228)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.639 (0.088)	-	-	6.718 (1.213)	8 (6+2)	-	-
Spatial Hierarchical Clustering	0.562 (0.062)	-	-	10.730 (1.200)	8 (6+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.497 (0.025)	-	-	5.000 (0.160)	8 (6+2)	-	-

TABLE A.51: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 2 Located in Figure 7.14(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.628 (0.043)	3.433 (1.977)	10.833 (2.793)	13.266	8 (6+2)	1.000 (0.000)	0.587 (0.140)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.589 (0.084)	-	-	7.911 (1.137)	8 (6+2)	-	-
Spatial Hierarchical Clustering	0.411 (0.092)	-	-	12.160 (1.110)	8 (6+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.481 (0.042)	-	-	6.050 (0.320)	8 (6+2)	-	-

TABLE A.52: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 3 Located in Figure 7.14(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.628 (0.038)	2.733 (1.530)	13.333(5.168)	16.066	8 (6+2)	1.000 (0.000)	0.541 (0.250)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.682 (0.034)	-	-	6.468 (1.135)	8 (6+2)	-	-
Spatial Hierarchical Clustering	0.573 (0.056)	-	-	10.890 (1.260)	8 (6+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.505 (0.034)	-	-	5.150 (0.170)	8 (6+2)	-	-

TABLE A.53: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 4 Located in Figure 7.14(a)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.457 (0.074)	2.800 (1.690)	27.000 (3.833)	29.800	8 (6+2)	1.000 (0.000)	0.226 (0.028)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.429 (0.055)	-	-	7.233 (1.511)	8 (6+2)	-	-
Spatial Hierarchical Clustering	0.455 (0.074)	-	-	11.850 (1.280)	8 (6+2)	-	-
Chameleon Spatial Hierarchical Clustering	0.405 (0.067)	-	-	6.200 (0.450)	8 (6+2)	-	-

TABLE A.54: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 1 Located in Figure 7.14(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.728 (0.048)	4.867 (1.306)	6.300 (0.596)	11.167	12 (6+6)	1.000 (0.000)	0.960 (0078)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.626 (0.083)	-	-	8.006 (1.623)	12 (6+6)	-	-
Spatial Hierarchical Clustering	0.645 (0.053)	-	-	12.180 (0.980)	12 (6+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.732 (0.031)	-	-	8.100 (0.200)	12 (6+6)	-	-

TABLE A.55: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 2 Located in Figure 7.14(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.639 (0.076)	7.033 (2.833)	7.067 (1.081)	14.100	12 (6+6)	1.000 (0.000)	0.867 (0.125)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.528 (0.059)	-	-	12.077 (1.360)	12 (6+6)	-	-
Spatial Hierarchical Clustering	0.445 (0.031)	-	-	14.250 (0.470)	12 (6+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.348 (0.011)	-	-	10.250 (0.420)	12 (6+6)	-	-

TABLE A.56: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 3 Located in Figure 7.14(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.723 (0.044)	4.967 (1.377)	6.600 (0.855)	11.567	12 (6+6)	1.000 (0.000)	0.922 (0.105)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.673 (0.074)	-	-	8.124 (1.315)	12 (6+6)	-	-
Spatial Hierarchical Clustering	0.656 (0.060)	-	-	13.010 (0.780)	12 (6+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.632 (0.033)	-	-	8.550 (0.350)	12 (6+6)	-	-

TABLE A.57: Summary of ARI, $G(H+J)$, TPR and TDR of Data from Distribution Set 4 Located in Figure 7.14(b)

Methods	Average Adjusted Rand Index (sd)	Average Estimated H(sd) Total Estimated Main Clusters	Average Estimated J (sd) Total Estimated Noise Points	Estimated Total No. Clusters G $H+J$(sd)	Actual Total No.Clusters $H+J$ (sd)	Average TPR (sd)	Average TDR (sd)
Spatially Constrained Finite Mixture Model with Noise Distribution	0.525 (0.092)	7.233 (2.788)	25.300 (8.065)	32.533	12 (6+6)	1.000 (0.000)	0.267 (0.110)
Spatially Constrained Finite Mixture Model with Prior Distribution	0.539 (0.044)	-	-	11.830 (2.311)	12 (6+6)	-	-
Spatial Hierarchical Clustering	0.470 (0.048)	-	-	14.310 (0.580)	12 (6+6)	-	-
Chameleon Spatial Hierarchical Clustering	0.465 (0.012)	-	-	9.550 (0.460)	12 (6+6)	-	-

A.10 Expanded Univariate Simulation Results for Chapter 9

TABLE A.58: Summary of ARI and BIC Based on Different Number of Clusters by Using 4 Groups Univariate Data

Fitted No.Clusters	Distribution Sets 1	Distribution Sets 2	Distribution Sets 3	Distribution Sets 4
Locations from Figure in 7.11(b) ARI Based on the Minimal BIC	0.905	0.789	0.893	0.791
Number of Clusters By Using BIC	9	10	9	12
True Number of Clusters	10	10	10	10
Locations from Figure in 7.12(a) ARI Based on the Minimal BIC	0.874	0.710	0.882	0.688
Number of Clusters By Using BIC	8	8	8	8
True Number of Clusters	8	8	8	8
Locations from Figure in 7.12(b) ARI Based on the Minimal BIC	0.861	0.643	0.846	0.682
Number of Clusters By Using BIC	11	12	10	10
True Number of Clusters	12	12	12	12
Locations from Figure in 7.13(a) ARI Based on the Minimal BIC	0.633	0.551	0.648	0.525
Number of Clusters By Using BIC	6	7	5	7
True Number of Clusters	4	4	4	4

TABLE A.59: Summary of ARI and BIC Based on Different Number of Clusters by Using 4 Groups Univariate Data

Fitted No.Clusters	Distribution Sets 1	Distribution Sets 2	Distribution Sets 3	Distribution Sets 4
Locations from Figure in 7.13(b) ARI Based on the Minimal BIC	0.707	0.539	0.721	0.565
Number of Clusters By Using BIC	8	7	9	8
True Number of Clusters	8	8	8	8
Locations from Figure in 7.14(a) ARI Based on the Minimal BIC	0.677	0.494	0.655	0.430
Number of Clusters By Using BIC	8	9	7	8
True Number of Clusters	8	8	8	8
Locations from Figure in 7.14(b) ARI Based on the Minimal BIC	0.797	0.650	0.762	0.647
Number of Clusters By Using BIC	11	12	9	13
True Number of Clusters	12	12	12	12

Bibliography

[1] 4 reasons to invest in Glasgow property: Price growth. https://www.selectproperty.com/2015/05/4-reasons-to-invest-in-glasgow-property-price-growth/. [Accessed: 2018-05-04].

[2] Average house price at 7.6 times annual salary, official figures show. https://www.theguardian.com/money/2017/mar/17/average-house-price-times-annual-salary-official-figures-ons. [Accessed: 2018-05-04].

[3] Best Areas of Glasgow To Live 2018. https://www.removalservicesscotland.co.uk/blog/best-areas-of-glasgow-to-live. [Accessed: 2018-12-06].

[4] Glossary of terms. Technical report, National Records of Scotland.

[5] Gradient descent. https://en.wikipedia.org/wiki/Gradientdescent. [Accessed: 2018-05-19].

[6] Prior probability. https://en.wikipedia.org/wiki/Priorprobability. [Accessed: 2018-05-19].

[7] sample size is less than 30. https://www.johndcook.com/blog/normalapproxtot/. [Accessed: 2018-05-12].

[8] Scottish Neighbour Statistics. http://statistics.gov.scot. [Accessed: 2018-05-14].

[9] Adair, A. and Berry, J. N. (1996). Hedonic modelling housing markets and residential valuation. *Journal of Property Research*, pages 67–83.

[10] adunaic (https://stats.stackexchange.com/users/13409/adunaic). Supervised clustering or classification? Cross Validated. URL:https://stats.stackexchange.com/q/39107 (version: 2012-10-10).

[11] Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. *Annals of the Institute of Statistical Mathematics*.

[12] Allard, D. and Fraley, C. (1997). Non-parametric maximum likelihood estimation of features in spatial point processes using Voronoi tesselation. *Journal of the American Statistical Association*, 92:1485–1493.

[13] Anderson, C., Lee, D., and Dean, N. (2014). Identifying clusters in Bayesian disease mapping. *Biostatistics*, pages 457–469.

[14] Arora, N., Sandeep, J., and Verma, S. K. (2006). Range clustering: an algorithm for empirical evaluation of classical clustering algorithms. *IEEE*, pages 246–263.

[15] Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821.

[16] Bayes, T. (1764). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418.

[17] Beatty, M. and Manjunath, B. (1997). Dimensionality reduction using Multi-Dimensional Scaling for content-based retrieval. *IEEE*, pages 835–838.

[18] Blekas, K., Likas, A., Galatsanos, N. P., and Lagaris, I. E. (2005). A spatially constrained mixture model for image segmentation. *IEEE*, 16:494–498.

[19] Bordley, R. F. (1985). Relating cross-elasticities to first choice/second choice data. *Journal of Business and Economic Statistics*.

[20] Browne, W. M. (1967). On oblique Procrustes rotation. *Psychometrika*, 32:125–132.

[21] Byers, S. and Raftery, A. E. (1998). Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association*, 93:577–584.

[22] Calinski, T. B. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3:1–27.

[23] Carlos, R. E. and Walker, S. G. (2012). Label switching in Bayesian mixture models: deterministic relabeling strategies. *Journal of Computational and Graphical Statistics*, 23:25–45.

[24] Casella, G. (1985). An introduction to empirical bayes data analysis. *American Statistician. American Statistical Association*, page 83–87.

[25] Cauchy, A. L. (1821). Sur les formules qui résultent de l'emploie du signe et sur ou, et sur les moyennes entre plusieurs quantités. *Cours d'Analyse*, 3:373–377.

[26] Cox, R. W. and Talyor, P. A. (2017). Stability of spatial smoothness and cluster-size threshold estimates in FMRI. Technical report, National Institute of Mental Health.

[27] Cuesta-Albertos, J. A., Gordaliza, A., and Matran, C. (1997). Trimmed k-means: an attempt to robustify quantizers. *The Annuals of Statistics*, 25:552–576.

[28] Day, B. (2003). Submarket identification in property markets: a hedonic housing price model for Glasgow. *Economic and Social Research*.

[29] Dean, S. K. (1977). Cross-sectional time-series experiments: Some suggested statistical analyses. *Psychological Bulletin*, page 489.

[30] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Royal Statistical Society*.

[31] Deza, E. and Deza, M. M. (2009). Encyclopedia of distances. *Springer*.

[32] Diggle, P. J. and Ribeiro Jr, P. J. (2007). *Model-based Geostatistics*. Springer.

[33] Donald, R. and Andrew, G. (2008). *Bayesian Methods for Data Analysis*. CRC Press.

[34] Everitt, B. S. and Hand, D. J. (1981). *Finite mixture distributions*. Chapman and Hall.

[35] Everitt, B. S., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster analysis*. Wiley Series in Probability and Statistics, 5 edition.

[36] Fiduccia, C. M. and Mattheyses, R. (1982). *A linear-time heuristic for improving network partitions*. IEEE.

[37] Findley, D. F. (1991). Counter examples to parsimony and BIC. *Annals of the Institute of Statistical Mathematics*, 43:505–514.

[38] Fraley, C. and Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. Technical report, University of Washington.

[39] Fraley, C. and Raftery, A. E. (2007). Bayesian regularization for Normal mixture estimation and model-based clustering. *Journal of Classification*, pages 155–181.

[40] Galton, F. (1886). Regression towards mediocrity in hereditary stature. *Journal of the Anthropological Institute of Great Britain and Ireland*, pages 246–263.

[41] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE*.

[42] Geman, S. and McClure, D. E. (1985). Bayesian image analysis: an application to single photon emission tomography. *Statistical computing section*, pages 12–18.

[43] Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). *Markov chain Monte Carlo in practice*. Chapman & Hall.

[44] Golub, G. and Van Loan, C. (1996). *Matrix Computations*. Baltimore, MD: Johns Hopkins.

[45] Good, P. and Hardin, J. (2012). Common errors in statistics (and how to avoid them). *John Wiley & Sons*, pages 129–131.

[46] Gopal, S. and Hebert, T. J. (1998). Bayesian pixel classification using spatially variant finite mixtures and the generalized EM algorithm. *IEEE*, 7:1014–1028.

[47] Grigsby, W. G. (1963). *Housing markets and public policy*. University of Pennsylvania Press.

[48] Guidolin, M. and Timmermann, A. (2002). International asset allocation under regime switching. *The Reviews of Financial Studies*, 15:1137–1187.

[49] Hammersley, J. M. and Clifford, P. E. (1971). Markov random fields on finite graphs and lattices. *Unpublished manuscript*.

[50] Harris, J. M. (2000). *Combinatorics and Graph Theory*. Springer.

[51] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*.

[52] Hees, J. (2016). Scipy hierarchical clustering and dendrogram tutorial. Technical report, German Research Center for Artificial Intelligence (DFKI).

[53] Henning, C. and Liao, T. F. (2013). How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society*, 62:309–369.

[54] Herd, L. and Sabanes Bove, D. (2014). *Likelihood*. Springer.

[55] Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New Phytologist*, 11:37–50.

[56] Jefferys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London*, 186:453–461.

[57] Johann, P. (1994). Parametric statistical theory. *Walter de Gruyter*, pages 207–208.

[58] Karypis, G., Aggarwal, R., and Kumar, V. (1999a). Multilevel hypergraph partitioning: applications in VLSI domain. *IEEE*, 7:69–79.

[59] Karypis, G., Han, E. H., and Kumar, V. (1999b). Chameleon: hierchical clustering using dynamic modeling. *Computer*, 32:68–75.

[60] Karypis, G. and Kumar, V. (1998a). A fast and high quality multilevel scheme for partitioning irregular graphs. *Society for Industrial and Applied Mathematics*, 20:359–392.

[61] Karypis, G. and Kumar, V. (1998b). Multilevel k-way hypergraph partitioning. Technical report, University of Minnesota.

[62] Karypis, G. and Kumar, V. (1998c). Multilevel k-way partitionin scheme for irregular graphs. *Journal of parallel and distributed computing*, 48:96–129.

[63] Kaufman, L. and Rousseeuw, P. J. (1990). *Finding groups in Data: an introduction to cluster analysis*. Wiley Series in Probability and Statistics.

[64] Kernighan, B. W. and Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307.

[65] Kindermann, R. and Snell, J. L. (1980). Markov random fields and their applications. *American Mathematical Society*.

[66] Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data*. Springer, Boston, USA.

[67] Konrad, Z. (1972). Der plankalkül. *Konrad Zuse Internet Archive*, pages 96–105.

[68] Kranowski, W. (2000). *Principals of Multivariate Analysis*. Oxford University Press.

[69] Ladroue, C. (2017). Graph bisection in R. Technical report, University of California, Berkeley.

[70] Laplace, P. S. (1774). *Memoir on the probability of causes of events*. Memoires de Mathématique et de Physique.

[71] Lawrence, H. and Phipps, A. (1985). Comparing partitions. *Journal of Classification*, (2):193–218.

[72] Lee, P. M. (2004). *Bayesian Statistics, an introduction*. Wiley.

[73] Luenberger, D. G. (1984). *Linear and Nonlinear Programming*. Addison-Wesley.

[74] Lukaszewez, F. K. and L., P. (1951). Sur la liaison et la division des points d'un ensemble fini. *Colloquium Mathematicium*, 2:282–285.

[75] Mackay, D. J. (1998). *Introduction to Monte Carlo methods*. Springer Netherlands.

[76] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297.

[77] Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2:49–55.

[78] Marsden, J. (2015). Housing prices in London − an economic analysis of London's housing market. *Greater London Authority Economics*.

[79] McLachlan, G. and Peel, D. (2001). *Finite Mixture Models*. Wiley Series in Probability and Statistics.

[80] McLachlan, G. J. and Krishnan, T. (1997). *The EM algorithm and extensions*. Wiley Series in Probability and Statistics.

[81] Metropolis, N. and Rosenbluth, A. W. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*.

[82] Neyman, J. (1936). Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society of London*, 236:333–380.

[83] Nikou, C., Galatsanos, N., and Likas, A. (1998). A class-adaptive spatially variant mixture model for image segmentation. Technical report, University of Ioannina.

[84] Nocedal, J. and Wright, S. J. (1999). *Numerical Optimization*. Springer.

[85] of Scotland, N. R. (2017). *Scottish Government Population Estimates (Current Geographic Boundaries)*.

[86] Oh, M.-S. and Raftery, A. E. (2001). Bayesian multidimensional scaling and choice of dimension. *Journal of the American Statistical Association*, 96(455):1031–1044.

[87] Oh, M.-S. and Raftery, A. E. (2007). Model-based clustering with dissmiliarities: A Bayesian approach. *Journal of the American Statistical Association*, 16(3):559–585.

[88] Orban, A. (2004). *Approximation of the minimum bisection and the hardware-software partitioning problem*. PhD thesis, Eotvos Lorand University.

[89] Papastamoulis, P. (2015). An R package for dealing with the label switching problem in McMC outputs. *Journal of Statistical Software*.

[90] Patrycja, K. R. (2014). An application of cluster analysis on the Polish housing market. In *The 8th International Days of Statistics and Economics*.

[91] Pearson, K. (1894). Contributions to the theory of mathematical evolution. *Philosophical Transcations of the Royal Society of London*, 185:71–110.

[92] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, pages 559–572.

[93] Procedure, T. C. (2009). Cluster methods. Technical report, SAS Institute.

[94] Pryce, G. (2009). Dwelling substitutability and the delineation of submarkets. In *Centre for Public Policy for Regions*.

[95] p.s. (https://math.stackexchange.com/user/1014356). What is the difference between gradient descent and newton's gradient descent? Mathematics Stack Exchange. URL:https://stackoverflow.com/questions/12066761 (version: 2012-11-30).

[96] p.s. (https://math.stackexchange.com/users/17433/p s). What is the difference between projected gradient descent and ordinary gradient descent? Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/572664 (version: 2013-11-20).

[97] Raftery, A. E. (1998). Bayes factors and BIC comment on weakliem. Technical report, University of Washington.

[98] Raiffa, H. and Schlaifer, R. (1961). Applied statistical decision theory. *Division of Research*.

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

[100] Rapkin, M. C. (1953). *Housing market analysis*. Housing and home finance agency.

[101] Ribeiro, P. J. and Diggle, P. J. (2006). *Model-based geostatistics*. Springer Series in Statistics.

[102] Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk metropolis algorithms. *The Annals of Applied Probability*, pages 110–120.

[103] Rokach, L. and Oded, M. (2005). Data mining and knowledge discovery handbook. *Springer US*, pages 321–352.

[104] Ross, A. (2016). Procrustes analysis. Technical report, University of South Carolina.

[105] Smyth, P. (2016). Note set 4: Finite mixture models and the em algorithm. Technical report, University of California.

[106] Sokal, R. R. and Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438.

[107] Sorensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on simiularity of species content and its application to analysis of the vegetation on Danish commons. *Biologiske Skrifter*, 5:1–34.

[108] Spiliopoulou, M., Schmidt-Thieme, L., and Janning, R. (2014). *Data Analysis, Machine Learning and Knowledge Discovery*. Springer.

[109] Stuart, G. and Donald, G. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE*, 6:721–741.

[110] Thorndike, R. L. (1953). Who belongs in the family. *Psychometrika*, 18:267–276.

[111] Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via gap statistic. *Royal Statistical Society*, 63:411–423.

[112] Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, 17:401–419.

[113] Tzeng, J., Lu, H. H.-S., and Li, W.-H. (2008). Multidimensional scaling for large genomic data sets. *BMC Bioinformatics*, 9.

[114] Ward, J. H. (1963a). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, pages 236–244.

[115] Ward, J. H. (1963b). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244.