

LOCAL INDICATORS OF SPATIAL ASSOCIATION FOR 2D AND 3D POINT PATTERNS

Weliwitage Jithmi Jannidi

Master's Thesis in Statistics

Supervised by:

Prof. Dr. Claudia Redenbach

Technische Universität Kaiserslautern

Fachbereich Mathematik

AG Statistik



September, 2020

Declaration

I declare that this thesis is the outcome of the independent study carried out by myself under the supervision and guidance of Prof.Dr. Claudia Redenbach. All the information used in this work is referenced and all the sources are provided in the Bibliography.

Weliwitage Jithmi Jannidi
Kaiserslautern,DE.

Abstract

This thesis project is carried out to treat clustered spatial point pattern data. The main objective of this study is applying local indicators of spatial association to identify the cluster points of a point pattern and these points belong to clusters are called as features through this study. Localized K-functions and product density functions and nearest neighbour distances were applied in cluster methods which are hierarchical clustering, EM algorithm and stochastic EM algorithm to extract features.

The similar research studies done for the 2D spatial point patterns are examined using simulated point process and then the methods are implemented to apply for 3D spatial point patterns.

Simulated 2D and 3D point patterns are analyzed using the above mentioned methods and finally the results of the methods are compared to check their accuracy.

The package "**spatstat**" built for spatial point patterns in **R** statistical software is used for programming the methods and analyzing the data.

Keywords: Point Process Statistics, Spatial Statistics, Clustering, LISA, Local Indicators of Spatial Association, K-function, EM algorithm, SEM, NN distances, HCA , Hierarchical Cluster Analysis, Stochastic EM

Acknowledgements

I would like to offer my gratitude to all those who gave me the strength to complete this thesis.

First and foremost, I would like to express my greatest and heartfelt grace to Prof. Dr.Claudia Redenbach for supervising me. Furthermore I would like to thank my professor for giving me the perfect knowledge of Spatial Statistics, clear instructions, great encouragement and enough freedom to work on my own way. Without her guidance and persistent helps this dissertation would not have been possible.

Then I would like to acknowledge this thesis to Jenifer Barriero and all other people from Materials Science Department at Saarbrücken University for their support me by giving data for this thesis work.

I also offer my special gratitude to all members of the statistics group for their precious advice and support. Finally I thank to my family and all of my friends who encouraged me to ahead with my thesis.

Contents

Declaration	ii
Abstract	iii
Acknowledgements	iv
List of Figures	vii
List of Tables	x
1 Introduction	1
2 Basics of Spatial Point Processes	3
2.1 Point Patterns	3
2.1.1 Definition	5
2.2 Methods of Statistics	7
2.2.1 Intensity	8
2.2.2 Variance Covariance	9
2.2.3 Correlation	10
2.2.3.a Fry plot	10
2.2.4 Edge effects	11
2.2.5 K - function	12
2.2.6 Pair correlation function	13
2.2.7 Product density function	14
2.2.8 Nearest neighbour function G	15

3 Cluster Point Process	17
3.1 nnclean	18
4 Methodology	22
4.1 Introduction	22
4.1.1 Local Indicators of Spatial Association	22
4.2 Product Density LISA Functions	25
4.2.1 Two Dimensional Point Processes	25
4.2.1.a Translation correction	27
4.2.1.b Example	28
4.2.2 3D Point Processes	29
4.2.2.a Example	31
4.3 Clustering of LISA functions	33
4.3.1 Hierarchical Cluster Analysis (HCA)	33
4.3.1.a HCA of LISA functions without covariances	38
4.3.1.b HCA of LISA functions with covariances	41
4.3.2 EM algorithm	48
4.3.2.a NN distances and EM algorithm	49
4.3.3 Method of SEM	53
5 Comparison of HCA, EM and SEM	65
6 Conclusions and Final remarks	78
Bibliography	81
Appendix A	83
A.1 List of Acronyms and Abbreviations	83
Appendix B	84
B.1 List of Terms	84

List of Figures

1.1	Nearest neighbour cleaning for simulated point pattern	1
2.1	Locations of 62 seedlings and saplings of California redwood trees[1]	4
2.2	The dataset chorley is a 2D point pattern representing marked locations of home addresses of cancer cases. [2]	5
2.3	Random generated Poisson point processes with different intensities.	8
2.4	Locations of nesting sites of gorillas with inhomogeneous intensity, in a National Park in Cameroon [3].	8
2.5	Simulated three different types of point patterns	10
2.6	Fry plot for the three point patterns	11
2.7	Simulated point pattern with edge effects	12
2.8	Empirical K-functions for the simulated patterns	14
2.9	Pair correlation functions for the simulated patterns	15
2.10	Nearest neighbour distance distribution functions for the simulated three different patterns	16
3.1	Simulated locations of mineral deposits	17
3.2	Histogram of nearest neighbour distances	19
3.3	feature detection of gem clusters	20
4.1	Local K-functions of redwood data[1]	23
4.2	HCA of local K-functions of redwood data[1]	23
4.3	Two different clusters of local K-functions of redwood data	24
4.4	2D simulated cluster point pattern	28
4.5	Product density LISA functions without edge correction	29

4.6	Edge corrected product density LISA functions	29
4.7	Simulated 3D point process	32
4.8	Product density LISA function with no edge effects	32
4.9	Edge corrected 3D product density LISA functions	33
4.10	Traditional representation of Divisive and Agglomerative methods	34
4.11	HCA of mineral deposits data	36
4.12	HCA of mineral deposits data with 3 clusters	36
4.13	HCA of mineral deposits data, cutting at height 6.	37
4.14	HCA of mineral deposits data with 11 clusters	37
4.15	Dendrogram of hierarchical cluster analysis of 2D simulated point pattern	39
4.16	Dendrogram of hierarchical cluster analysis of 3D simulated point pattern	39
4.17	The cluster points detected by HCA of 2D simulated point pattern	40
4.18	The cluster points detected by HCA of 3D simulated point pattern	40
4.19	Dendrogram of the matrix D of 2D simulated point pattern	46
4.20	Dendrogram of the matrix D of 3D simulated point pattern	47
4.21	Feature detection from nnClean in 2D simulated point pattern	51
4.22	Feature detection from nnClean in 3D simulated point pattern	52
4.23	Histogram of L1 distances of the 2D simulated point pattern	54
4.24	2D simulated feature and clutter points for s-step in SEM	57
4.25	Convergence of parameters of SEM algorithm in 2D case- for $\tau = 0.2$ and L_1 distances	58
4.26	Convergence of parameters of SEM algorithm in 2D case- for $\tau = 0.6$ and L_1 distances	58
4.27	Convergence of parameters of SEM algorithm in 2D case- for $\tau = 0.2$ and L_2 distances	59
4.28	Convergence of parameters of SEM algorithm in 2D case- for $\tau = 0.6$ and L_2 distances	59
4.29	Feature detection from 2D SEM	60
4.30	Feature detection from 2D SEM	61

4.31 Convergence of parameters of SEM algorithm in 3D case- for $\tau = 0.3$ and L_1 distances	62
4.32 Convergence of parameters of SEM algorithm in 3D case- for $\tau = 0.7$ and L_1 distances	62
4.33 Convergence of parameters of SEM algorithm in 3D case- for $\tau = 0.3$ and L_2 distances	63
4.34 Convergence of parameters of SEM algorithm in 3D case- for $\tau = 0.7$ and L_2 distances	63
4.35 Estimated feature points of 3D simulated clustered process from SEM method	64
5.1 Simulated 2D feature-clutter process X_2	67
5.2 The dendrogram of X_2	67
5.3 Estimated feature and clutter of X_2 using HCA without covariance	68
5.4 Estimated feature and clutter of X_2 using EM method with $k = 1$	68
5.5 Estimated feature and clutter of X_2 using EM method with $k = 2$	69
5.6 Estimated feature and clutter of X_2 using EM method with $k = 5$	69
5.7 Simulated point process for the s-step	70
5.8 SEM method for L_1	70
5.9 SEM method for L_2	71
5.10 Simulated 3D point process X_3	71
5.11 The dendrogram of HCA of X_3	72
5.12 HCA of X_3 with $k = 2$	73
5.13 HCA of X_3 with $k = 3$	73
5.14 EM method of X_3 with $k = 1$	74
5.15 EM method of X_3 with $k = 2$	74
5.16 EM method of X_3 with $k = 5$	75
5.17 Simulated feature and clutter for the s-step in the SEM method of X_3	75
5.18 Estimated feature and clutter of X_3 from SEM method with L_1	76

List of Tables

3.1	Numerical values of simulated mineral deposits	18
3.2	Estimated parameter values of k=1	20
3.3	Estimated parameter values of k=5	20
3.4	Numerical analysis of simulated locations of mineral deposits	21
4.1	Numerical analysis of HCA without covariance	40
4.2	Summary of nnClean function in 2D point pattern	52
4.3	Summary of nnClean function in 3D point pattern	52
4.4	Numerical analysis of 2D SEM	60
4.5	Numerical analysis of 3D SEM	64
5.1	Summarized results of 2D clustering	65
5.2	Summarized results of 3D clustering	65
5.3	Summarized results of clustering of X_2	76
5.4	Summarized results of clustering of X_3	77

Chapter 1

Introduction

This study is based on the analysis of spatial point pattern data. If we consider the spatial arrangement of point patterns then we could clearly see how the points are located inside the observation window and it gives the different patterns to the data such as independent (complete spatial random), regular (points tend to avoid each other) or clustered (some points tend to be closed together). Here we discuss about the spatial point patterns with clusters and methods to identify the clustered data from the patterns. In this situation we can divide our pattern into two different densities one with low intensity and other one with high intensity. Through this study we define cluster data as features and the other points of the observation area as clutter.

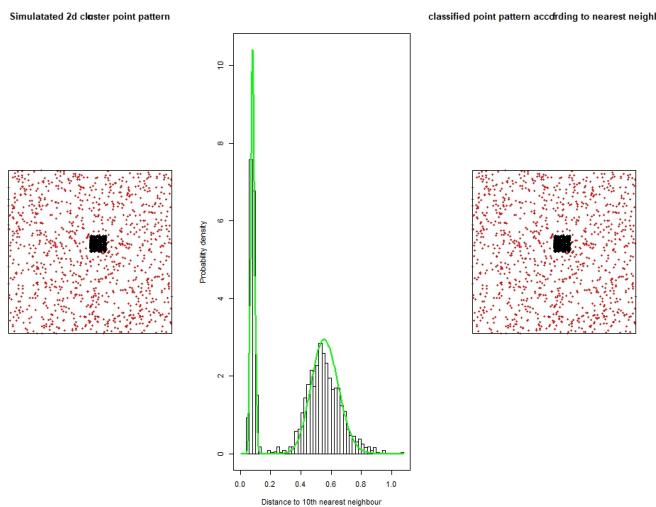


Figure 1.1: Nearest neighbour cleaning for simulated point pattern

The Figure 1.1 shows an artificial example of a simulated cluster point pattern using spatstat

package in R for quick visualization of intensity regions of a cluster point pattern.

The left image of the Figure 1.1 shows the simulated cluster point pattern. The middle image is the histogram of the tenth nearest neighbour distances and the green line over the histogram is the fitted density of the model that shows two different intensity peaks. These two different peaks of the histogram tell that the points of process come from two different point processes (one is feature process and other one is clutter) with different intensities. To separate the points into feature/cluster and clutter we can use a clustering method. The right image is the classified pattern of features and clutter data using **nnclean** function of spatstat. The red color points are clutter data and black points are features in both cases. If we compare the left and right images visually, we could see that the feature points identified from **nnclean** are mostly same as the simulated cluster.

In the literature of spatial point process analysis, there are some assumptions that have been made for cluster point process (page 460[10]). To make progress in our study we assume that the clutter is distributed as a Poisson process and the different clusters are independent from each other.

In our study we use the local indicators of spatial association (LISA) functions to detect the features. The K -function and the product density function are used as the measures of point process to construct the model of LISA functions. These LISA functions are applied in hierarchical cluster method, EM algorithm and SEM to detect the high intensity regions of the point patterns. The methods and ideas of LISA functions that used for 2D point process analysis are implemented for 3D point process in this thesis study as a new task.

The basic definitions and background knowledge related to the study are provided in the second chapter. The third chapter contains exploration of cluster point patterns and available methods in **spatstat** for cluster detection. The methodology of building product density LISA functions is widely described in the chapter four with the statistical theories. Also clustering of product density LISA functions using hierarchical cluster method, EM algorithm and SEM algorithm are included in the chapter four. The comparison study of the results from the different cluster methods for the simulated 2D and 3D point processes is carried out in the fifth chapter. Finally we discuss about the future works for further study and give conclusions that are made from the results of the study.

Chapter 2

Basics of Spatial Point Processes

This chapter describes the basic mathematical theories and statistics used in spatial point patterns and practical techniques to handle them using the **spatstat** data package in **R**[10]. The simulated point patterns and available point patterns in **spatstat** package are used for the given examples. These examples will help to understand the concepts of mathematical theorems and definitions.

2.1 Point Patterns

A dataset giving the observed spatial locations of things or events in d-dimensional space such as location of trees in a forest, gold deposits mapped in a geological survey, stars in a star cluster, road accidents, earthquake epicenters, mobile phone calls or cases of a rare disease is called as a spatial point pattern. A spatial point process is a point process which models data that is localized at a discrete set of locations in space or, more specifically, on a plane(see 2.1.1). This kind of data sets are popular in many research studies in different fields of science, mainly in Geo-science, astronomy, ecology and econometrics. We can plot the spatial locations of a point pattern to see its arrangement graphically. The locations of 62 redwood tress has been plotted in the Figure 2.1.

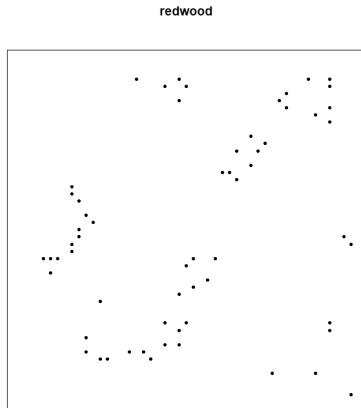


Figure 2.1: Locations of 62 seedlings and saplings of California redwood trees[1]

The main task of analysis of point patterns is identifying the important features of individual points or point sets that can be used to describe the behavior of the points and make the predictions.

The point patterns can be simply described as homogeneous (points are placed uniformly in observation window), non homogeneous (points are not uniformly placed) or clustered patterns according to spaces between the points inside the observation window but this is not enough to answer the research questions like intensity of a process, mapping of distances between points, classifying the points into clusters, hypothesis testing and correlations of points etc. Statistically accepted methods are needed to attain better inferences in the case of research works of point pattern data. Also there are different types of points can be included in a point pattern. As an example, if we consider the locations of trees then there are different kinds of trees in a forest and these locations may be classified due to their species. If we consider a cluster from this kind of point process then the cluster may contain different type of points. Such a point pattern is shown in the Figure 2.2. The point pattern in the Figure 2.2 describes the locations of home addresses of two types of cancer cases. If we look at the points in a cluster then we can see it as a mixture of the two types of points.

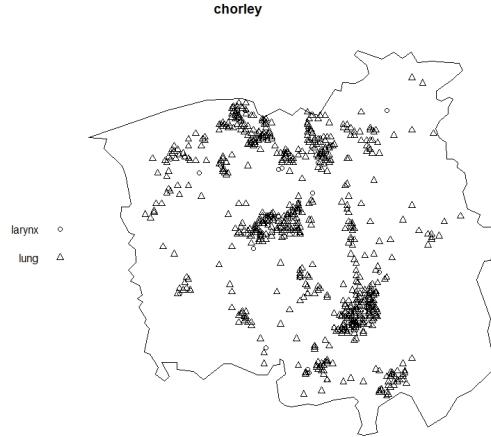


Figure 2.2: The dataset chorley is a 2D point pattern representing marked locations of home addresses of cancer cases. [2]

A point pattern data set brings its own details together with data which are number of data points, coordinates, intensity, area of window, unit of area measurement and mark values etc. These details are very basic but important and describe the pattern. The **summary** function of spatstat package can be simply used to get these basic summary statistics of a point pattern.

2.1.1 Definition

The mathematical definition for a spatial point process can be expressed as follows.

Let,

S : Polish state space of the point process equipped with the σ -algebra of Borel sets \mathcal{B}

A set of points $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ and $A \subseteq S : \mathbf{X}_A = \mathbf{X} \cap A$

N_{lf} : the space of locally finite arrangement of points, i.e.

$\{\mathbf{X}, n(\mathbf{X}_A)\} = |\mathbf{X}_A| < \infty, \forall A \text{ bounded} \subseteq S$

Then a point process \mathbf{X} defined on S is a measurable application defined on some probability space [9] (Ω, \mathcal{F}, P) with values on N_{lf} .

If we select a region in d -dimensional space which is subset of A and get the points included in this region then the selected region is called as the observation window. In our study, we assume that $S = \mathbb{R}^d$ and give a simple definition for a point process,

A point pattern is a collection of random locations of points (a realisation of a point process) in d-dimensional space. A point pattern \mathbf{X} with number of \mathbf{n} finite locations contained in observation region \mathbf{W} is defined as,

$$\mathbf{X} = \{x_1, x_2, \dots, x_n\} \in \mathbf{W}.$$

Locations of individual points can be arranged in a vector format. If we consider x_i , the location of point i,

$$x_i = (x_i) \text{ in 1-d}$$

$$x_i = (x_i, y_i) \text{ in 2-d}$$

$$x_i = (x_i, y_i, z_i) \text{ in 3-d}$$

etc. If every possible outcome of a point process is a point pattern with a finite number of points and number of points falling in any region \mathbf{A} is a well defined random variable then the process is called a finite point process.

Homogeneity: The points have no preference for any spatial location which means that the expected number of points in a region is proportional to its area on average.

$$\mathbb{E}[n(\mathbf{X} \cap \mathbf{W})] = \lambda |\mathbf{W}| \quad (2.1)$$

where λ is the intensity of the point process \mathbf{X} and $|\mathbf{W}|$ is the area of the region.

Independence: The points in one region of the space have no influence from points in other region.

If we take the different regions from the point process region \mathbf{A} as $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_m$ which do not overlap then the number of points falling to these subregions $n(\mathbf{X} \cap \mathbf{A}_1), n(\mathbf{X} \cap \mathbf{A}_2), \dots, n(\mathbf{X} \cap \mathbf{A}_m)$ are independent random variables.

The Poisson point process

A Poisson point process \mathbf{X} is a point process which has the property of independence and the number $n(\mathbf{X} \cap \mathbf{A})$ of random points falling in the test region \mathbf{A} has a Poisson distribution

$$P(n(X_A) = k) = e^{-\lambda} \frac{\lambda^k}{k!} \quad (2.2)$$

where λ is the intensity of the point process.

A point process followed the properties of homogeneity and independence is called **homogeneous**.

geneous Poisson process or complete spatial random (CSR) process.

2.2 Methods of Statistics

A simple way to interpret a point pattern is use of summary statistics. Randomness (or independence) and homogeneity of a point pattern can be described using basic numerical values and further we can use summary functions like pair correlation functions instead of numerical values. But still there is a need of statistical methods to analyze data in complex situations to answer the scientific questions.

There are some assumptions that are made for statistical analysis. Some statistical tests can be used only if the points satisfy the required assumptions for the tests. Most common standard assumptions for the analysis are:

1. point locations are measured exactly
2. no two points lie at exactly the same location
3. no errors in detecting the presence of points of the random process within the region \mathbf{W}
4. points could have been observed at any location in the region \mathbf{W}

Other than above assumptions, researchers make another assumptions about the random point process itself to move forward in the analysis. A point process is called

Stationary if, the statistical properties (mean, variance, covariance) of the process are the same in each position even when the process is shifted without changing its directional orientation. If we shifted the point process \mathbf{X} with vector \mathbf{u} , then the statistical properties of the resulting point process $\mathbf{X} + \mathbf{u}$ are same as of the point process \mathbf{X} for any choice of \mathbf{u} .

Isotropic if its statistical properties are unchanged when it is rotated around the origin through any given angle.

We have to check whether the point pattern follows the assumptions before using the statistical tests for getting significant answers. For example, the K -function is well used when the process is stationary, if not we need an estimate function of the intensity.

2.2.1 Intensity

The intensity is a measure for density of a point process that means expected number of points per unit area. The value of intensity depends on the unit of measurement. The Figure 2.3 shows three different Poisson point processes with intensities 10, 100 and 1000 from left to right respectively generated in unit squares.

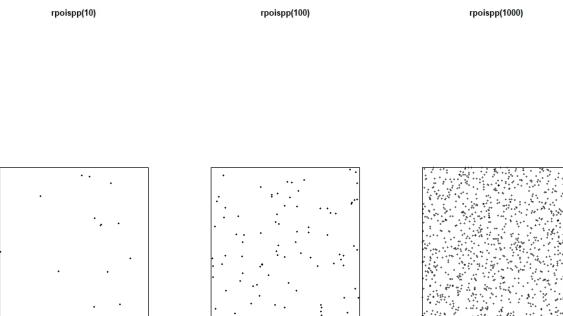


Figure 2.3: Random generated Poisson point processes with different intensities.

The most interest research question is whether the intensity is homogeneous or not. The intensity is given as a function of locations when the process is inhomogeneous. An inhomogeneous point process is shown in the Figure 2.4.

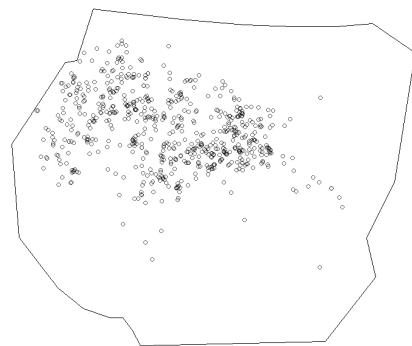


Figure 2.4: Locations of nesting sites of gorillas with inhomogeneous intensity, in a National Park in Cameroon [3].

It is proven that the sample mean of a data set is an unbiased estimator for the population mean. In similar way we can estimate the intensity of a point process by the empirical density of points. Let's consider a point process \mathbf{X} and a point pattern data \mathbf{x} (a realisation of \mathbf{X}) with $n = n(\mathbf{x})$ points obtained in region \mathbf{W} . If we assume that the process has homogeneous intensity λ then its unbiased estimate is given by

$$\hat{\lambda} = \frac{n(\mathbf{x})}{|W|} \quad (2.3)$$

If the point pattern is not homogeneous then the intensity is defined as a function of location. At a spatial location x , the intensity is $\lambda(x)$. For any region \mathbf{W} , the integral of the intensity function gives the expected number of points. i.e

$$E[n(\mathbf{X} \cap W)] = \int_W \lambda(x) dx \quad (2.4)$$

2.2.2 Variance Covariance

The variance of the count $n(\mathbf{X} \cap W)$ is defined as

$$\text{var}[n(\mathbf{X} \cap W)] = E[n(\mathbf{X} \cap W)^2] - \{E[n(\mathbf{X} \cap W)]\}^2 \quad (2.5)$$

if we consider the covariance of two such counts,

$$\text{cov}[n(\mathbf{X} \cap W_1), n(\mathbf{X} \cap W_2)] = E[n(\mathbf{X} \cap W_1)n(\mathbf{X} \cap W_2)] - E[n(\mathbf{X} \cap W_1)]E[n(\mathbf{X} \cap W_2)] \quad (2.6)$$

$n(\mathbf{X} \cap W_1)n(\mathbf{X} \cap W_2)$ is equal to the number of ordered pairs (x, x') of points in the point process \mathbf{X} such that $x \in W_1$ and $x' \in W_2$.

The mean measure of \mathbf{X} given by the equation (2.4) is denoted as $\nu_1(W)$. The second order moment measure is

$$\nu_2(W_1, W_2) = \int_{W_1} \int_{W_2} \lambda_2(x_1, x_2) dx_1 dx_2 \quad (2.7)$$

Then the covariance structure of \mathbf{X} is

$$\begin{aligned}
 Cov[n(\mathbf{X}_{W_1}), n(\mathbf{X}_{W_2})] &= \nu_2(W_1, W_2) + \nu_1(W_1 \cap W_2) - \nu_1(W_1)\nu_1(W_2) \\
 &= \int_{W_1} \int_{W_2} [\lambda_2(x_1, x_2) - \lambda(x_1)\lambda(x_2)] dx_1 dx_2 + \int_{W_1 \cap W_2} \lambda(x) dx
 \end{aligned} \tag{2.8}$$

2.2.3 Correlation

The statistical measure of correlation i.e covariance determines the dependence between points. Basically three types of patterns which are regular, independent and clustered defined according to the points placing in the area. The simulated regular, independent and cluster processes are plotted in the Figure 2.5.

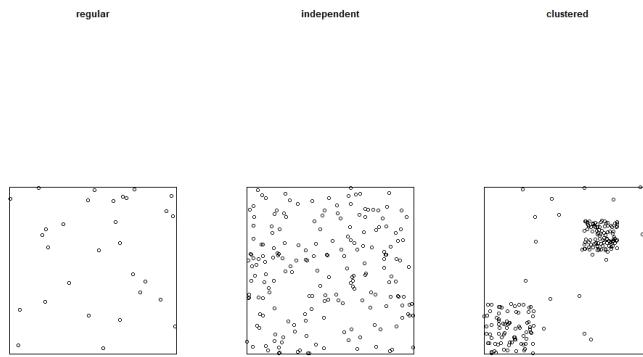


Figure 2.5: Simulated three different types of point patterns

It is defined that the regular pattern has negative covariance, the random pattern has zero covariance and the clustered pattern has positive covariance. There are simple manual techniques, Morisita Index, Fry plot that we can use to calculate covariances in point patterns.

2.2.3.a Fry plot

We can use the information of the spaces between points to say about correlations. The idea of Fry plot based on the crystallography developed by Patterson[11]. Mathematically, a Fry plot

is a scatter plot of the vector differences between all pairs of distinct points. The details way of drawing Fry plot can be studied from [201-203][10].

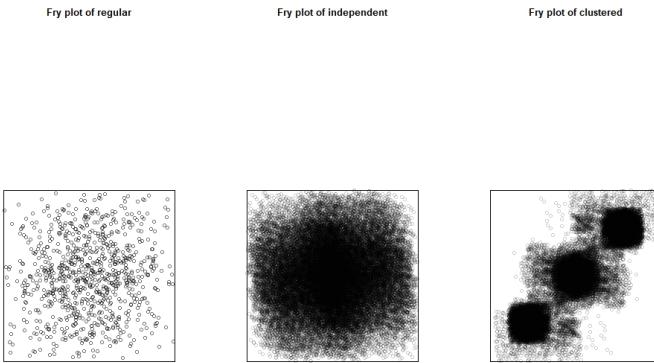


Figure 2.6: Fry plot for the three point patterns

The Figure 2.6 shows Fry plots of the simulated three different point processes shown in the Figure 2.5.

2.2.4 Edge effects

Let's consider a point pattern observed in a window \mathbf{W} from a point process. If we want to count the number of r -nearest neighbour points for any point x_i then we can draw a circle at x_i with radius r and count the points belong inside the circle. But if we consider a point x_i near to the border of the window then this circle may goes outside the window. In this case if we do not know the points which lie out side the observation window then the information is lost to count the points which belongs to the circle. This type of problem is called edge effects in point pattern statistics.

The Figure 2.7 shows some circles of selected points with given distance r which lie outside the observation window \mathbf{W} of a simulated point pattern.

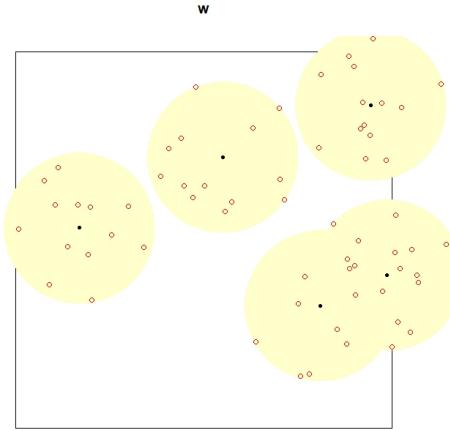


Figure 2.7: Simulated point pattern with edge effects

2.2.5 K - function

It is clear that many pairwise distances of a cluster process are small since points are located close to each other and a regular pattern, these pairwise distances become large values. The information of distances or spaces between points that well gathered and measured in statistical way is defined by K- function to analysis spatial correlation in point patterns.

If we consider point process \mathbf{X} with n number of points and all the pairwise distances $d_{ij} = \|x_i - x_j\|$ where x_i and x_j are distinct points in the pattern then the fraction of pairs for which the distance d_{ij} is less than or equal to r , $\widehat{H}(r)$ (the empirical cumulative distribution function of d_{ij} s) is given by following equation.

$$\widehat{H}(r) = \frac{1}{n(n-1)} \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \mathbf{1}\{d_{ij} \leq r\} \quad (2.9)$$

The indicator $\mathbf{1}$ equals to 1 if d_{ij} is less than r and otherwise zero. The number of r -neighbours $t_i(r)$ for the point x_i is give by

$$t_i(r) = \sum_{j \neq i} \mathbf{1}\{d_{ij} \leq r\} \quad (2.10)$$

In another way, if we draw a circle at point x_i with radius r then the number of data points

which fall inside the circle (without x_i) is given by $t_i(r)$. If we take the sum and divide by the total number of points then we can get the average number of r-neighbours per data point. i.e,

$$t(r) = \frac{1}{n} \sum_i t_i(r) \quad (2.11)$$

Then we can redefine

$$\widehat{H}(r) = \frac{1}{(n-1)} t(r) \quad (2.12)$$

The maximum number of point for any data point is $(n-1)$ and we can approximate the intensity dividing $(n-1)$ by area of the point pattern and the function $\widehat{H}(r)$ can be standardised using this intensity approximation. Then the function $|\mathbf{W}|\widehat{H}(r)$ gives the standardised average number of r-neighbours of a data point where $|\mathbf{W}|$ is the area of the observation window \mathbf{W} .

Generally point patterns are observed in different windows. Calculating the function $|\mathbf{W}|\widehat{H}(r)$ for any given windows leads to modification of the function with edge effects.

The cumulative average number of data points within distance r of a typical data point with corrected edge effects and standardized by the intensity is called the **empirical K- function** where e_{ij} is an edge correction weight. The empirical K-functions of the simulated point processes (Figure 2.5) are shown in the Figure 2.8.

$$\widehat{K}(r) = \frac{|\mathbf{W}|}{n(n-1)} \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \mathbf{1}\{d_{ij} \leq r\} e_{ij}(r) \quad (2.13)$$

2.2.6 Pair correlation function

We know that the K- function collects the information of all interpoint distances less than or equal to r . But if we consider contributions only from interpoint distances equal to r then it is called as the pair correlation function $g(r)$.

$$g(r) = \frac{K'(r)}{2\pi r} \quad (2.14)$$

where $K'(r)$ is the first derivative of the K- function w.r.t r .

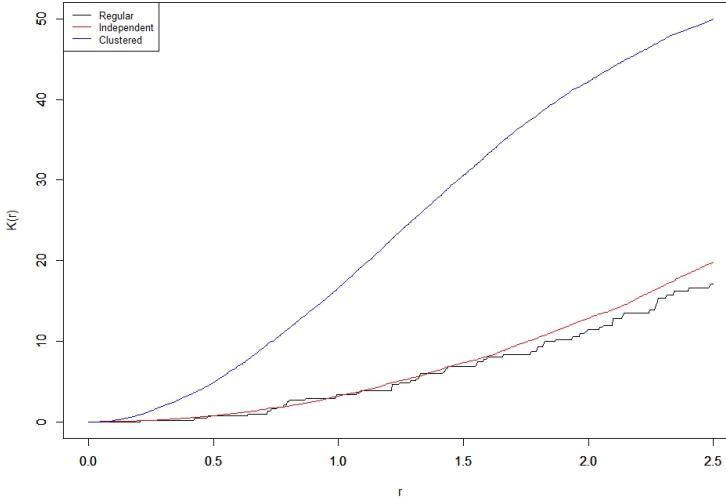


Figure 2.8: Empirical K-functions for the simulated patterns

The $g(r)=1$ indicates the complete spatial randomness and $g(r) < 1$ shows the process is regular and $g(r) > 1$ suggests that the process is clustered. The Figure 2.9 shows how a point process changes from CSR according to its pair correlation function for the three different point processes given in the Figure 2.5.

2.2.7 Product density function

Consider a point process \mathbf{X} with n number of points observed in the region $\mathbf{W} \subset \mathbb{R}^2$ of area a . The empirical K-function given by the equation (2.13) is a summary of the pairwise distances and if we assume that \mathbf{X} is a stationary point pattern with the intensity λ then for any $r \geq 0$, the product density function $\rho(r)$ is obtained from the differential of K-function,

$$\rho(r) = \frac{\lambda^2 K'(r)}{2\pi r}, r > 0 \quad (2.15)$$

We can re-arrange the equation (2.15) by considering the pair correlation function given by the equation (2.14) as

$$\rho(r) = \lambda^2 g(r), r > 0 \quad (2.16)$$

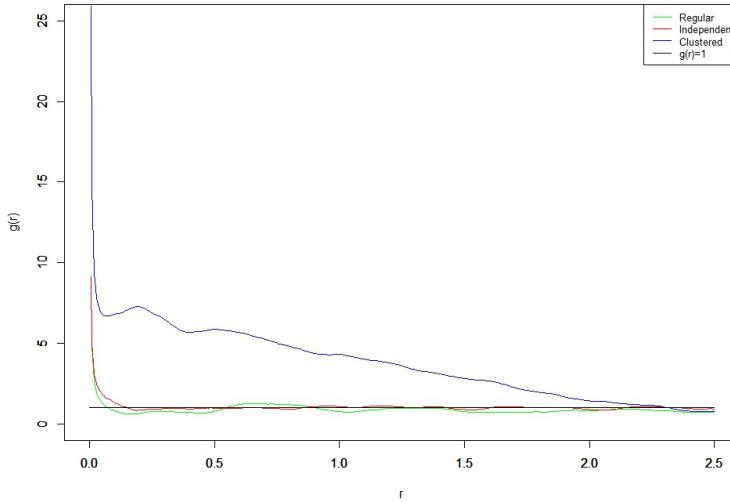


Figure 2.9: Pair correlation functions for the simulated patterns

The empirical product density function that estimated from data of a point process describes the density of interevent distances and it helps to identify the clusters which have high values of $\hat{\rho}(r)$ for small r .

2.2.8 Nearest neighbour function G

Nearest neighbour distance

If we consider any point x_i of the point pattern \mathbf{X} then the nearest neighbour distance (d_i) of the point x_i is the minimum distance from each points in the pattern to x_i , i.e $d_i = \min_{j \neq i} \|x_j - x_i\|$. Also we can rewrite this as follows,

$$d_i = d(x_i, \mathbf{X} \setminus x_i) \quad (2.17)$$

where d_i is the shortest distance from x_i to the point process of all the points without the point x_i which is denoted by $\mathbf{X} \setminus x_i$.

The nearest neighbour distance distribution function $G(r)$ is the cumulative distribution function of the nearest neighbour distance at a point x of the point process \mathbf{X} . For the different type of point processes (Figure 2.5), the nearest neighbour distance functions are plotted in the

Figure 2.10.

$$G(r) = \mathbf{P}\{d(x, \mathbf{X} \setminus x) \leq r \mid \mathbf{X} \text{ has a point at } x\} \quad (2.18)$$

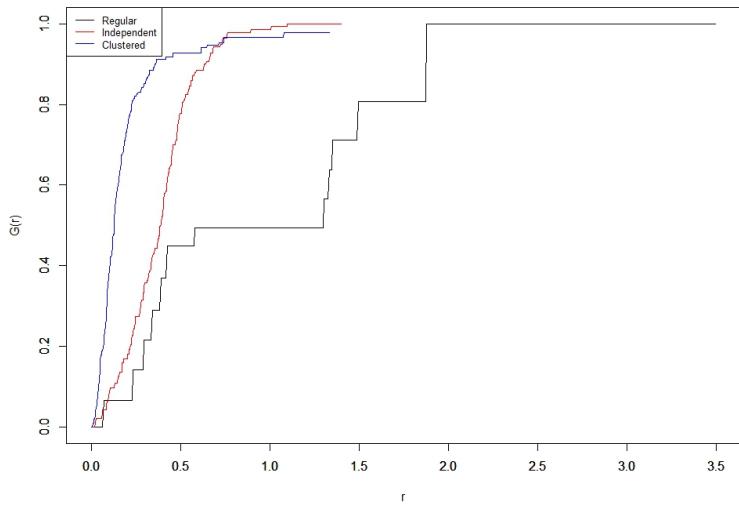


Figure 2.10: Nearest neighbour distance distribution functions for the simulated three different patterns

Chapter 3

Cluster Point Process

In the second chapter we discussed about the different kind of point processes. But in this thesis our main focus is studying about the clustered point processes. A cluster is group of points that are located close to each other. A clustered point process has data points that lying in the observation window but do not belong to any group of points. The points of groups that are cluster points are named as features and the other points of the process are called clutter throughout the our study. Detecting the cluster points, finding intensity of each cluster and extracting individual clusters are some common problems in cluster point pattern analysis.

Example Let's assume that we have 125 locations of mineral deposits in a $10m \times 10m$ square region of a mining land and we are interested about the distribution of gem deposits in this area. A simulated locations regarding this problem are shown by following Figure 3.1 .

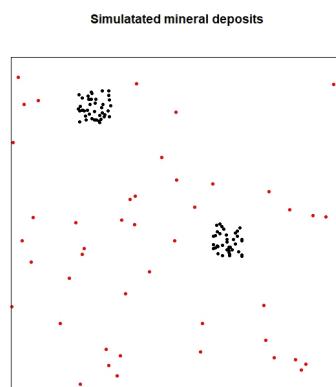


Figure 3.1: Simulated locations of mineral deposits

The numerical values of the cluster intensities, number of points in each cluster and their volumes are presented in the Table 3.1.

Table 3.1: Numerical values of simulated mineral deposits

	mineral deposits	gem deposits cluster 1	gem deposits cluster 2
number of locations	44	42	39
intensity	0.5	35	35
volume	10×10 units	1×1 units	1×1 units

The locations of gem deposits are indicated by black colour points and the locations of the other minerals are shown by red colour. In this point pattern we can see two clusters of gem deposits and our problem is dividing the points into two sets that one is gem (features) and other one is minerals (clutter) and extracting these two gem deposit clusters from the pattern.

Before we go to building our method for solving this kind of cluster process problems, we check out the available functions built in **R** that can be used to solve the problem.

There are functions in **R** for clustering. We will first use these functions with our simulated data set to get motivation towards finding a new method for our task.

3.1 nnclean

The function `nnclean` [7] in **R** is based on EM algorithm (4.3.2) and NN distance distributions. First compute the distance to the kth nearest neighbour for each point in the data set. Then the EM algorithm is used to fit a mixture distribution to the kth nearest neighbour distances which has a form of a mixture of two transformed Gamma distributions. The mixture components represent the feature(cluster) and the clutter. The mixture model can be used to classify each point as belong to one or other component. `nnclean` estimates the numerical values for the parameter p which is the probability that each point belongs to a cluster and the intensity values for feature and clutter distributions.

The assumptions[7] made in method of nnclean are as follows.

- The clutter is distributed as a homogeneous Poisson process on a certain region.
- The clusters are distributed as a homogeneous Poisson process with larger intensity on a subregion (disconnected in case of more than one cluster).

We applied the nnclean to our simulated data set for $k=1$ and $k=5$. The histograms of the mixture distribution of the nearest neighbour distances and their probability density functions for both $k=1$ and $k=5$ are shown in the Figure 3.2. The Figure 3.3 shows the identified clusters as feature points for the both cases.

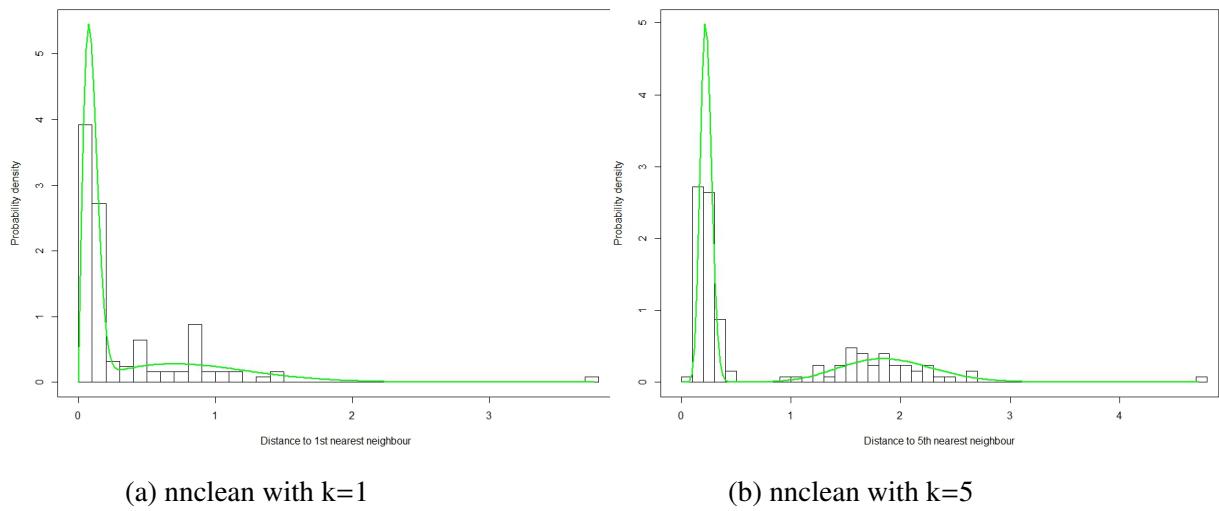


Figure 3.2: Histogram of nearest neighbour distances

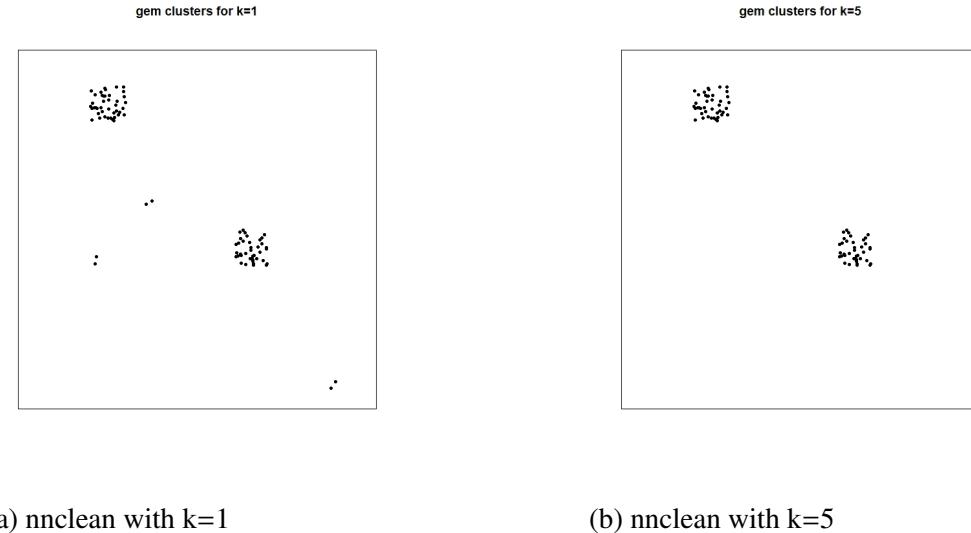


Figure 3.3: feature detection of gem clusters

When we run the function nnclen for the first nearest neighbour distances then it has given small clusters of less points other than the main two clusters of gems. These small cluster points are coming from clutter but they are wrongly detected as features. Clearly we can see the function with $k=5$ detected the two gem deposit clusters better than the case of $k=1$. The Table 3.2 and the Table 3.3 show the numerical values of the two different cases.

Table 3.2: Estimated parameter values of $k=1$

	clutter	features
p	-	0.67758
intensity	0.32371	27.496
number of locations	38	87

Table 3.3: Estimated parameter values of $k=5$

	clutter	features
p	-	0.64786
intensity	0.42364	29.536
number of locations	44	81

In any cluster process, some points from the clutter can be identified as features and vice

versa. To find out which points are falsely identified, we can use the labels of points that given by **nnclean** function and compare them with the true labels. The Table 3.4 shows the mapping of two different cases and results shows how many points are correctly fallen to the clusters, that means number of true cluster points that are identified as cluster points by nnclean.

Table 3.4: Numerical analysis of simulated locations of mineral deposits

	Exact deposits		Estimated number of locations				Correct identification rate	
	Minerals (clutter)	Gems (features)	clutter		features		clutter	features
			$c \mapsto c$	$f \mapsto c$	$f \mapsto f$	$c \mapsto f$		
k=1	44	81	38	0	81	7	86.36%	100%
k=5	44	81	44	0	81	0	100%	100%

It shows that **nnclean** separates locations into clutter and feature with high accuracy rate but still we need a method to extract the individual clusters and their parameters.

In chapter two we discussed about the basic statistics for the spatial point process analysis and in this chapter we learn about method of clustering. By gathering all these information we discussed in chapter two and three, our aim is building a method for spatial point process clustering for 2D and 3D point process. In next chapter we will discuss about the details of our method of clustering.

Chapter 4

Methodology

4.1 Introduction

In this chapter, we discuss about how to use local indicators of spatial association for a point process to identify features. We discussed summary statistic functions in chapter two those give global values of a point pattern. But in our study we are interested about local versions of summary functions.

4.1.1 Local Indicators of Spatial Association

Local indicators are calculated by decomposing summary statistics into contribution from each of the data points in a point pattern and used for handling inhomogeneous point patterns. As an example if we consider the K-function that we discussed in the section 2.2.5 and if we break the empirical K-function given by the equation 2.13 to calculate the values at each points then we can get local K-functions of a point pattern. We define the local K-function at point x_i as $\widehat{K}(r, x_i)$. To identifying the features of a point pattern we can classify LISA functions into several groups using suitable cluster technique.

Example: Let's consider simple example using the **spatstat** functions to understand the idea of using LISA function for clustering. First we get the local K functions of the "redwood" dataset (see Figure 2.1) in **R** which describes the locations of 62 seedlings and saplings of California redwood trees using **localK** function. The Figure 4.1 shows the plotted local K-

functions of the "redwood" data set.

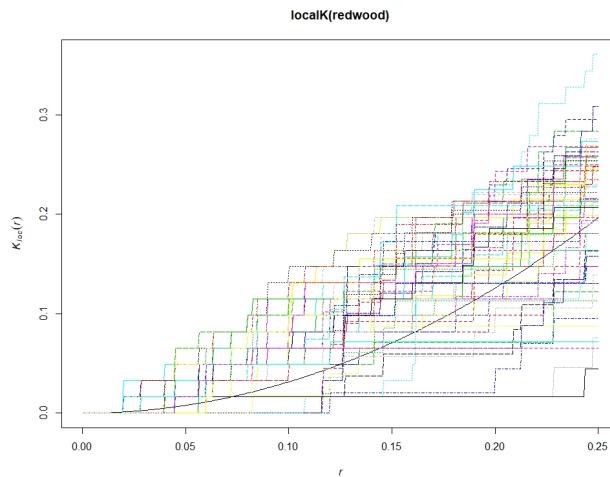


Figure 4.1: Local K-functions of redwood data[1]

We use **R** function **hclust** to applying hierarchical cluster method and Figure 4.2 shows the dendrogram of the local K-functions.

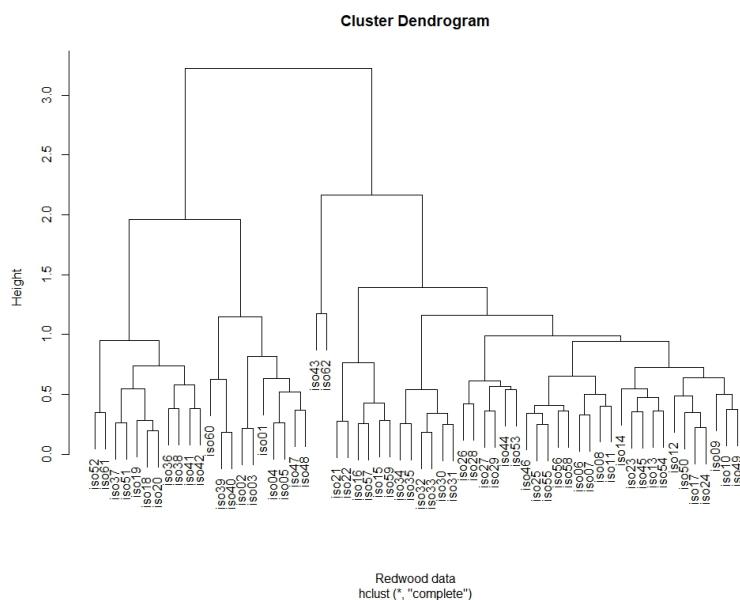


Figure 4.2: HCA of local K-functions of redwood data[1]

If we choose any height along the y-axis of the dendrogram, and move across the dendrogram counting the number of lines that we cross, each line represents a group that was identified when objects were joined together into clusters. The observations in that group are represented by the branches of the dendrogram that spread out below the line. For example, if we look at a height of 2.5, and move across the x-axis at that height, we'll cross two lines. That defines a two-cluster solution; by following the line down through all its branches, we can see the names of the redwood trees that are included in these two clusters.

Looking at the dendrogram, there are clearly two very distinct groups; the left hand group seems to consists of two more distinct clusters, while most of the observations in the right hand group are clustering together at about the same height but if we count the very close point then the right hand group consists three clusters and one of them is clearly shown that it has only two members which are iso43 and iso62. It looks like either two or five groups might be an interesting place to start investigating.

If we cut the dendrogram for two clusters then the result is shown in Figure 4.3.

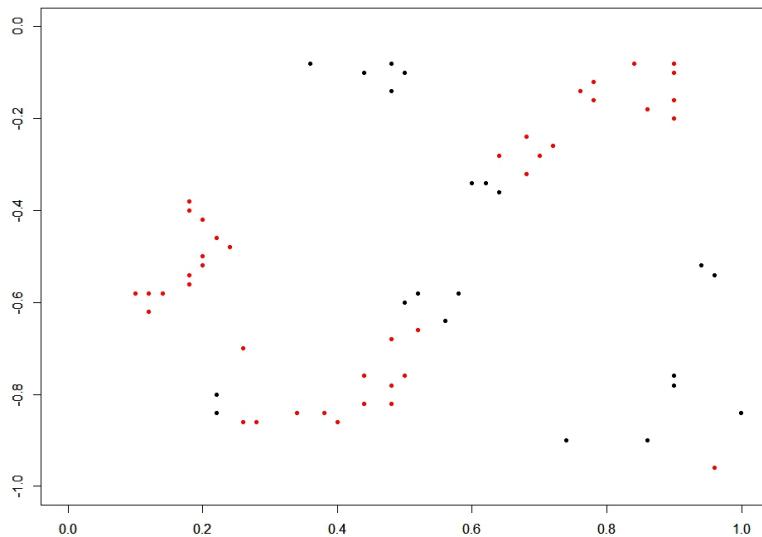


Figure 4.3: Two different clusters of local K-functions of redwood data

4.2 Product Density LISA Functions

Product density function is a second order characteristic of spatial point processes and we can use product densities of each points to see their relationships with neighboring points. Our goal is detecting the features of a cluster point pattern and similar studies were done by some scientists using product density LISA functions. The ideas of the methods developed by J.Mateu et al. 2007 [13] and N.Cressie and L.B.Collins 2001 [14] are followed in our study and extended for 3D point patterns.

4.2.1 Two Dimensional Point Processes

Consider a point process \mathbf{X} with n number of points observed in the region $\mathbf{W} \subset \mathbb{R}^2$ of area a . The global product density function is given by the equation (2.15). To estimate the product density LISA function we follow the study done by N.Cressie and L.B.Collins(2001)[14]. They described the mathematical background to build product density LISA functions very descriptively in their study.

Estimating of product density function follows kernel estimation of $\lambda^2 K'(r)$. It takes the form without considering the edge effects,

$$\widehat{\lambda^2 K'(r)} = a^{-1} \sum_{i=1}^n \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r), \quad r > \epsilon > 0 \quad (4.1)$$

where $\|x_i - x_j\|$ is the Euclidean distance between x_i and x_j , f_ϵ is a kernel function and ϵ is a bandwidth. The estimations are only valid till the one quarter of the window edge length. Then we define r as $r=0.25$ (value of minimum edge length).

In our study, we use the Epanechnikov kernel function

$$f_\epsilon(s) = \begin{cases} \frac{3}{4\epsilon} \left(1 - \frac{s^2}{\epsilon^2}\right) & \text{for } -\epsilon \leq s \leq \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

and bandwidth $\epsilon = (5^{1/2}/10)\hat{\lambda}^{-1/2}$.

By substituting equation (4.1) and equation (4.2) to the equation (2.15), we can get the global kernel density estimate of the product density function,

$$\hat{\rho}_\epsilon(r) = \frac{1}{2\pi r a} \sum_{i=1}^n \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r), \quad r > \epsilon > 0 \quad (4.3)$$

The local product density function of a point x_i says its contribution to the above global product density function. We start from the local K-function of a point x_i to derive the local product density function of x_i . Let's consider the expected number of points with distance from x_i less than or equal to r of the point process \mathbf{X} without counting the point x_i .

$$\{\lambda K(r)\}^i = E[\mathbf{X}(b(x_i, r) \setminus \{x_i\}) | x_i \in \mathbf{X}], \quad r > 0 \quad (4.4)$$

The kernel density estimate for the $\{\lambda K'(r)\}^i$ can be determined by the equation (4.1),

$$\widehat{\{\lambda K'(r)\}}^i = \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r), \quad r > \epsilon > 0 \quad (4.5)$$

From the properties of a homogeneous Poisson process we can get $(n-1)/a$ as the unbiased estimator for the intensity of reduced point process $\mathbf{X}_{x_i}^!$ and multiply it by the equation (4.5) to obtain the kernel density estimate for the derivative $\{\lambda^2 K'(r)\}^i$ for $r > 0$.

$$\hat{\lambda} \{\widehat{\lambda K'(r)}\}^i = (n-1)a^{-1} \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r), \quad r > \epsilon > 0 \quad (4.6)$$

A local product density estimate for a point x_i is given by,

$$\hat{\rho}_\epsilon^i(r) = \frac{n-1}{2\pi r a} \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r), \quad r > \epsilon > 0 \quad (4.7)$$

For fixed r , sum of local product density functions is proportional to the global product density, i.e.,

$$\hat{\rho}_\epsilon(r) = \frac{1}{n-1} \sum_{i=1}^n \hat{\rho}_\epsilon^i(r) \quad (4.8)$$

As yet we do not consider the edge effects. Simply an edge effect problem happens when we count the number of points inside a circle of radius r , centered on a point process inside \mathbf{W} but the circle extends outside \mathbf{W} . There are different methods to use for edge correction discussed in point process analysis and here we use translation edge correction method.

4.2.1.a Translation correction

Let's take a stationary point process \mathbf{X} means if we shift the process with vector s then the shifted process $\mathbf{X}+s$ is statistically equivalent to \mathbf{X} . For any random points x_i and x_j in \mathbf{X} have new positions x_i+s and x_j+s in $\mathbf{X}+s$. The vector difference $v=x_i - x_j$ from x_i to x_j remains unchanged. If we connect the points x_i and x_j by an arrow then the length and direction of the arrow remain fixed but the position is sifted by vector s . We can observe such an arrow inside \mathbf{W} only when both end points of x_i and x_j fall in \mathbf{W} .

Consider the possible positions of starting point x_i that give arrows inside \mathbf{W} for both end points x_i and x_j . It is clear that $x_i + v \in \mathbf{W}$ if and only if $x_i \in \mathbf{W} - v$. Both end points of the arrow fall inside \mathbf{W} if and only if $x_i \in \mathbf{W} \cap (\mathbf{W} - v)$. The probability $x_i \in \mathbf{W} \cap (\mathbf{W} - v)$ is related to the area of $\mathbf{W} \cap (\mathbf{W} - v)$. For any pair of points (x_i, x_j) , the translation edge correction weight is defined as $e_{ij} = |\mathbf{W}|/|\mathbf{W} \cap (\mathbf{W} - (x_i - x_j))|$.

In the study of N.Cressie and L.B.Collins, the edge corrected global estimate of the product density function is given by

$$\hat{\rho}_\epsilon(r) = \frac{1}{2\pi r a} \sum_{i=1}^n \sum_{j \neq i} \frac{2\pi \|x_i - x_j\|}{|\partial b(x_i, \|x_i - x_j\|) \cap \mathbf{W}|} f_\epsilon(\|x_i - x_j\| - r), \quad r > \epsilon > 0 \quad (4.9)$$

We treat the edge corrections of the above function with translation edge correction weights then

$$\hat{\rho}_\epsilon(r) = \frac{1}{2\pi r a} \sum_{i=1}^n \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r) \cdot e_{ij}, \quad r > \epsilon > 0 \quad (4.10)$$

Similarly, we can get the edge corrected product density LISA function as,

$$\hat{\rho}_\epsilon^i(r) = \frac{n-1}{2\pi r a} \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r) \cdot e_{ij}, \quad r > \epsilon > 0 \quad (4.11)$$

4.2.1.b Example

Let's consider a simulated 2D cluster point pattern (shown in Figure 4.4) of 203 points in $[0, 2] \times [2, 4]$ units window. We can plot the product density LISA functions of the pattern for both non edge corrected and edge corrected versions, i.e equation (4.7) and equation (4.11).

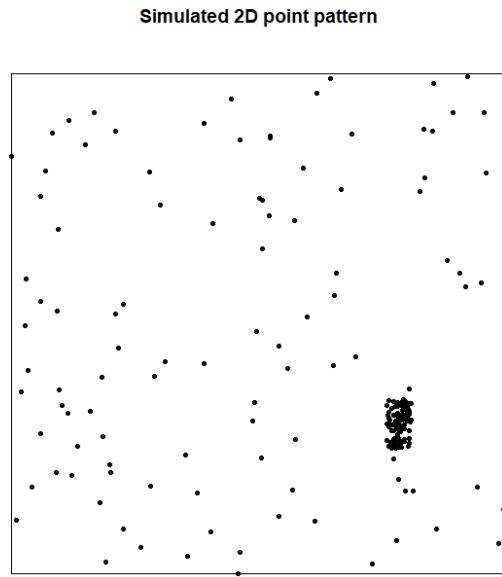


Figure 4.4: 2D simulated cluster point pattern

Figure 4.5 shows the product density LISA functions of the pattern without edge corrections. We can see high product density values for lower distances which represent a cluster behaviour.

The equation (4.11), Translation edge corrected product density LISA functions for the point pattern are plotted in Figure 4.6.

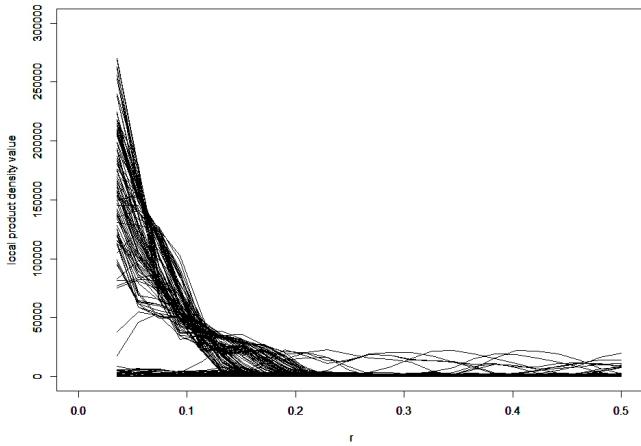


Figure 4.5: Product density LISA functions without edge correction

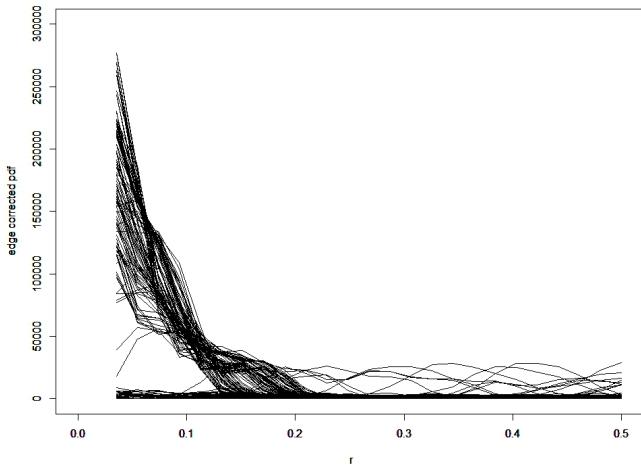


Figure 4.6: Edge corrected product density LISA functions

4.2.2 3D Point Processes

Here we develop the idea of 2D product density LISA functions for 3D point processes. If we consider a point process with n points in the 3-dimensional space in the region of $\mathbf{W} \subset \mathbb{R}^3$ of volume v then the definition for the K-function for 3D point processes ($K_3(r)$) is same as defined for the 2D case. It means that $\lambda K_3(r)$ is the expected number of points lying within a distance r of a typical point of the process. The intensity λ of a 3D point process is defined as

the expected number of points per unit volume.

$$K_3(r) = \frac{1}{\lambda} \mathbb{E}[\text{number of } r\text{-neighbours of } u | \mathbf{X} \text{ has a point at location } u] \quad (4.12)$$

for any $r \geq 0$ and any location u .

The product density function takes the form,

$$\rho_3(r) = \frac{\lambda^2 K'_3(r)}{4\pi r^2}, \quad r > 0 \quad (4.13)$$

We use here same kernel estimation of $\lambda^2 K'(r)$ to estimate $\lambda^2 K'_3(r)$ and replace area a by volume v .

$$\widehat{\lambda^2 K'_3(r)} = v^{-1} \sum_{i=1}^n \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r), \quad r > \epsilon > 0 \quad (4.14)$$

where $\|x_i - x_j\|$ is the 3-dimensional Euclidean distance between the points x_i and x_j . f_ϵ is the kernel function that we used in 2D point process given by the equation (4.2) and bandwidth $\epsilon = (5^{1/2}/10)\hat{\lambda}^{-1/2} = (5^{1/2}/10)v^{1/2}n^{-1/2}$.

By plugging the equation (4.14) to the equation (4.13), we can get the global kernel density estimate of the 3D product density function,

$$\hat{\rho}_3(r) = \frac{1}{4\pi r^2 v} \sum_{i=1}^n \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r), \quad r > \epsilon > 0 \quad (4.15)$$

If we consider any point x_i from the 3D point process \mathbf{X} , then the expected number of points with distance from x_i less than or equal to r of the process \mathbf{X} without counting the point x_i is given by

$$\{\lambda K_3(r)\}^i = E[\mathbf{X}(b(x_i, r) \setminus \{x_i\}) | x_i \in \mathbf{X}], \quad r > 0 \quad (4.16)$$

and the kernel estimate of $\{\lambda \widehat{K'_3(r)}\}^i$ is

$$\{\lambda \widehat{K'_3(r)}\}^i = \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r), \quad r > \epsilon > 0 \quad (4.17)$$

The unbiased estimator for the intensity of the reduced point process $\mathbf{X}_{x_i}^!$ is $(n-1)/v$ and

applying this value we can obtain the kernel density estimate for the derivative $\{\lambda^2 K'_3(r)\}^i$ for $r > 0$.

$$\hat{\lambda}\{\lambda\widehat{K'_3(r)}\}^i = (n-1)\mathbf{v}^{-1} \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r), \quad r > \epsilon > 0 \quad (4.18)$$

Hence, the local product density estimate for any point x_i is given by

$$\hat{\rho}_3^i(r) = \frac{n-1}{4\pi r^2 \mathbf{v}} \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r), \quad r > \epsilon > 0 \quad (4.19)$$

As we discussed in the section 4.2.1.a, we can develop the equation (4.19) for the edge correction. If we treat the local product density function with Translation edge correction weights e_{ij} of a 3D point process then the edge corrected local product density function of a 3D point process can be obtained by,

$$\hat{\rho}_3^i(r) = \frac{n-1}{4\pi r^2 \mathbf{v}} \sum_{j \neq i} f_\epsilon(\|x_i - x_j\| - r).e_{ij}, \quad r > \epsilon > 0 \quad (4.20)$$

4.2.2.a Example

A 3D point process is simulated in a box of $[0, 5] \times [0, 5] \times [0, 10]$ units. It has 198 points and some points belong to a cluster. The Figure 4.7 shows how the point pattern looks like. The Figure 4.8 shows the product density LISA functions of the simulated 3D point process which is the equation (4.20) without considering the edge effects. The translation edge corrected product density LISA functions are plotted in Figure 4.9.

Simulated 3D point process

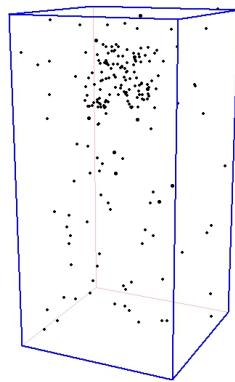


Figure 4.7: Simulated 3D point process

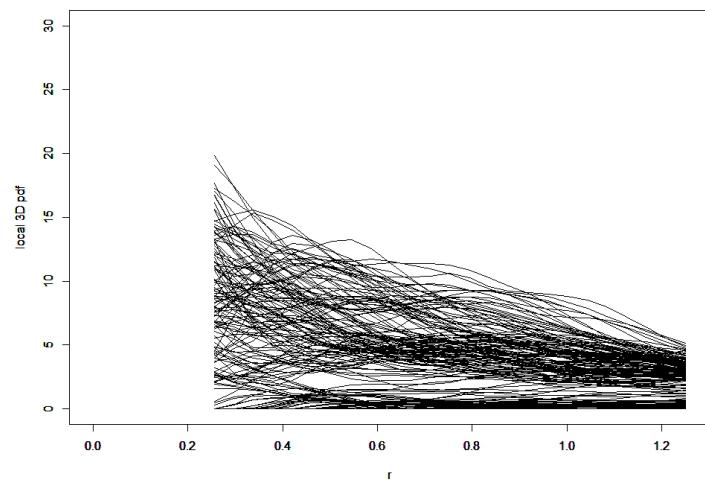


Figure 4.8: Product density LISA function with no edge effects

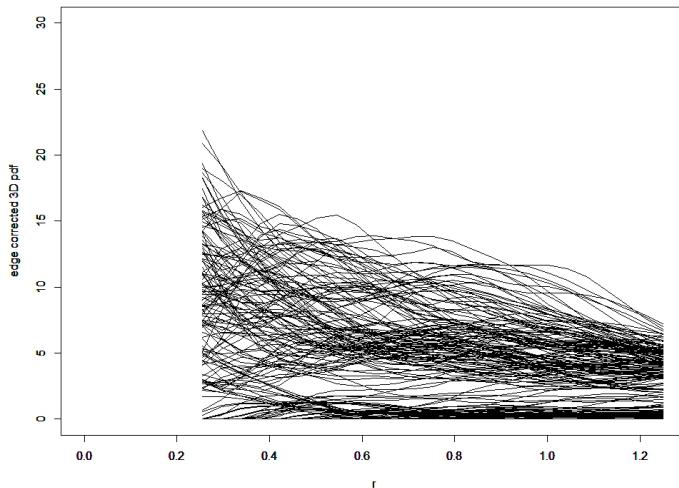


Figure 4.9: Edge corrected 3D product density LISA functions

4.3 Clustering of LISA functions

In this section we discuss the main part of the our study which is identifying the features of a cluster point pattern. We use local product densities of points detect the features. We make some assumptions to apply the cluster methods to our point patterns. We assume that the clutter of the point process is randomly distributed the observed window and the features are also distributed randomly in certain part of the window.

We used two different cluster methods in our study. The method of hierarchical clustering and EM algorithm are applied on local product density LISA functions to detect the features.

4.3.1 Hierarchical Cluster Analysis (HCA)

Hierarchical clustering

Hierarchical clustering[4] (or HCA) arranges points in a hierarchy based on the distance or similarity between them. The graphical representation of the resulting hierarchy is a tree-structured graph called a dendrogram. There are two type of hierarchical cluster methods, Divisive and Agglomerative.

In Divisive method (or "top-down" approach), initially all observations stay in one cluster.

At next step, select a cluster and split it into two sub clusters. The Divisive methods is run until each leaf cluster contains only one observation.

Agglomerative method (or "bottom-up" approach) assigns each observation to its own cluster at initial step. Then, compute the similarity between each of the clusters and merge two clusters which are most similar to each other. At final step, all observations are merged into a single cluster.

The difference between the Divisive and Agglomerative methods is shown in Figure 4.10.

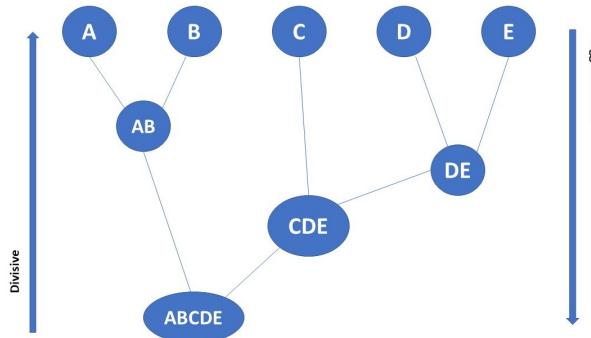


Figure 4.10: Traditional representation of Divisive and Agglomerative methods

Pairwise distances between observations are used to measure the similarity between set of observations to decide which cluster should be combined (in Agglomerative method) or where a cluster should be split (in Divisive method). The most common similarity measure is Euclidean distance. Some commonly used distances in HCA are,

Let's consider any two observations p and q from a data set in d -dimensional space

- Euclidean distance

$$\|p - q\|_2 = \sqrt{\sum_{i=1}^d (p_i - q_i)^2}$$

- Squared Euclidean distance

$$\|p - q\|_2^2 = \sum_{i=1}^d (p_i - q_i)^2$$

- Manhattan distance

$$\|p - q\|_1 = \sum_{i=1}^d |p_i - q_i|$$

- Maximum distance

$$\|p - q\|_\infty = \max_i |p_i - q_i|$$

The distances between each points are stored in a matrix which is called proximity matrix in HCA.

Linkage criteria is used to determine the distances between the clusters. If we consider two different clusters C_1, C_2 and observations $p \in C_1$ and $q \in C_2$ then some commonly used linkage methods can be defined as follows,

- Single-linkage

The distance between two clusters is defined as the shortest distance between two points in each cluster.

$$r_{min}(C_1, C_2) = \min r(p, q)$$

- Complete-linkage

The distance between two clusters is defined as the longest distance between two points in each cluster.

$$r_{max}(C_1, C_2) = \max r(p, q)$$

- Average-linkage

The distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.

$$r_{avg}(C_1, C_2) = \frac{1}{n_{C_1} n_{C_2}} \sum_{i=1}^{n_{C_1}} \sum_{j=1}^{n_{C_2}} r(p_i, q_j)$$

To extract the desired number of clusters k from a dendrogram, we can cut the dendrogram at a level that results in k sub-trees.

The result of HCA performed in **R** for mineral deposits data with Euclidean distance measure and complete linkage criteria is given by the Figure 4.11.

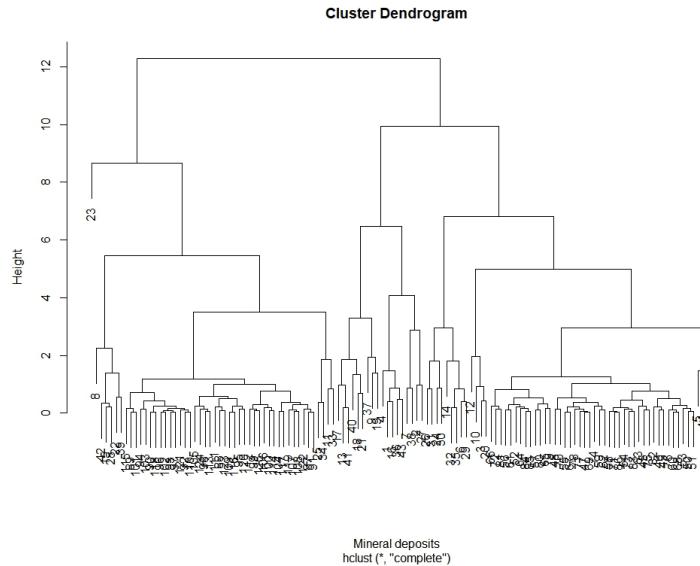


Figure 4.11: HCA of mineral deposits data

We can cut the dendrogram for given number of clusters or given height. If we cut the dendrogram for 3 clusters then the result is shown in Figure 4.12.

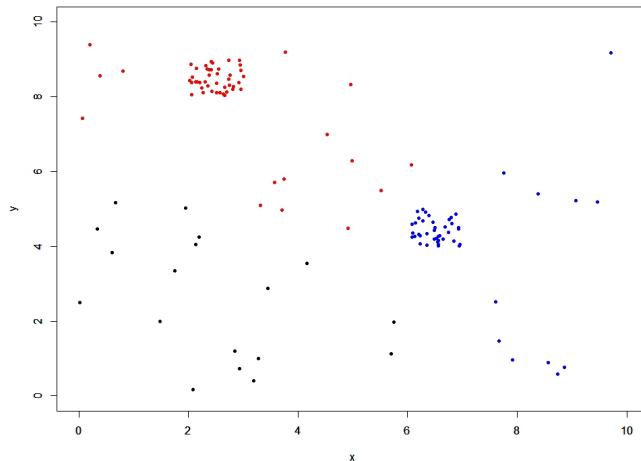


Figure 4.12: HCA of mineral deposits data with 3 clusters

In Figure 4.12 , we can see that many other locations of mineral are fallen to the clusters of gem deposits.

The clusters given by cutting the dendrogram at height 6 are shown in Figure 4.13. It gives 6 different groups. The Figure 4.13 shows that some locations of the red cluster in Figure 4.12 are removed when the dendrogram is cut at height 6.

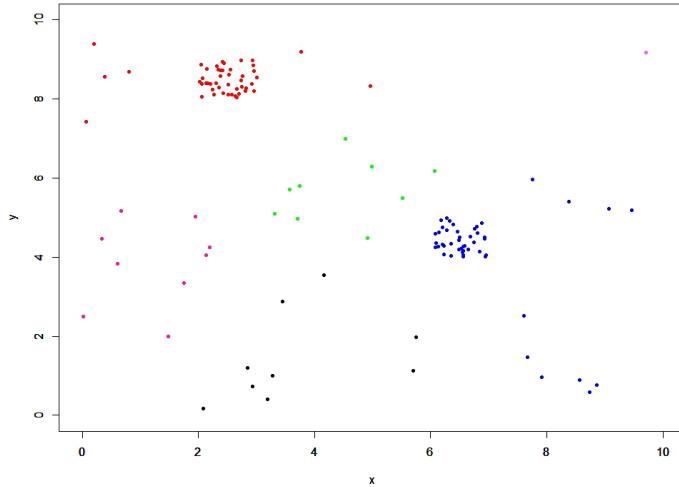


Figure 4.13: HCA of mineral deposits data, cutting at height 6.

By changing the height or number of clusters, we can remove the locations of clutter which are fallen to clusters. As an example if we cut the dendrogram to 11 clusters the we can remove many unwanted locations from the gem clusters. These 11 clusters are shown in Figure 4.14.

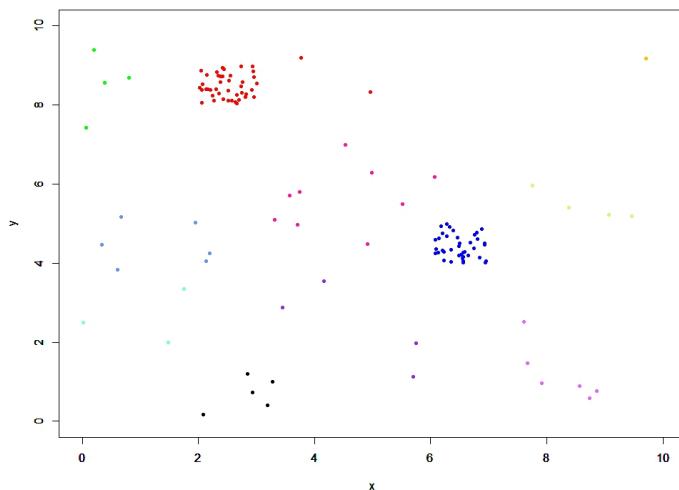


Figure 4.14: HCA of mineral deposits data with 11 clusters

Dissimilarities of local product densities can be analyzed using hierarchical cluster method[4]. HCA builds a hierarchy of clusters as we can see in Figure 4.2. We can apply HCA to LISA functions with or without considering their covariances.

In general if we consider any point x_i then the local product density LISA functions at each distance values of the interval of estimation are given by vector Y_i ,

$$Y_i = [\hat{\rho}^i(r_1), \hat{\rho}^i(r_2), \hat{\rho}^i(r_3), \dots, \hat{\rho}^i(r_k)]^T$$

where if we divide the distance of estimation into k intervals and we can observe all Y_i 's for $i=1\dots n$.

4.3.1.a HCA of LISA functions without covariances

We can arrange all LISA functions in a matrix that each row contains Y_i for the point x_i . The hierarchical cluster algorithm can be applied to the matrix of mean square distance (page 248 of [10]) between each local indicator functions. The result of HCA of edge corrected product density LISA functions of 2D and 3D point processes are shown in following Figure 4.15 and Figure 4.16 respectively. The function **hclust** in **R** was used to for hierarchical cluster analysis on simulated point patterns.

Both simulated point patterns contain only one cluster. To extract these clusters from each point patterns we need to separate the point patterns into clutter and features i.e two subsets. If we draw a horizontal line at height 50 in the dendrogram of 3D point process (Figure 4.16) then it cuts the tree at two branches and we can group the points connected to these two branches to separate the clusters. The **R** function **cutree** can be used to cut the three by giving either hight or an integer with the desired number of clusters (subsets).

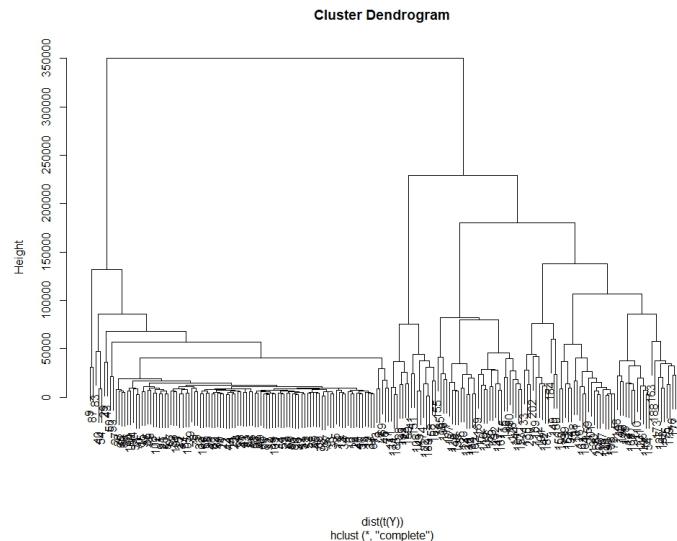


Figure 4.15: Dendrogram of hierarchical cluster analysis of 2D simulated point pattern

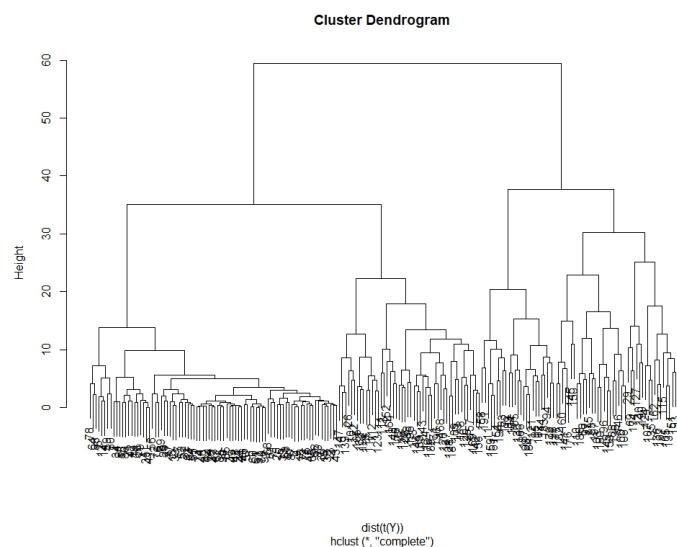


Figure 4.16: Dendrogram of hierarchical cluster analysis of 3D simulated point pattern

The feature points extracted from the **cutree** function in **R** for the 2D and 3D point patters are shown in Figure 4.17 and Figure 4.18.

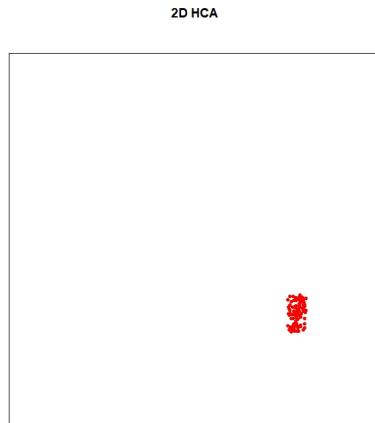


Figure 4.17: The cluster points detected by HCA of 2D simulated point pattern

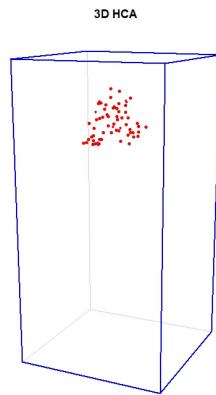


Figure 4.18: The cluster points detected by HCA of 3D simulated point pattern

The numerical comparison study for testing the accuracy of the hierarchical clustering is given in the Table 4.1.

Table 4.1: Numerical analysis of HCA without covariance

	True number of points		Estimated number of points				The accuracy	
	clutter	features	clutter		features			
			$c \leftrightarrow c$	$f \leftrightarrow c$	$f \leftrightarrow f$	$c \leftrightarrow f$		
2D	105	98	104	0	98	1	99.5%	
3D	92	106	88	44	62	4	75.75%	

The accuracy of the function nnclean is defined as the percentage value of
 $\frac{\text{number of points correctly identified for both cluster and clutter}}{\text{total number of points}}$.

The above table values show how the points were separated into clutter and cluster. In 2D point process, all clutter data comes from true clutter data set and only one clutter point is misclassified into feature data set. In same way we could see 44 cluster points and 4 clutter data were misclassified into clutter and feature set respectively in 3D point pattern.

According to the accuracy, we can recommend the method of hierarchical clustering without covariances for point process cluster analysis.

4.3.1.b HCA of LISA functions with covariances

For any two different points x_i and x_l , we can define the distance between their vectors Y_i and Y_l . Considering all pairs of points we can get a symmetric matrix of distances and the method of hierarchical clustering can be applied to this matrix.

The distance between Y_i and Y_l is defined as follows,

$$d_{il} = [Y_i - Y_l]^T \Sigma_{il}^{-1} [Y_i - Y_l] \quad (4.21)$$

where Σ_{il} is the covariance matrix for the vector $[Y_i - Y_l]$

The k^{th} diagonal element of Σ_{il} corresponding to the r_k is given by

$$\text{var}_{!!}[\hat{\rho}^i(r_k) - \hat{\rho}^l(r_k)] = \sigma_{ii}^{r_k r_k} + \sigma_{ll}^{r_k r_k} - 2\sigma_{il}^{r_k r_k} \quad (4.22)$$

and an off-diagonal element for distinct values r_k and r_m

$$\text{cov}_{!!}[(\hat{\rho}^i(r_k) - \hat{\rho}^l(r_k)), (\hat{\rho}^i(r_m) - \hat{\rho}^l(r_m))] = \sigma_{ii}^{r_k r_m} + \sigma_{ll}^{r_k r_m} - \sigma_{il}^{r_k r_m} - \sigma_{li}^{r_k r_m} \quad (4.23)$$

If we get the $n \times n$ symmetric matrix D of statistical distances d_{il} given by the equation (4.21) then the hierarchical cluster method can be applied to the matrix D for analysing dissimilarities of the local product densities.

Variances and Covariances of the edge-corrected product density LISA functions

To compute each d_{ij} values for a point process we must know the variances and covariances

of the product density LISA functions. First we consider the variance and covariance for 2D point process described in N.Cressies's study (see Appendix A)14.

If we consider any point x_i in given 2D point process then the variance of the edge corrected product density LISA function (equation (4.11)) is given by

$$\text{var}[\hat{\rho}_\epsilon^i(r)] = \frac{\lambda^3 + 3\lambda^2/a + \lambda/a^2}{2\pi r^2} \int_{r-\epsilon}^{r+\epsilon} e_{ij} s f_\epsilon^2(s-r) ds + 3(\lambda^3/a + \lambda^2/a^2) \quad (4.24)$$

and the covariance for the two different distances r_1 and r_2 is given as

$$\text{cov}[\hat{\rho}_\epsilon^i(r_1), \hat{\rho}_\epsilon^i(r_2)] = \frac{\lambda^3 + 3\lambda^2/a + \lambda/a^2}{(2\pi)^2 r_1 r_2} \int_{\mathbf{W}} g(x_i, x, r_1) g(x_i, x, r_2) dx + 3(\lambda^3/a + \lambda^2/a^2) \quad (4.25)$$

where $g(x, y, r) = e_{x,y} f_\epsilon(\|x - y\| - r)$.

To estimate the values of the equation (4.24) and the equation (4.25), we replace the λ by its unbiased estimator $\hat{\lambda} = (n - 1)/a$ which is unbiased estimator of the reduced point process $\mathbf{X}_{x_i}^!$.

For two distinct points x_i and x_l , the covariance between the product density LISA functions can be written as

$$\begin{aligned} \text{cov}_{!!}[\hat{\rho}_\epsilon^i(r_1), \hat{\rho}_\epsilon^l(r_2)] &= \frac{\lambda/a}{(2\pi)^2 r_1 r_2} g(x_i, x_l, r_1) g(x_l, x_i, r_2) + \frac{\lambda^2/a}{\pi} \left[\frac{g(x_i, x_l, r_1)}{r_1} + \frac{g(x_l, x_i, r_2)}{r_2} \right] \\ &\quad + \frac{\lambda^3 + 5\lambda^2/a + 2\lambda/a^2}{(2\pi)^2 r_1 r_2} \int_{\mathbf{W}} g(x_i, x, r_1) g(x_l, x, r_2) dx + 3\lambda^3/a + 5\lambda^2/a^2 \end{aligned} \quad (4.26)$$

In the equation (4.26), the expectations are taken respect to the doubly reduced (x_i and x_l) point process $\mathbf{X}_{x_i, x_l}^{!!}$ and covariance for the two different distances in $\mathbf{X}_{x_i, x_l}^{!!}$ takes the form,

$$\begin{aligned} \text{cov}_{!!}[\hat{\rho}_\epsilon^i(r_1), \hat{\rho}_\epsilon^i(r_2)] &= \frac{\lambda/a}{(2\pi)^2 r_1 r_2} g(x_i, x_l, r_1) g(x_i, x_l, r_2) + \frac{\lambda^2/a}{\pi} \left[\frac{g(x_i, x_l, r_1)}{r_1} + \frac{g(x_i, x_l, r_2)}{r_2} \right] \\ &\quad + \frac{\lambda^3 + 5\lambda^2/a + 2\lambda/a^2}{(2\pi)^2 r_1 r_2} \int_{\mathbf{W}} g(x_i, x, r_1) g(x_i, x, r_2) dx + 3\lambda^3/a + 5\lambda^2/a^2 \end{aligned} \quad (4.27)$$

The unbiased estimator of λ for the doubly reduced point process is $\hat{\lambda} = (n - 2)/a$ and similarly, λ^2 and λ^3 can be replaced as $\hat{\lambda}^2 = (n - 2)(n - 3)/a^2$ and $\hat{\lambda}^3 = (n - 2)(n - 3)(n - 4)/a^3$.

The two dimensional integration parts in both equations (4.26) and (4.27) are approximated by simple Monte Carlo integration. First we take a set of M points ($\{x_m\}$) which are uniformly distributed in the domain of integrand and define $h(x) = g(x_i, x, r_1) g(x_l, x, r_2)$. The integral of $h(x)$ is estimated by the empirical average of the integrand, which is $(1/M) \sum_{m=1}^M h(x_m)$, multiplied by the area of the domain.

The equations of the covariances with Monte Carlo integral approximation are given by

$$\begin{aligned} \text{cov}_{!!}[\hat{\rho}_\epsilon^i(r_1), \hat{\rho}_\epsilon^i(r_2)] &= \frac{\lambda/a}{(2\pi)^2 r_1 r_2} g(x_i, x_l, r_1) g(x_l, x_i, r_2) + \frac{\lambda^2/a}{\pi} \left[\frac{g(x_i, x_l, r_1)}{r_1} + \frac{g(x_l, x_i, r_2)}{r_2} \right] \\ &\quad + \frac{\lambda^3 + 5\lambda^2/a + 2\lambda/a^2}{(2\pi)^2 r_1 r_2} \frac{a}{M} \sum_{m=1}^M g(x_i, x_m, r_1) g(x_l, x_m, r_2) + 3\lambda^3/a + 5\lambda^2/a^2 \end{aligned} \quad (4.28)$$

$$\begin{aligned} \text{cov}_{!!}[\hat{\rho}_\epsilon^i(r_1), \hat{\rho}_\epsilon^i(r_2)] &= \frac{\lambda/a}{(2\pi)^2 r_1 r_2} g(x_i, x_l, r_1) g(x_i, x_l, r_2) + \frac{\lambda^2/a}{\pi} \left[\frac{g(x_i, x_l, r_1)}{r_1} + \frac{g(x_i, x_l, r_2)}{r_2} \right] \\ &\quad + \frac{\lambda^3 + 5\lambda^2/a + 2\lambda/a^2}{(2\pi)^2 r_1 r_2} \frac{a}{M} \sum_{m=1}^M g(x_i, x_m, r_1) g(x_i, x_m, r_2) + 3\lambda^3/a + 5\lambda^2/a^2 \end{aligned} \quad (4.29)$$

Let's consider a 3D point process now. For any point x_i , the expectation of edge corrected product density LISA function (equation (4.20)) is given as

$$E_![\hat{\rho}_3^i(r)] = \frac{1}{4\pi r^2 \mathbf{v}} E_![(\mathbf{X}(\mathbf{W}) - 1) \sum_{j \neq i} e_{ij} f_\epsilon(\|x_i - x_j\| - r)] \quad (4.30)$$

The random variable $\mathbf{X}(\mathbf{W}) - 1$ follows Poisson distribution with mean $\lambda\mathbf{v}$ with respect to the reduced point process distribution $\mathbf{X}_{x_i}^!$. Then $E_![(\mathbf{X}(\mathbf{W}) - 1)^2] = \lambda^2\mathbf{v}^2 + \lambda\mathbf{v}$. By considering the area of a Bin with bandwidth ϵ which contains all points x_j and expected value of reduced point distribution, we can rewrite the above equation (4.30) as follows

$$\begin{aligned}
 E_![\hat{\rho}_3^i(r)] &= \frac{1}{4\pi r^2 \mathbf{v}} E_![E\{(n-1) \sum_{j \neq i} e_{ij} f_\epsilon(\|x_i - x_j\| - r) | \mathbf{X}(\mathbf{W}) = n\}] \\
 &= \frac{1}{4\pi r^2 \mathbf{v}} E_![\mathbf{X}(\mathbf{W}) - 1]^2 \frac{1}{\mathbf{v}} \int_W e_{ij} f_\epsilon(\|x_i - x\| - r) dx \\
 &= \frac{1}{4\pi r^2 \mathbf{v}} (\lambda^2 \mathbf{v}^2 + \lambda \mathbf{v}) \frac{1}{\mathbf{v}} \int_{r-\epsilon}^{r+\epsilon} \frac{|\mathbf{W}|}{|\mathbf{W} \cap (\mathbf{W} - s)|} f_\epsilon(s - r) \times |\mathbf{W} \cap (\mathbf{W} - s)| ds \quad (4.31) \\
 &= \frac{\lambda^2 + (\lambda/\mathbf{v})}{4\pi r^2} \int_{r-\epsilon}^{r+\epsilon} \mathbf{W} f_\epsilon(s - r) ds \\
 &= \lambda^2 + (\lambda/\mathbf{v})
 \end{aligned}$$

In similar way of 2D case , we can define variance and covariance for a 3D product density LISA function as,

$$\text{var}_![\hat{\rho}_3^i(r)] = \frac{\lambda^3 + 3\lambda^2/\mathbf{v} + \lambda/\mathbf{v}^2}{4\pi r^4} \int_{r-\epsilon}^{r+\epsilon} e_{ij} s^2 f_\epsilon^2(s - r) ds + 3(\lambda^3/\mathbf{v} + \lambda^2/\mathbf{v}^2) \quad (4.32)$$

$$\text{cov}_![\hat{\rho}_3^i(r_1), \hat{\rho}_3^i(r_2)] = \frac{\lambda^3 + 3\lambda^2/\mathbf{v} + \lambda/\mathbf{v}^2}{(4\pi)^2 r_1^2 r_2^2} \int_{\mathbf{W}} g_3(x_i, x, r_1) g_3(x_i, x, r_2) dx + 3(\lambda^3/\mathbf{v} + \lambda^2/\mathbf{v}^2) \quad (4.33)$$

where $g_3(x, y, r) = e_{x,y} f_\epsilon(\|x - y\| - r)$.

To obtain the values of the equations(4.32) and (4.33) from the data , we can replace λ by its unbiased estimator $\hat{\lambda} = (n - 1)/\mathbf{v}$.

From a homogeneous Poisson process, we can define the covariance between two product density LISA functions of two distinct points x_i and x_l .

$$\begin{aligned} \text{cov}_{!!}[\hat{\rho}_3^i(r_1), \hat{\rho}_3^l(r_2)] &= \frac{\lambda/\mathbf{v}}{(4\pi)^2 r_1^2 r_2^2} g_3(x_i, x_l, r_1) g_3(x_l, x_i, r_2) + \frac{\lambda^2/\mathbf{v}}{\pi} \left[\frac{g_3(x_i, x_l, r_1)}{r_1^2} + \frac{g_3(x_l, x_i, r_2)}{r_2^2} \right] \\ &\quad + \frac{\lambda^3 + 5\lambda^2/\mathbf{v} + 2\lambda/\mathbf{v}^2}{(4\pi)^2 r_1^2 r_2^2} \int_{\mathbf{W}} g_3(x_i, x, r_1) g_3(x_l, x, r_2) dx + 3\lambda^3/\mathbf{v} + 5\lambda^2/\mathbf{v}^2 \end{aligned} \quad (4.34)$$

The covariance for the two different distances is given by,

$$\begin{aligned} \text{cov}_{!!}[\hat{\rho}_3^i(r_1), \hat{\rho}_3^i(r_2)] &= \frac{\lambda/\mathbf{v}}{(4\pi)^2 r_1^2 r_2^2} g_3(x_i, x_l, r_1) g_3(x_i, x_l, r_2) + \frac{\lambda^2/\mathbf{v}}{\pi} \left[\frac{g_3(x_i, x_l, r_1)}{r_1^2} + \frac{g_3(x_i, x_l, r_2)}{r_2^2} \right] \\ &\quad + \frac{\lambda^3 + 5\lambda^2/\mathbf{v} + 2\lambda/\mathbf{v}^2}{(4\pi)^2 r_1^2 r_2^2} \int_{\mathbf{W}} g_3(x_i, x, r_1) g_3(x_i, x, r_2) dx + 3\lambda^3/\mathbf{v} + 5\lambda^2/\mathbf{v}^2 \end{aligned} \quad (4.35)$$

To obtain the values of the equations (4.34) and (4.35) from the data, we can replace λ , λ^2 and λ^3 from their unbiased estimators $\hat{\lambda} = (n-2)/\mathbf{v}$, $\hat{\lambda}^2 = (n-2)(n-3)/\mathbf{v}^2$ and $\hat{\lambda}^3 = (n-2)(n-3)(n-4)/\mathbf{v}^3$.

As we used Monte Carlo integration to evaluate the integral parts in 2D equations of covariances, here also we use Monte Carlo integration to evaluate the integral parts in the equations (4.34) and (4.35) by using a simulated 3D point pattern which is uniformly distributed in the domain of the integrand. Then the covariances with Monte Carlo approximation are given by

$$\begin{aligned} \text{cov}_{!!}[\hat{\rho}_3^i(r_1), \hat{\rho}_3^l(r_2)] &= \frac{\lambda/\mathbf{v}}{(4\pi)^2 r_1^2 r_2^2} g_3(x_i, x_l, r_1) g_3(x_l, x_i, r_2) + \frac{\lambda^2/\mathbf{v}}{\pi} \left[\frac{g_3(x_i, x_l, r_1)}{r_1^2} + \frac{g_3(x_l, x_i, r_2)}{r_2^2} \right] \\ &\quad + \frac{\lambda^3 + 5\lambda^2/\mathbf{v} + 2\lambda/\mathbf{v}^2}{(4\pi)^2 r_1^2 r_2^2} \frac{\mathbf{v}}{M} \sum_{m=1}^M g(x_i, x_m, r_1) g(x_l, x_m, r_2) + 3\lambda^3/\mathbf{v} + 5\lambda^2/\mathbf{v}^2 \end{aligned} \quad (4.36)$$

$$\begin{aligned} \text{cov}_{!!}[\hat{\rho}_3^i(r_1), \hat{\rho}_3^i(r_2)] &= \frac{\lambda/\mathbf{v}}{(4\pi)^2 r_1^2 r_2^2} g_3(x_i, x_l, r_1) g_3(x_i, x_l, r_2) + \frac{\lambda^2/\mathbf{v}}{\pi} \left[\frac{g_3(x_i, x_l, r_1)}{r_1^2} + \frac{g_3(x_i, x_l, r_2)}{r_2^2} \right] \\ &\quad + \frac{\lambda^3 + 5\lambda^2/\mathbf{v} + 2\lambda/\mathbf{v}^2}{(4\pi)^2 r_1^2 r_2^2} \frac{\mathbf{v}}{M} \sum_{m=1}^M g(x_i, x_m, r_1) g(x_i, x_m, r_2) + 3\lambda^3/\mathbf{v} + 5\lambda^2/\mathbf{v}^2 \end{aligned} \quad (4.37)$$

The dendrogram of the matrix D of 2D point process is given in the Figure 4.19 and the Figure 4.20 shows the dendrogram of the matrix D of 3D point process.

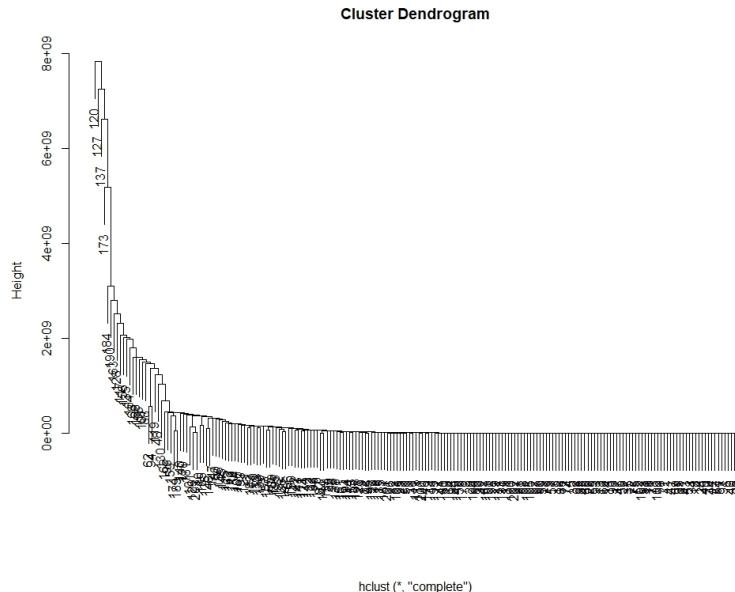


Figure 4.19: Dendrogram of the matrix D of 2D simulated point pattern

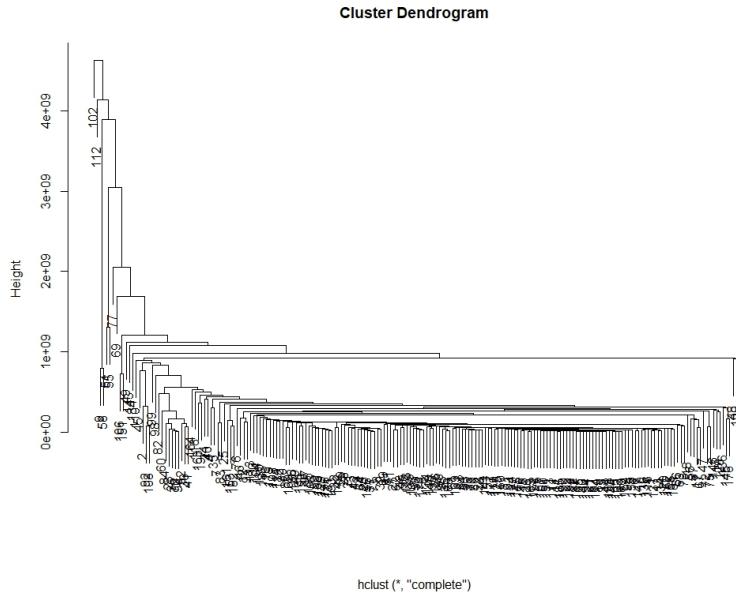


Figure 4.20: Dendrogram of the matrix D of 3D simulated point pattern

If we cut the above dendrograms of the 2D and 3D point processes into two groups then the dendrograms give one cluster with single points and the other cluster with all other points in both 2D and 3D cases. It is clear that the obtained results of the method show unexpected behaviour. The solutions to correct the method for getting acceptable results are explained in the section of final remarks.

4.3.2 EM algorithm

In this section first we briefly review how does EM (Expectation-Maximization) work.

EM algorithm is an iterative method to find maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables[15]. That is, either missing values exist among the data, or the model can be formulated more simply by assuming the existence of further unobserved data points. For example, a mixture model can be described more simply by assuming that each observed data point has a corresponding unobserved data point, or latent variable, specifying the mixture component to which each data point belongs. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

Derivation of EM-algorithm

Suppose we have a training set \mathbf{X} of observed data, a set of unobserved latent data or missing values \mathbf{Z} , and a vector of unknown parameters θ , along with a likelihood function $L(\theta; \mathbf{X}, \mathbf{Z}) = p(\mathbf{X}, \mathbf{Z}|\theta)$, the maximum likelihood estimate of the unknown parameters is determined by maximizing the marginal likelihood of the observed data

$$L(\theta; \mathbf{X}) = p(\mathbf{X}|\theta) = \int p(\mathbf{X}, \mathbf{Z}|\theta) d\mathbf{Z} \quad (4.38)$$

The EM algorithm seeks to find the MLE of the marginal likelihood by iteratively applying following two steps:

E step

Expectation step (E step): Define $\mathbf{Q}(\theta|\theta^{(t)})$ as the expected value of the log likelihood function of θ , with respect to the current conditional distribution of \mathbf{Z} given \mathbf{X} and the current estimates of the parameters $\theta^{(t)}$:

$$\mathbf{Q}(\theta|\theta^{(t)}) = \mathbf{E}_{\mathbf{Z}|\mathbf{X}, \theta^{(t)}}[\log L(\theta; \mathbf{X}, \mathbf{Z})] \quad (4.39)$$

M step

Maximization step (M step): Find the parameters that maximize the above quantity:

$$\theta^{(t+1)} = \arg \max_{\theta} \mathbf{Q}(\theta | \theta^{(t)}) \quad (4.40)$$

4.3.2.a NN distances and EM algorithm

The k^{th} nearest neighbour distances of a point process can be treated using EM algorithm to separate the clusters and clutter. In **R** this method is done by the **nnclean** function discussed in section 3.1. Here we overview the mathematical functions used in this method.

The method was presented by Byers and Raftery[7] and they assumed that

- The features are distributed as a homogeneous Poisson point process.
- The clutter is distributed as a Poisson process restricted to a certain part of the image and overlaid on the cluster.

Let's start the method from **2D** point process. Under the assumptions of a homogeneous Poisson process, the nearest-neighbour distance distribution function (page 264 of [10]) is

$$G(d) = 1 - \exp(-\lambda\pi d^2) \quad (4.41)$$

this function is used to get the distribution function of the distance to k^{th} nearest neighbour D_k [12].

$$D_k(d) = 1 - \sum_{j=0}^{k-1} \exp(-\lambda\pi d^2) \frac{(\lambda\pi d^2)^j}{j!} \quad (4.42)$$

then the probability density function ($D'_k(d)$) of the k^{th} NN distance [12] f_{D_k} of a randomly chosen point in the process is given by

$$f_{D_k}(d) = \frac{e^{-\lambda\pi d^2} 2(\lambda\pi)^k d^{2k-1}}{(k-1)!} \quad (4.43)$$

where d is the nearest neighbour distance ($d \geq 0$).

If we take a variable Y as $(D_k)^2$ then Y follows a Gamma distribution which is $Y \sim \Gamma(k, \lambda\pi)$

[13]. The mixture model of nearest neighbour distances of features and clutter is given by

$$D_k \sim \tau \Gamma^{(1/2)}(k, \lambda_1 \pi) + (1 - \tau) \Gamma^{(1/2)}(k, \lambda_2 \pi) \quad (4.44)$$

where τ is the mixing probability and λ_1, λ_2 are the intensities of feature and clutter distributions respectively.

Now we can apply the EM algorithm to the mixture model, the equation (4.44). The classification variable δ is defined as $\delta_i \in \{0, 1\}$ for each data point where $\delta_i = 1$ if i^{th} point comes from feature distribution and $\delta_i = 0$ if i^{th} point comes from clutter distribution. EM algorithm estimates the values of the parameter set $(\tau, \lambda_1, \lambda_2)$ for given k .

E- step

$$E(\widehat{\delta}_i^{(t+1)}) = \frac{\widehat{\tau}^{(t)} f_{D_k}(d_i, \widehat{\lambda}_1^{(t)})}{\widehat{\tau}^{(t)} f_{D_k}(d_i, \widehat{\lambda}_1^{(t)}) + (1 - \widehat{\tau}^{(t)}) f_{D_k}(d_i, \widehat{\lambda}_2^{(t)})} \quad (4.45)$$

M- step

$$\widehat{\tau}^{(t+1)} = \sum_{i=1}^n \frac{\widehat{\delta}_i^{(t+1)}}{n} \quad (4.46)$$

$$\widehat{\lambda}_1^{(t+1)} = \frac{k \sum_{i=1}^n \widehat{\delta}_i^{(t+1)}}{\pi \sum_{i=1}^n d_i^2 \widehat{\delta}_i^{(t+1)}} \quad (4.47)$$

$$\widehat{\lambda}_2^{(t+1)} = \frac{k \sum_{i=1}^n (1 - \widehat{\delta}_i^{(t+1)})}{\pi \sum_{i=1}^n d_i^2 (1 - \widehat{\delta}_i^{(t+1)})} \quad (4.48)$$

To applying EM algorithm for NN distances of a **3D** point process, we only need the nearest-neighbour distance distribution function of 3D point process $G_3(d)$ (page 655 of [10]).

$$G_3(d) = 1 - \exp(-\frac{4}{3} \lambda \pi d^3) \quad (4.49)$$

The steps for EM algorithm can be continued same as 2D case replacing $G(d)$ by $G_3(d)$.

The **nnclean** function was applied to simulated 2D and 3D point process discussed in the Example 4.2.1.b and the Example 4.2.2.a. The results are as follows.

The Figure 4.21 and the Figure 4.22 show how cluster points have been detected for different nearest neighbours. For $k = 1$, function detects small clusters in the patterns but if we are not interested in these clusters with small number of points then we can remove them by considering 2nd, 3th,... nearest neighbours, i.e increasing the value of k .

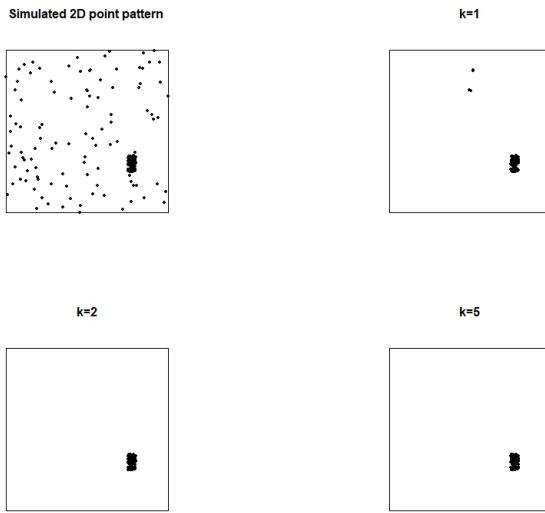


Figure 4.21: Feature detection from nnclean in 2D simulated point pattern

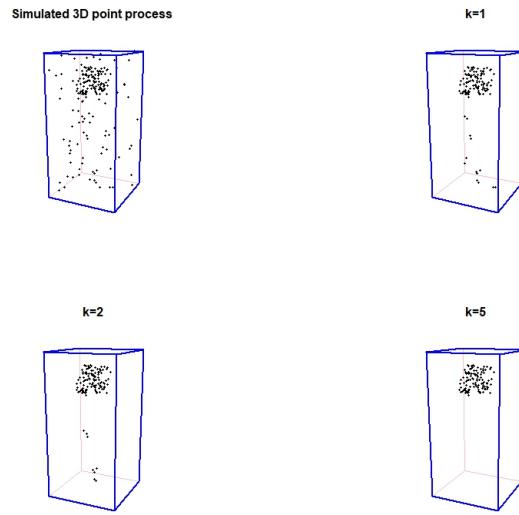


Figure 4.22: Feature detection from nnClean in 3D simulated point pattern

Table 4.2: Summary of nnClean function in 2D point pattern

	True value	k=1	k=2	k=5
Intensity of clutter	25	25.172	24.449	22.251
Intensity of feature	5000	4207.9	3986.9	4185.3
Number of points in feature	98	100	99	99
Number of points correctly fall into the feature	-	95	98	98
Accuracy of nnClean	-	96.06%	99.5%	99.5%

Table 4.3: Summary of nnClean function in 3D point pattern

	True value	k=1	k=2	k=5
Intensity of clutter	0.4	0.34588	0.26116	0.2201
Intensity of feature	10	9.8025	9.0748	9.3338
Number of points in feature	106	125	122	114
Number of points correctly fall into the feature	-	106	106	106
Accuracy of nnClean	-	90.4%	91.91%	95.95%

If we consider the numerical values given in Table 4.2 and Table 4.3, we can see how the

difference between true and estimate values of the parameters is changed when k is increased. For $k = 1$, parameter estimation values are very closed to the true values but the accuracy of detecting points for clutter and features is lower than $k = 2$ and $k = 5$. It is clear that we can select a value for k which gives high accuracy and low error of parameter estimation.

4.3.3 Method of SEM

In this section we discuss about how to use stochastic EM algorithm and product density LISA functions of a clustered point process together for feature-clutter classification. The method of SEM presented in this thesis is an re-arrangement of the SEM method published by J.Mateu et al. 2007[13]. In the section (4.3.2), we discussed about the EM algorithm for the nearest neighbour distances and the distribution function of the k -nn distance is known. In the stochastic version of EM, We take the distances between the edge corrected LISA functions and its expected value under CSR and the distribution function of these distances is unknown. Hence, we assume the density values of these distances using kernel density approximation from a simulated point process.

The assumptions made in this method are,

- The clutter is randomly distributed in the region and follows a homogeneous Poisson process.
- The features are distributed in a certain part of the region and overlaid on the clutter following a Poisson process.

For a given point process \mathbf{X} , we can define a marked point process $Y = \{x_i, y_i, d_i, \delta_i\}_{i=1}^n$, where (x_i, y_i) are the co-ordinates of the location of the i^{th} point of the original point pattern, d_i is the distance between the edge corrected LISA functions and its expected value under a homogeneous Poisson process. $\delta_i \in \{0, 1\}$ and $\delta_i = 1$ if the i^{th} point belongs to a feature, otherwise $\delta_i = 0$. The EM algorithm plays a role in SEM method to estimate the classification parameter δ_i .

To measure the distance d_i , we can use different distance measures which are L_1 distance, L_2 distance [8] and Kullback symmetric divergence (KSD). The expected value of a LISA

function under homogeneous Poisson process is given by $E[\hat{\rho}_\epsilon^i(r)] = \lambda^2 + \frac{\lambda}{a}$ which is same as proven in the equation (4.31). The unknown density function of distances d_i is denoted as $f_{DLISA}(d_i)$.

The d_i distances are supposed to be distributed as a mixture of two distributions. We can see the difference between feature and clutter by plotting a histogram of distances d_i of a point process when the histogram becomes clearly bimodal. The Figure 4.23 shows the histogram of L_1 distances of the simulated 2D point process.

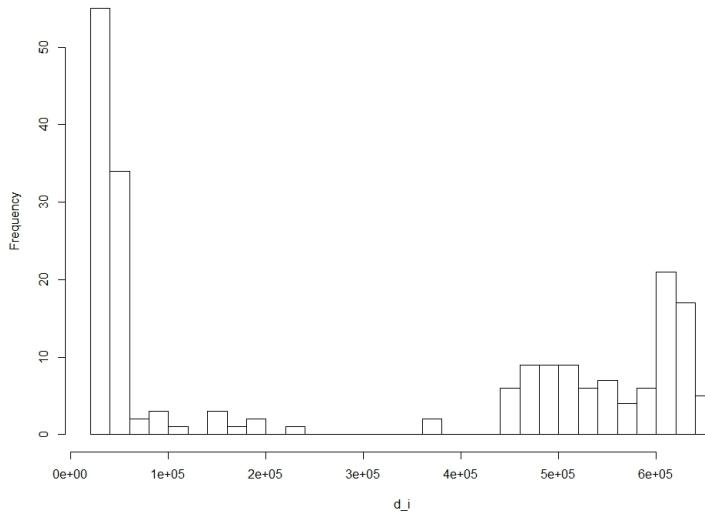


Figure 4.23: Histogram of L_1 distances of the 2D simulated point pattern

The steps of the SEM algorithm are as follows,

- First d_i distances are calculated for the original clustered point pattern.
- Set initial approximations for the classification parameter δ , number of feature points n_f , number of clutter points n_c , intensity of feature process λ_f and intensity of clutter process λ_c .

To initial guess of δ , we use the values of d_i . We set $\delta_i = 0$ for the points which take small values of d_i and $\delta_i = 1$ for large values of d_i . One can separate d_i into small and large values as d_i which are smaller and larger than its mean value. The initial δ_i values are used to make other initial approximations.

The initial n_f = number of $\{\delta_i = 1\}$

The initial n_c = number of $\{\delta_i = 0\}$

$$\text{The initial } \lambda_f = \frac{n_f}{\mathbf{a}}$$

$$\text{The initial } \lambda_c = \frac{n_c}{\mathbf{a}}$$

where \mathbf{a} is the area of the observation window of the original clustered point pattern.

- S-step: Simulate clutter and feature processes

In this step we simulate points for the clutter and feature processes. Points for the clutter are simulated in the region of the original point process \mathbf{W} . Then we select a region for the feature points which lies in the original region and approximately equal to the area of the original feature points and let's denote it as \mathbf{w}_f .

The important criteria of simulating feature and clutter processes is each d_i value must be fallen into at least region of the density function of simulated clutter or feature process. The intensity plays key role to simulate points which are satisfied the above criteria. First we start selecting the intensities for the simulated point process by initial number of points that are estimated in the second step. Then the initial intensity for the clutter is $\frac{\text{The initial } n_c}{\mathbf{w}}$ and the initial intensity of the feature is $\frac{\text{The initial } n_f}{\mathbf{w}_f}$.

Now we fit the simulated clutter and feature points in the observation window of the original process to get a mixture of feature and clutter points. Then we calculate LISA functions for the simulated mixture process and take the distances between each LISA functions and the expected values. These distance values of the feature points are denoted as d_f and d_c denotes the distances of the clutter points.

The densities of d_f and d_c are estimated by kernel density estimation[5] and denoted as $f_{D_{\text{LISA}}}(d_f)$ and $f_{D_{\text{LISA}}}(d_c)$ respectively. Then the d_i values are checked to see whether they fall into the range of density functions of simulated processes.

We can simulate points for the clutter and feature by increasing the number of points till they satisfy the given criteria.

- EM-algorithm

We use non parametric density function in SEM method. Then we do not have to estimate any parameter value for the density function. We start the EM algorithm by giving initial values for the set of parameters $(\tau, n_c, n_f, \lambda_c, \lambda_f)$.

In t^{th} iteration, for a given d_i ,

E-step

We estimate the classification parameter δ_i using the equation 4.45. If d_i , falls in the region of the simulated density function of clutter (and feature) then the density value of d_i is estimated by linear interpolation[6]. If d_i does not lie in the region of density function of the simulated clutter (and feature) then the density value of d_i is assigned to zero.

M-step

In this step, we update the parameters for the next iteration.

$$\tau^{t+1} = \frac{\sum_{i=1}^n \delta_i}{n}$$

We take another parameter \mathbf{z} for classification of δ_i . For any d_i ,

$$\mathbf{z}_i = \begin{cases} 1 & \text{if } \delta_i > 1 - \delta_i \\ 0 & \text{if } \delta_i < 1 - \delta_i \end{cases}$$

Then,

$$n_f^{t+1} = \text{number of } \mathbf{z} | \mathbf{z} = 1$$

$$n_c^{t+1} = \text{number of } \mathbf{z} | \mathbf{z} = 0$$

$$\lambda_f^{t+1} = \frac{n_f^{t+1}}{W}$$

$$\lambda_c^{t+1} = \frac{n_c^{t+1}}{W}$$

The SEM method for 3D point process takes same steps as mentioned above. In 3D SEM method, we take the LISA functions of 3D point patterns and simulate the clutter and feature points in 3D boxes.

SEM method for 2D and 3D simulated point processes

Here we discuss the results of SEM method of simulated 2D and 3D point processes which are given in example 4.2.1.b and example 4.2.2.a.

We used a clutter with intensity 76 (304 points) and a feature with intensity 3500 (points 38) for s-step in 2D SEM algorithm. Clutter points are simulated in the same window of 2D simulated clustered process, $[0, 2] \times [2, 4]$ units window. Feature points are simulated in the $[0.2, 0.3] \times [2.8, 2.9]$ units window. The Figure 4.24 shows the mixture of feature-clutter process which is used in s-step in SEM method for 2D case.

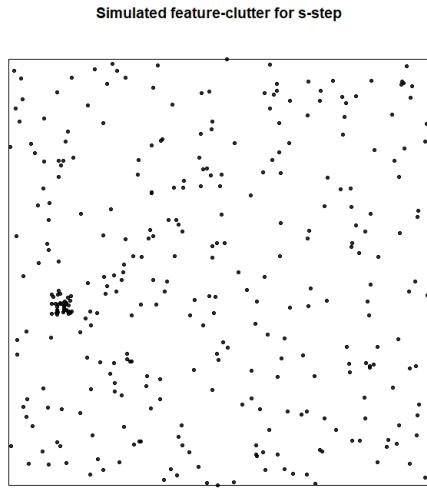


Figure 4.24: 2D simulated feature and clutter points for s-step in SEM

L_1 and L_2 are used to measure the distances. SEM algorithm is tested with $\tau = 0.2$ and $\tau = 0.6$. The number of iterations is 20.

The Figure 4.25 shows the values of the parameter set $(\tau, n_c, n_f, \lambda_c, \lambda_f)$ in each iteration for L_1 distance measure with initial $\tau = 0.2$. The convergence of parameters with L_1 distances with initial $\tau = 0.6$ shows in the Figure 4.26. The same plots for L_2 distances with initial $\tau = 0.2$ and $\tau = 0.6$ are shown in the Figure 4.27 and the Figure 4.28 respectively.

In all these four different studies of SEW for 2D simulated process, we could get same result. That means feature-clutter classifications of L_1, L_2 distances for both $\tau = 0.2$ and $\tau = 0.6$ are same. The numerical results are shown in the Table 4.4 and the Figure 4.29 shows the feature points which are detected by SEM method .

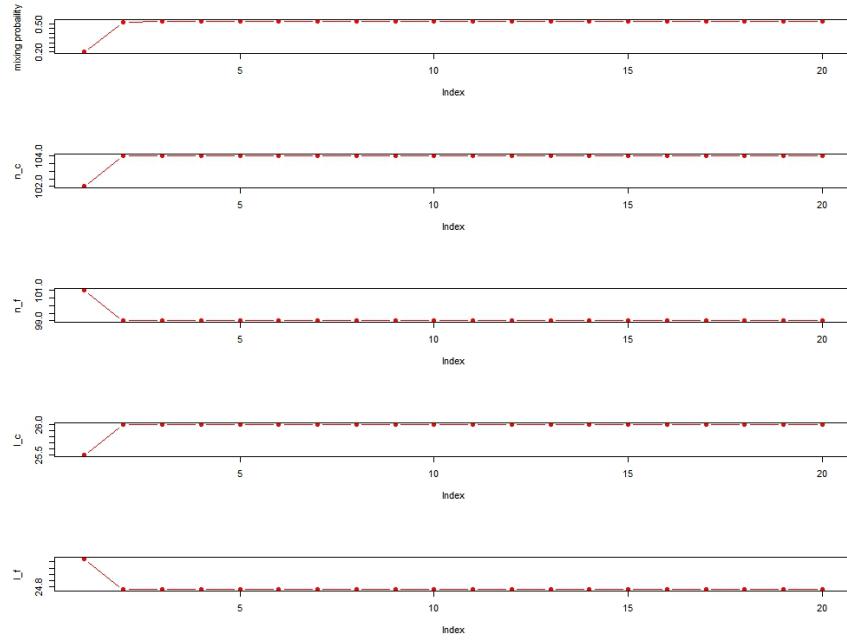


Figure 4.25: Convergence of parameters of SEM algorithm in 2D case- for $\tau = 0.2$ and L_1 distances

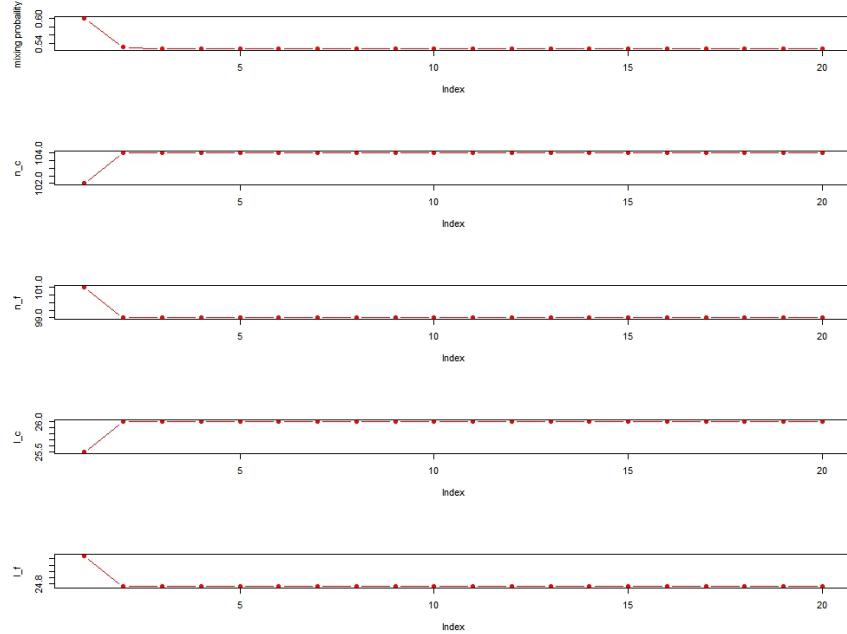


Figure 4.26: Convergence of parameters of SEM algorithm in 2D case- for $\tau = 0.6$ and L_1 distances

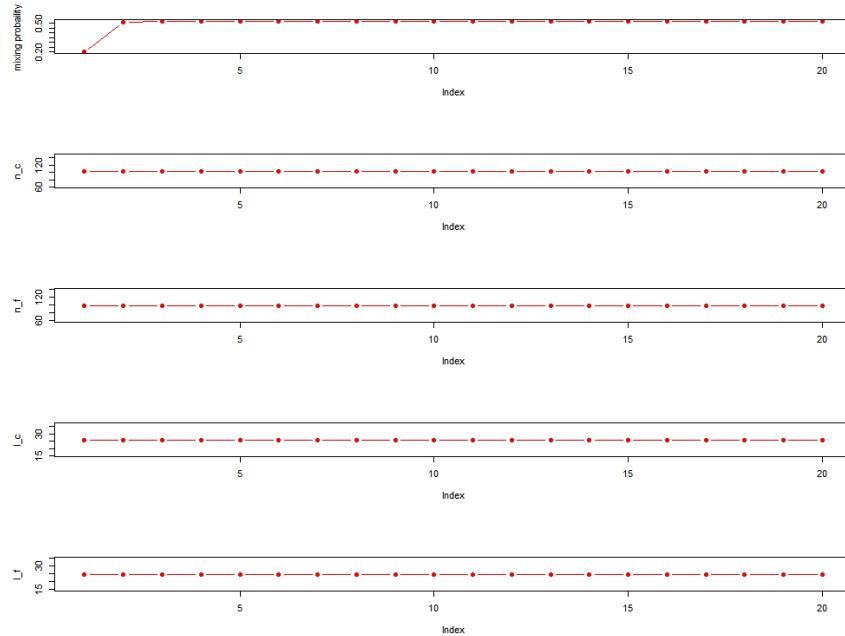


Figure 4.27: Convergence of parameters of SEM algorithm in 2D case- for $\tau = 0.2$ and L_2 distances

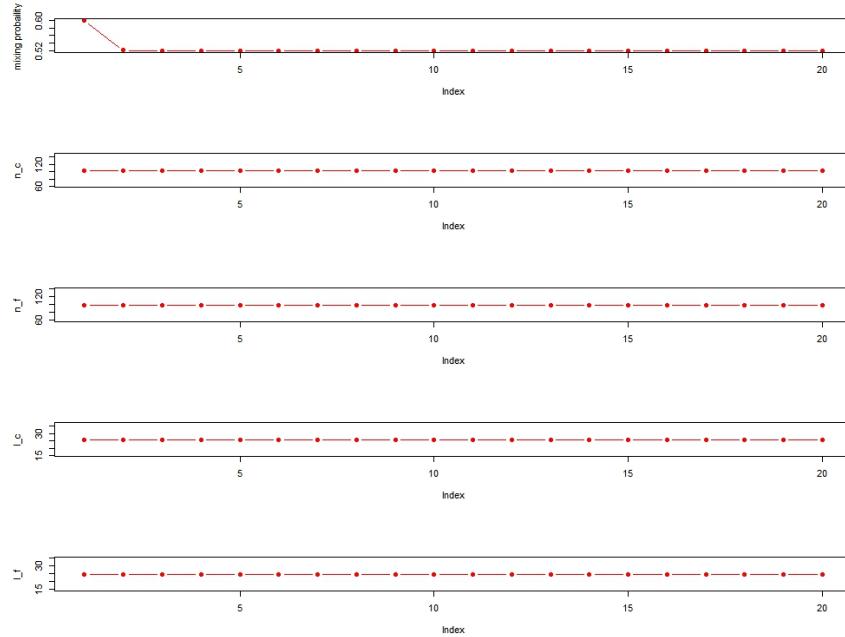


Figure 4.28: Convergence of parameters of SEM algorithm in 2D case- for $\tau = 0.6$ and L_2 distances

Table 4.4: Numerical analysis of 2D SEM

	True number of points		Estimated number of points				The accuracy	
	clutter	features	clutter		features			
			$c \mapsto c$	$f \mapsto c$	$f \mapsto f$	$c \mapsto f$		
2D	105	98	104	0	98	1	99.5%	

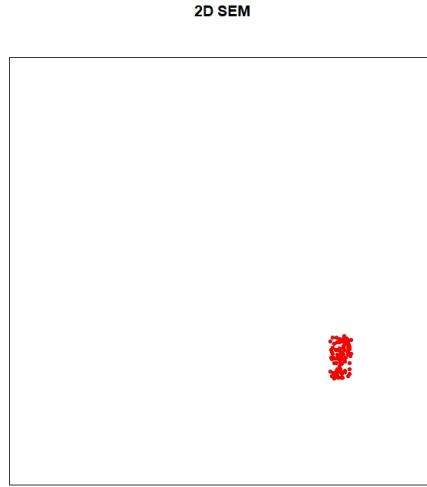


Figure 4.29: Feature detection from 2D SEM

In 3D SEM algorithm, we simulated a clutter with intensity 1.2 in the same region (3D box $[0, 5] \times [0, 5] \times [0, 10]$) of the original 3D simulated clustered process for the s-step. The simulated clutter has 303 points. A feature process is simulated with intensity 40 in an unit area (3D box $[3, 4] \times [2, 3] \times [5, 6]$). The feature process has 32 points. The Figure 4.30 shows simulated clutter, feature and feature-clutter mixture processes for the s-step of 3D SEM.

Simulated 3D clutter points for s-step in SEM Simulated 3D feature points for s-step in SEM Simulated 3D mixture process for s-step in SEM

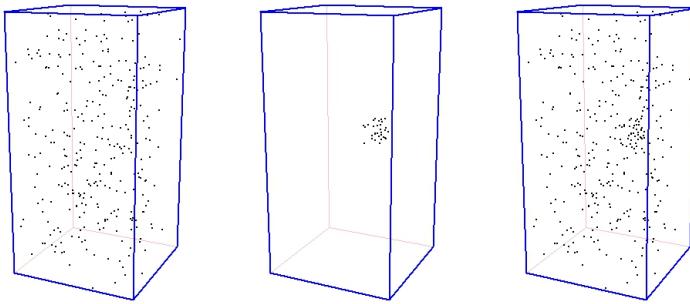


Figure 4.30: Feature detection from 2D SEM

L_1 and L_2 distances used in 3D SEM algorithm with two different initial mixing probabilities $\tau = 0.3$ and $\tau = 0.7$. The 3D SEM algorithm was run for 20 iterations.

The Figure 4.31 shows the values of the parameter set $(\tau, n_c, n_f, \lambda_c, \lambda_f)$ in each iteration for L_1 distance measure with initial $\tau = 0.3$. The convergence of parameters with L_1 distances with initial $\tau = 0.7$ shows in the Figure 4.32. The same plots for L_2 distances with initial $\tau = 0.3$ and $\tau = 0.7$ are shown in the Figure 4.33 and the Figure 4.34 respectively.

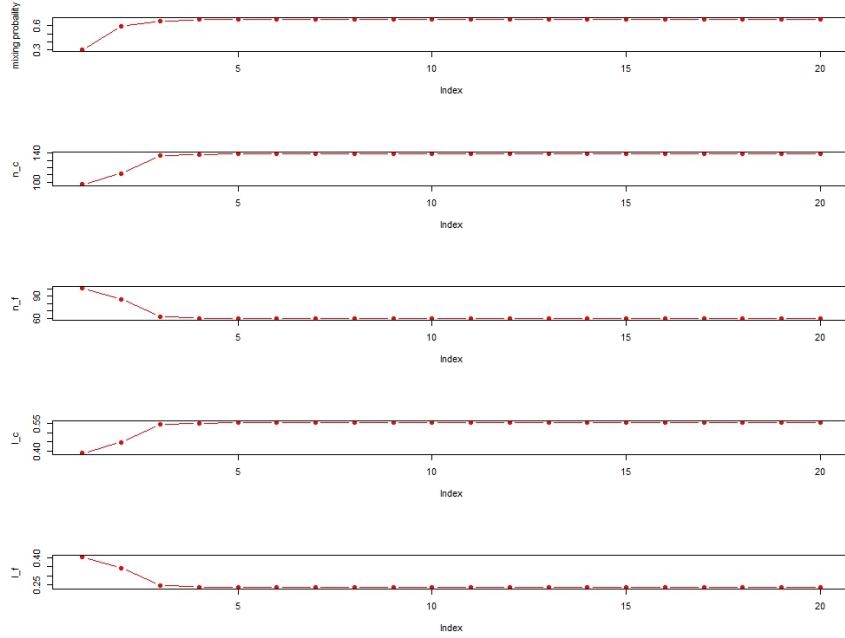


Figure 4.31: Convergence of parameters of SEM algorithm in 3D case- for $\tau = 0.3$ and L_1 distances

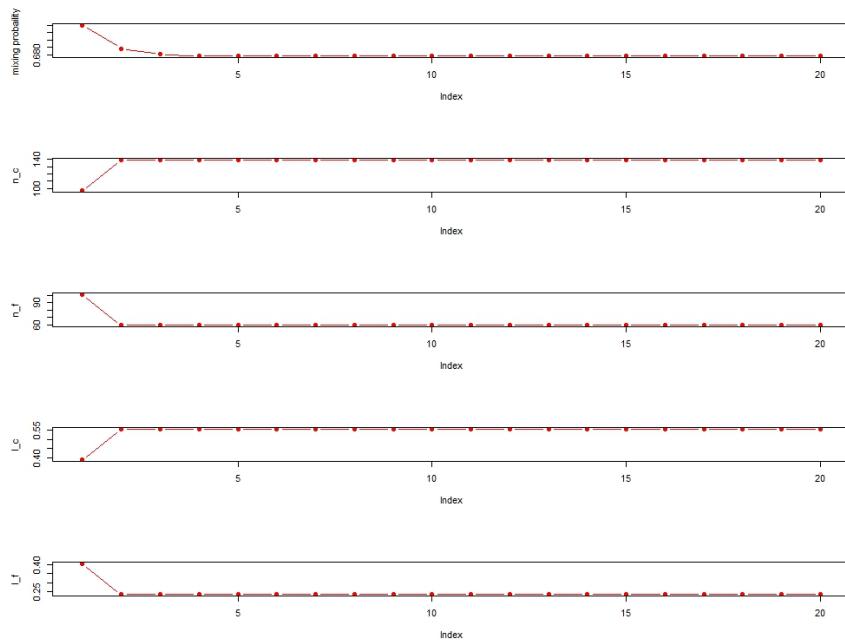


Figure 4.32: Convergence of parameters of SEM algorithm in 3D case- for $\tau = 0.7$ and L_1 distances

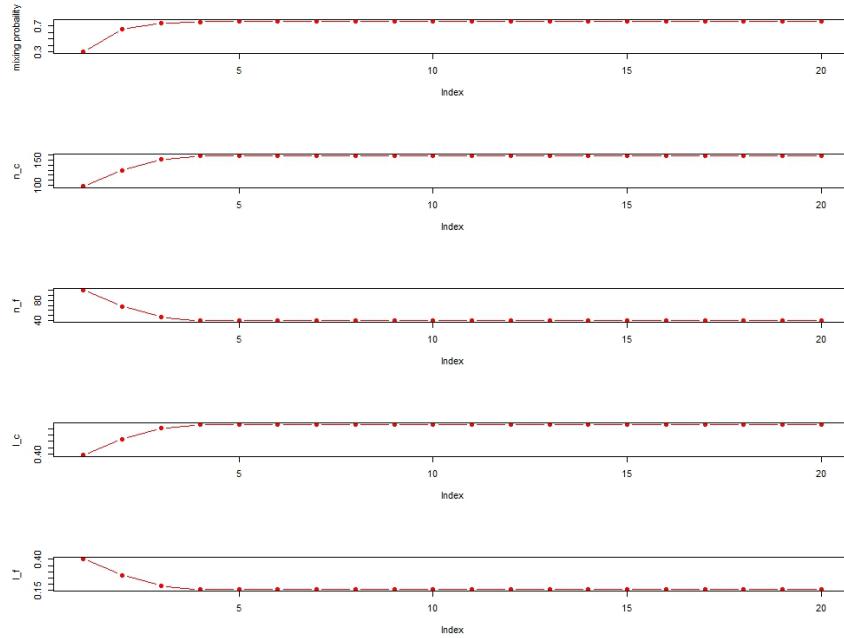


Figure 4.33: Convergence of parameters of SEM algorithm in 3D case- for $\tau = 0.3$ and L_2 distances

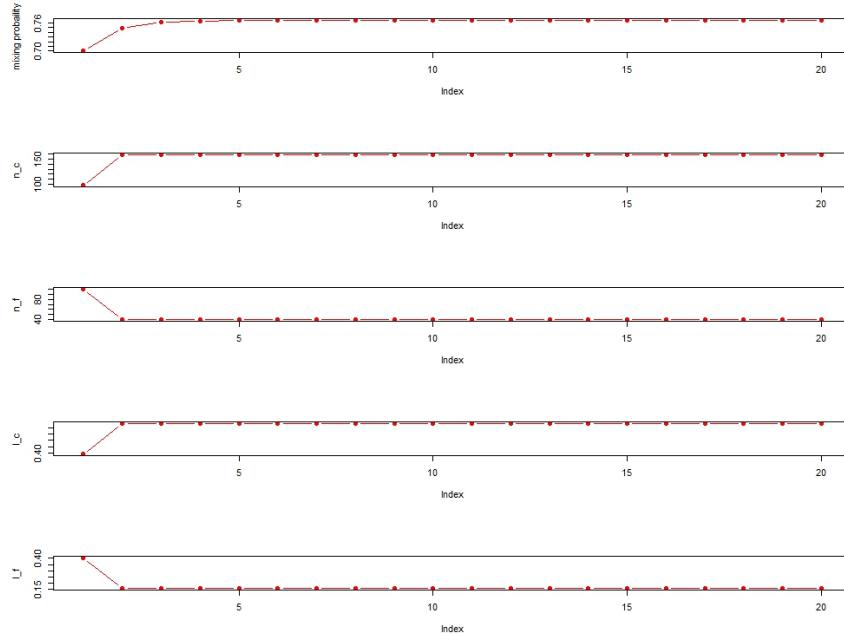


Figure 4.34: Convergence of parameters of SEM algorithm in 3D case- for $\tau = 0.7$ and L_2 distances

The results of feature-clutter identification for distances L_1 and L_2 are different but same for initial mixing probabilities $\tau = 0.3$ and $\tau = 0.7$. Then we can compare the results for L_1 and L_2 . The Figure 4.35 shows feature points extracted with L_1 and L_2 . The numerical results of SEM method for 3D clustered process is given by the Table 4.5.

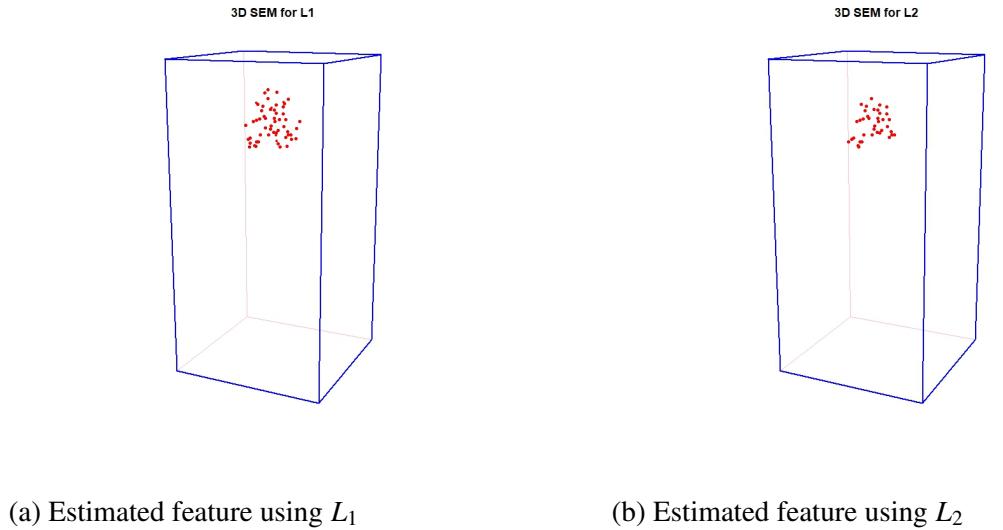


Figure 4.35: Estimated feature points of 3D simulated clustered process from SEM method

Table 4.5: Numerical analysis of 3D SEM

	True number of points		Estimated number of points				The accuracy	
	clutter	features	clutter		features			
			$c \mapsto c$	$f \mapsto c$	$f \mapsto f$	$c \mapsto f$		
L_1	92	106	87	52	54	5	71.21%	
L_2	92	106	88	71	35	4	62.12%	

Chapter five presents a comparison study of the results obtained from the methods that we have discussed in this chapter for both 2D and 3D point processes.

Chapter 5

Comparison of HCA, EM and SEM

In this chapter, first we compare the the results of the cluster methods explained in the chapter four.

Table 5.1: Summarized results of 2D clustering

		True value	HCA without covariance	NN distances & EM method			SEM method			
				L ₁			L ₂			
				k=1	k=2	k=5	τ = 0.2	τ = 0.6	τ = 0.2	τ = 0.6
2D	n_f	98	99	100	99	99	99	99	99	99
	$n_{f_{true}}$	-	98	98	98	98	98	98	98	98
	DR_f	-	100	100	100	100	100	100	100	100
	Accuracy	-	99.5	99.0	99.5	99.5	99.5	99.5	99.5	99.5

Table 5.2: Summarized results of 3D clustering

		True value	HCA without covariance	NN distances & EM method			SEM method			
				L ₁			L ₂			
				k=1	k=2	k=5	τ = 0.3	τ = 0.7	τ = 0.3	τ = 0.7
3D	n_f	106	66	125	122	114	59	59	39	39
	$n_{f_{true}}$	-	62	106	106	106	54	54	35	35
	DR_f	-	58.5	100	100	100	50.9	50.9	32.1	32.1
	Accuracy	-	75.75	90.4	91.19	95.95	71.2	71.2	62.1	62.1

The Table 5.1 gives the summary of numerical results of clustering of 2D simulated point process given by Example (4.2.1.b) and the numerical results of clustering of 3D simulated points process given by Example (4.2.2.a) are summarized in the Table 5.2.

n_f : number of feature points estimated from the method

$n_{f_{true}}$: number of estimated feature points which are come from true feature points

DT_f : rate of detection of true feature points into the estimated feature

Accuracy: correctly detection rate of both feature and clutter points

If we consider the 2D point process, the Table 5.1 shows that all three methods of clustering HCA (without covariance), EM method and SEM method work well on the 2D point process. All these methods were able to identify all the feature points correctly from the cluster process. The method HCA and SEM that we used LISA functions for clustering give similar results as EM algorithm. Then it is clear that even we do not the distribution of the LISA functions, we can use them well in point process clustering. The results in the Table 5.2 shows that the accuracy of the EM method is higher than the HCA and SEM methods. **nncclean** function of EM method for $k = 5$ identified less false feature points than $k = 1$ and $k = 2$ in 3D point process clustering.

In both 2D and 3D cases, we can see that the estimated results did not depend on the initial guess of mixing probability. L_1 measure could identify more feature points correctly than L_2 measure in 3D SEM clustering. Also in 3D case, HCA and SEM with L_1 have slightly similar accuracy but the correctly feature detection rate of HCA is a bit higher than the SEM method with L_1 .

We can use a well mixed feature-clutter point process with more than one feature for better comparison of the methods. For this, an another 2D cluster process was simulated with two clusters in $[0, 1] \times [0, 1]$ units window and it is named as X_2 . The point process X_2 has two clusters with 204 points and 325 clutter points. The Figure 5.1 shows the feature and clutter of the point process X_2 .

The dendrogram of the HCA of X_2 is given in the Figure 5.2. The estimated feature and clutter from the HCA method (without covariances) are shown in the Figure 5.3.

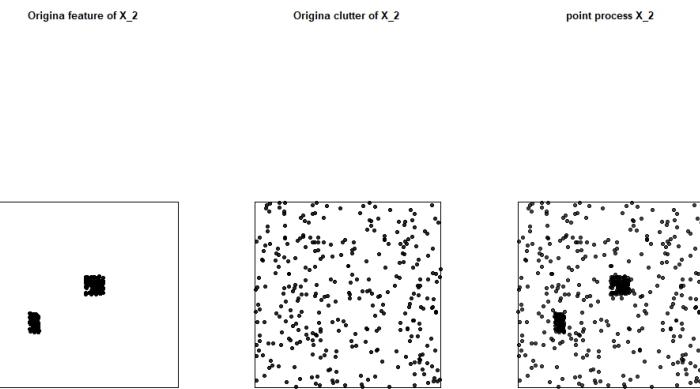


Figure 5.1: Simulated 2D feature-clutter process X_2

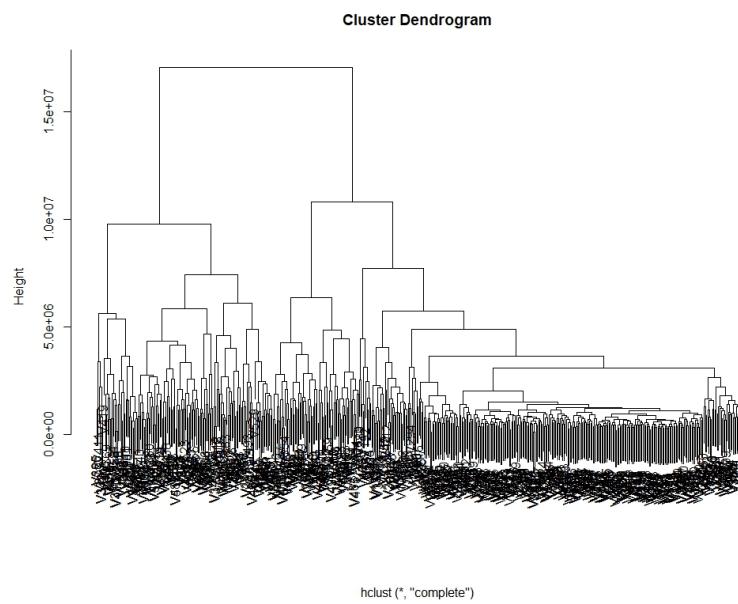


Figure 5.2: The dendrogram of X_2

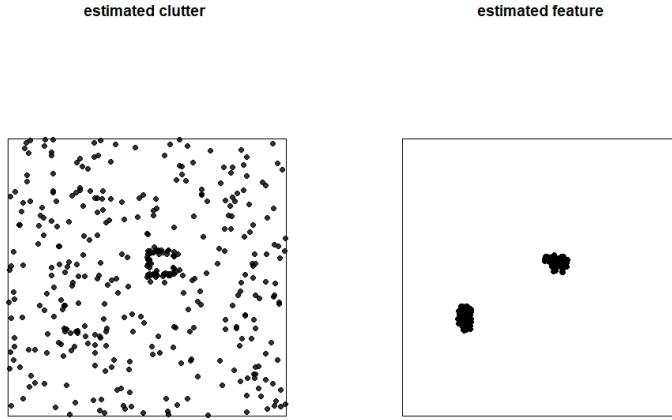


Figure 5.3: Estimated feature and clutter of X_2 using HCA without covariance

The clustering of NN distances of X_2 was done for 1st, 2nd and 5th nearest neighbour. The results of the EM method of X_2 are shown in Figure 5.4, Figure 5.5 and Figure 5.6 for $k = 1, k = 2$ and $k = 5$ respectively.

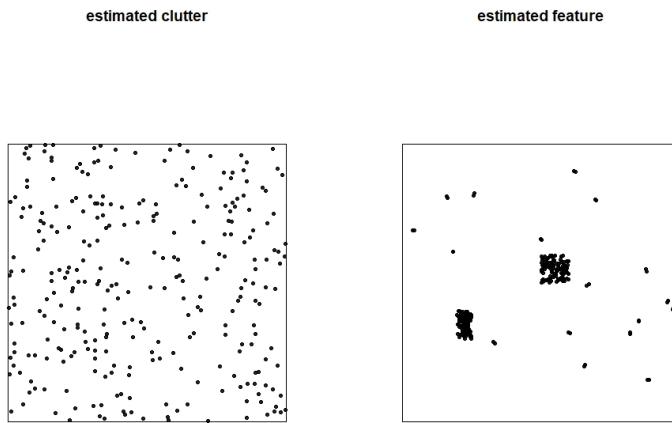


Figure 5.4: Estimated feature and clutter of X_2 using EM method with $k = 1$

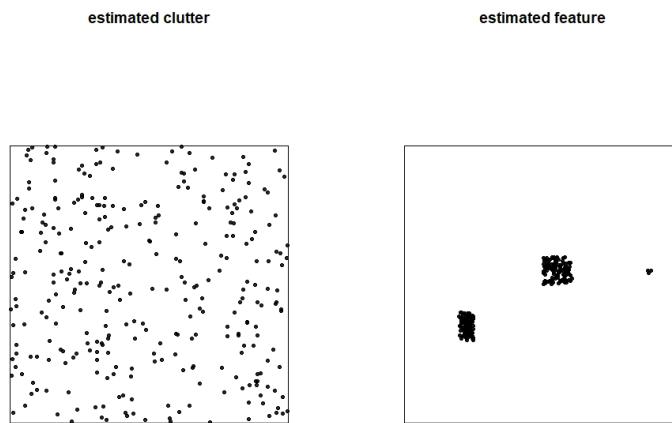


Figure 5.5: Estimated feature and clutter of X_2 using EM method with $k = 2$

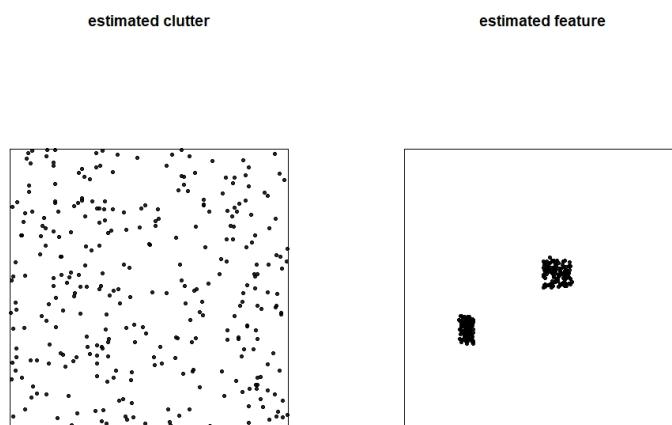


Figure 5.6: Estimated feature and clutter of X_2 using EM method with $k = 5$

We simulated a clutter with 217 points in $[0, 1] \times [0, 1]$ units window and a feature with 289 points in $[0.2, 0.3] \times [0.8, 0.9]$ units window for the s-step of the SEM method. The simulated feature-clutter point process for the s-step is shown in the Figure 5.7. The SEM method was done for the X_2 point process using initial mixing probability $\tau = 0.3$ for both L_1 and L_2 measures..

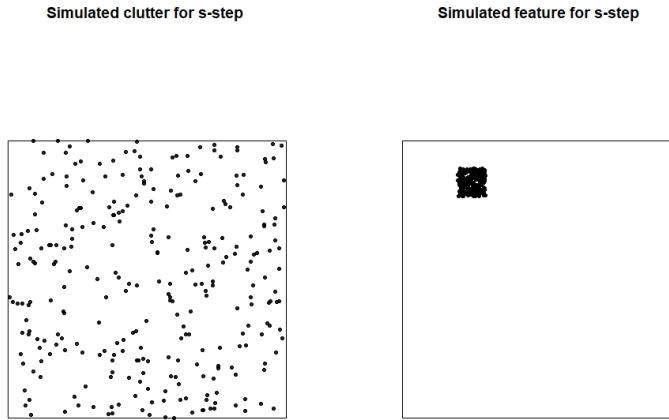


Figure 5.7: Simulated point process for the s-step

The estimated features and clutters from the SEM method for both L_1 and L_2 case are shown in the Figure 5.8 and the Figure 5.9.

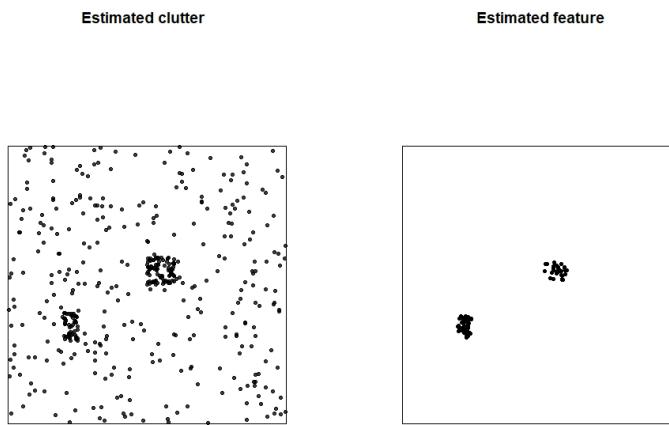


Figure 5.8: SEM method for L_1

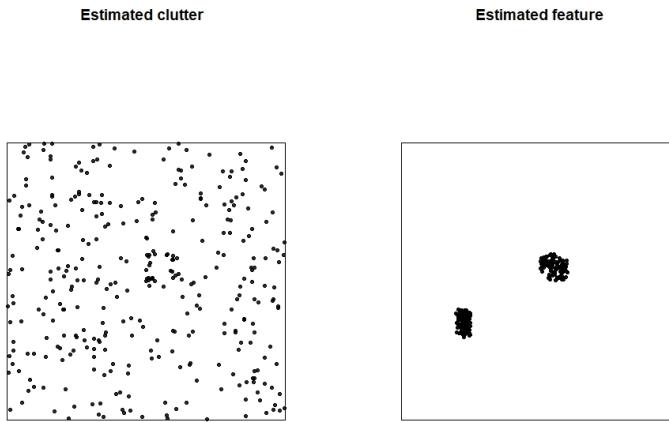


Figure 5.9: SEM method for L_2

The results of clustering X_2 using HCA method,EM method and SEM method are summarized in the Table 5.3.

We continued the methods for a simulated point process. A 3D point process simulated in $[0, 1] \times [0, 1] \times [0, 1]$ units window and named as X_3 . Point process X_3 has 948 clutter points and 199 feature points. The Figure 5.10 shows simulated feature and clutter of X_3 .

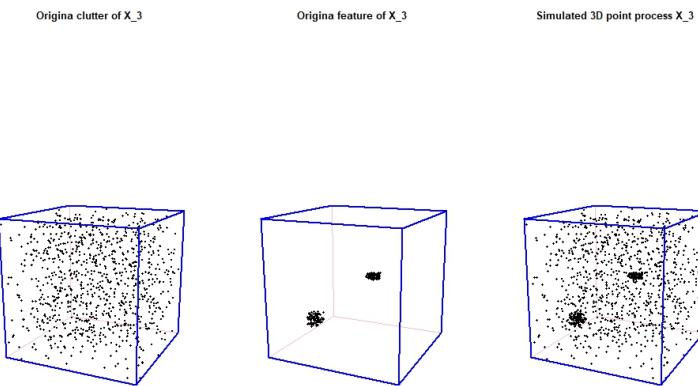


Figure 5.10: Simulated 3D point process X_3

The dendrogram of the HCA of X_3 is given in the Figure 5.11. If we cut the dendrogram into the two groups then the estimated feature and clutter are shown in the Figure 5.12. In this case we can see that more feature points are fallen to the clutter. To avoid this we can clean up more feature points by cutting the dendrogram into many groups. If we cut the dendrogram into 3 groups then it gives the clutter and two clusters. We added points of the two clusters into one group as feature points. The results of the HCA with $k = 3$ are shown in the Figure 5.13.

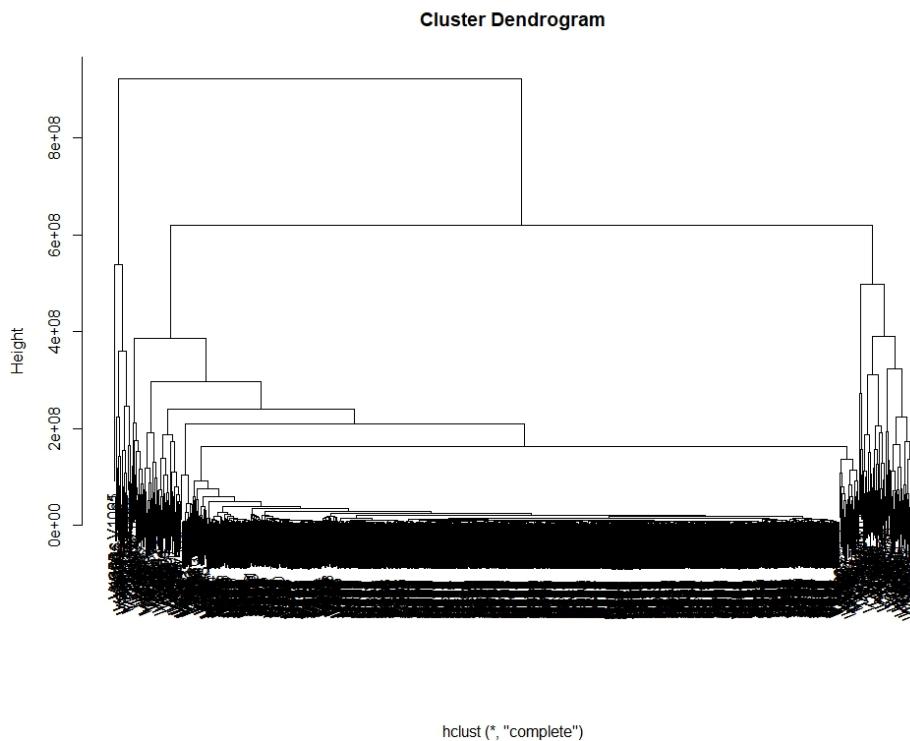


Figure 5.11: The dendrogram of HCA of X_3

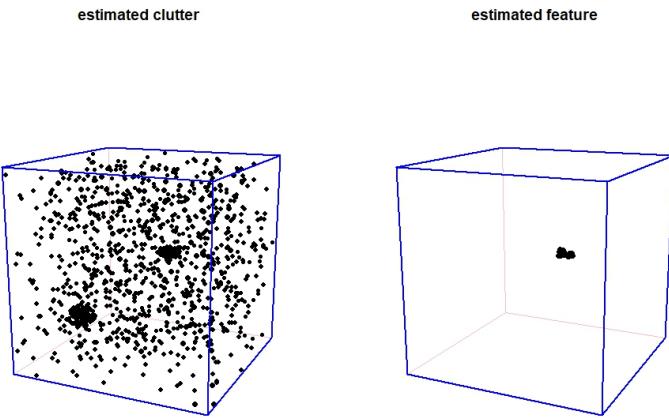


Figure 5.12: HCA of X_3 with $k = 2$

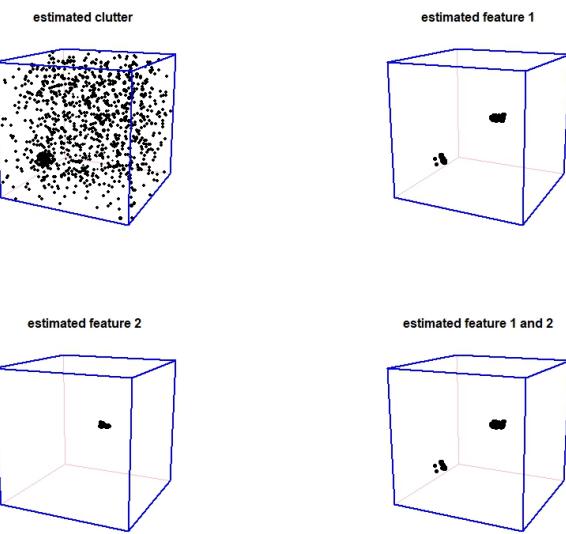


Figure 5.13: HCA of X_3 with $k = 3$

We preformed EM method for nearest neighbour distances of X_3 for different degrees of neighbour. The estimated features and clutters for $k=1, k=2$ and $k=5$ are given in the Figure 5.14, Figure 5.15 and Figure 5.16 respectively.

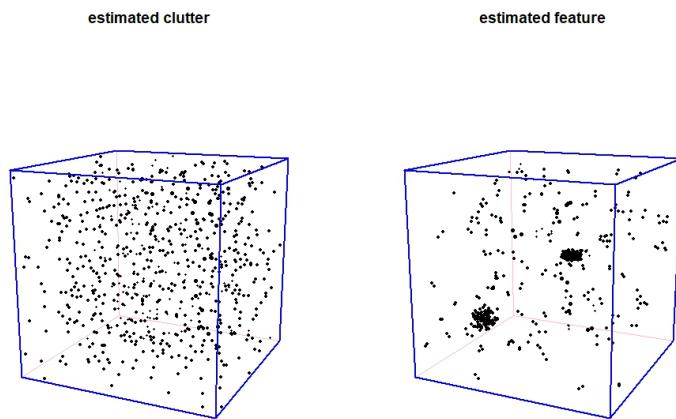


Figure 5.14: EM method of X_3 with $k = 1$

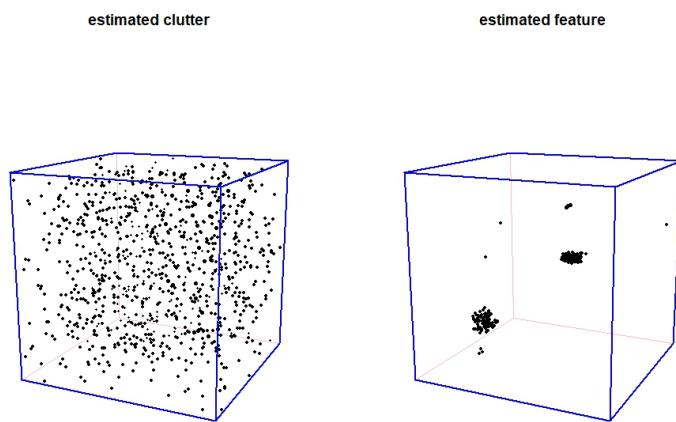


Figure 5.15: EM method of X_3 with $k = 2$

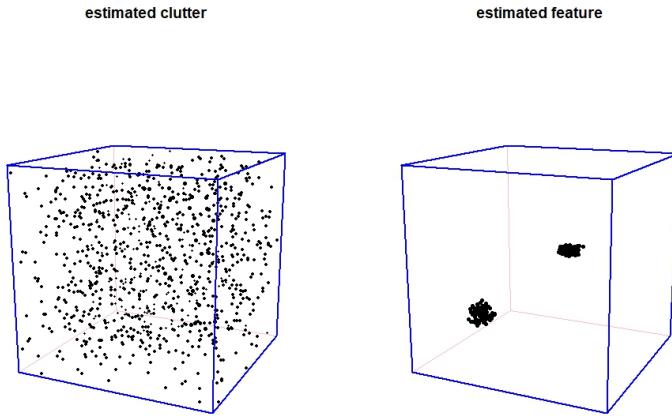


Figure 5.16: EM method of X_3 with $k = 5$

A clutter of 742 points in $[0, 1] \times [0, 1] \times [0, 1]$ units window and a feature of 257 points in $[0.2, 0.3] \times [0.3, 0.4] \times [0.2, 0.3]$ units window were simulated in the s-step of the SEM method. The simulated feature and clutter are shown in the Figure 5.17.

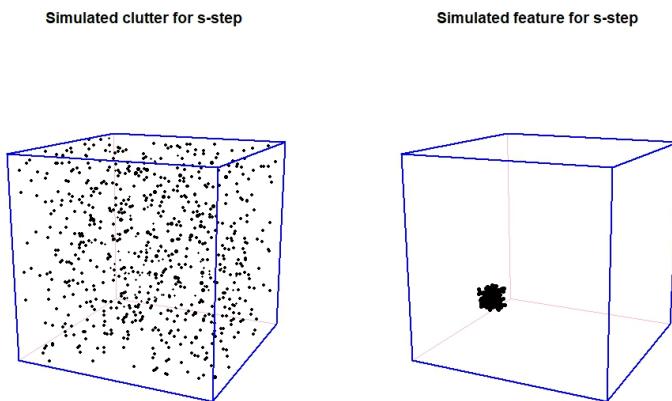


Figure 5.17: Simulated feature and clutter for the s-step in the SEM method of X_3

The SEM method of X_3 was done for L_1 distance measure by giving the initial mixing probability $\tau = 0.3$. The estimated feature and clutter of X_3 from the SEM method using L_1

are shown in the Figure 5.18.

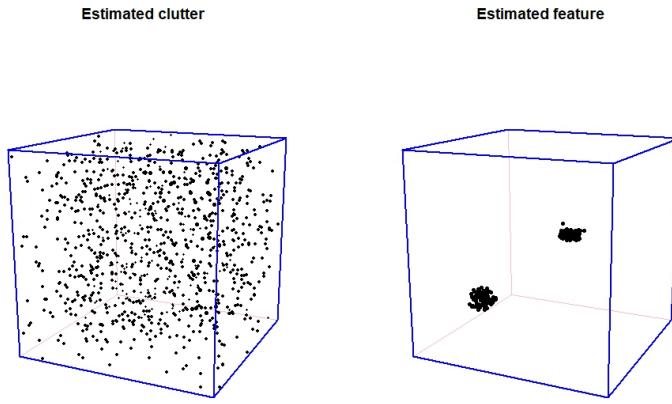


Figure 5.18: Estimated feature and clutter of X_3 from SEM method with L_1

The summarized results of cluster methods of X_3 are given in the Table 5.4.

Table 5.3: Summarized results of clustering of X_2

		True value	HCA without covariance	NN distances & EM method			SEM method	
				k=1	k=2	k=5	L_1	L_2
X_2	n_f	204	145	243	217	216	77	185
	$n_{f_{true}}$	-	138	200	203	204	72	178
	DR_f	-	67.65	98	99.5	100	37.7	87.3
	Accuracy	-	86.2	91.1	97.2	97.7	74.1	93.8

Table 5.4: Summarized results of clustering of X_3

		True value	HCA without covariance		NN distances & EM method			SEM method L_1
			k=2	k=3	k=1	k=2	k=5	
X_3	n_f	199	25	103	448	210	203	203
	$n_{f_{true}}$	-	25	102	199	195	197	196
	DR_f	-	12.6	51.3	100	97.9	98.9	98.5
	Accuracy	-	84.8	91.5	78.3	98.3	99.3	99.1

Summarized numerical results of 2D clustering of X_2 shows that clustering of product density LISA functions using the SEM method with L_2 measure identified more feature points of X_2 than the both HCA and SEM methods with L_1 measure. Even the values of 2D clustering given in the Table 5.1 show that all three method give similar results and estimate the features with higher detection rate, the results of X_2 show how the methods work differently when there are more than one feature and when the features are well mixed with the clutter. The EM method of NN distances was be done by changing the degree of neighbour to clear the features and for k=5, the EM method identified the all feature points of X_2 which are coming from two clusters.

In 3D clustering of the point process X_3 , the HCA method has the lowest feature detection rate which is 12.6%. If we consider the EM method, for k=1,it identified the all the feature points of the two clusters of the X_3 but the accuracy is lower than the other two cases of the method for k=2 and k=5 because of the large false identifications of the clutter points. The EM method for k=5 obtained good feature detection rate (98.9%) as well as good accuracy (99.3%). But the SEM shows that we can use product density LISA functions to get similar good results by obtaining 98.5 % feature detection rate and 99.1 % accuracy with L_1 distances.

Chapter 6

Conclusions and Final remarks

We have studied the methods of clustering for applying to spatial point pattern data. Nearest neighbour distance and product density LISA function are the statistic used in cluster methods. The methods were treated using simulated point processes.

Cluster Point Process Simulation

- Select the window size and intensity (or number of points) for clutter.
- Generate a random Poisson point process with the selected intensity and window size for the locations of the clutter.
- Select number of clusters, their intensities and shapes. The regions of the shapes should be inside the region of the clutter.
- Generate separate random Poisson point processes for each cluster.
- Fit all locations of clutter and features inside the selected window for the clutter.

The available functions of **spatstat** package in **R** were used to simulate the 2D and 3D point processes.

The EM cluster method of nearest neighbour distances was studied for applying 2D and 3D cluster point processes. This method can be done in **R** using **nnclean** function. The EM method of NN estimated feature points with high accuracy and high detection rate. It is clear

that we can use **nnclean** function for better feature-clutter classification of 2D and 3D clustered point processes.

Our goal was to check whether the product density LISA functions of point process could be used for better cluster analysis as same as **nnclean**. The research studies have been carried out for calculating 2D product density LISA functions in literature of point process statistics. Mathematical background of the product density LISA functions of 2D point process was well studied to build the formula for the product density LISA function of a 3D point process. Hence from this study we formulated the mathematical equation for the product density LISA function of 3D point process.

Hierarchical cluster analysis method and Stochastic EM method were used to analysis product density LISA functions. We can consider product density LISA functions with and without covariances in HCA. When we applied the HCA method with covariances for our simulated point processes, the unexpected results came out. As explained in the section 4.3.1.b, we use simulated uniformly distributed point process for Monte Carlo integration to approximate the variance and covariance values of LISA functions. This simulated point process for Monte Carlo integration plays a major role in the method and it might cause for the unexpected behaviour of the results. We can do the method by simulating different point processes for MC integration but it takes a lot of time since the running of the method is very large. During our limited study time, we were unable to treat the method by simulating different point processes for the MC integration part. But in future studies, the method can be developed to reduce the running time. Then the method can be easily run till it gives nice classification by simulating different point processes for MC integration part. Also we can try the method using a mixture of uniformly distributed feature and clutter point processes for MC integration part.

From the results shown in the Tables of the chapter five, we can see that the feature detection rates of the HCA of product density LISA functions without covariances for 3D point processes take lower percentage values than the 2D point processes. Then we can conclude that the HCA of product density LISA functions without covariances is not a very good applicable method for 3D point process clustering.

The second cluster method for the product density LISA functions that we discussed in our study is stochastic EM method. The results of the SEM method of 2D simulated point

processes show that the both L_1 and L_2 measures can be used in single cluster process to extract the feature points properly. But when the point process has more than one cluster then we can detect more feature points using L_2 measure than using L_1 measure. In the case of 3D point process, L_1 measure is an acceptable solution for both single and multiple cluster process analysing. Specially it shows that the detection rate and the accuracy of the SEM method are approximately same to the values of the EM method. The most advantage of the SEM method is its proper functioning without knowing the distribution function of product density LISA functions. In general, the SEM method can be brought forward as a good cluster method for point process feature extraction using product density LISA functions.

In the SEM method, kernel density estimation is used to estimate the densities of simulated feature and clutter at the s-step. In our study we used Gaussian kernel function. But in future studies, different kernel functions can be used to check how the accuracy of the method is changed with kernel functions. Also different interpolation methods can be used to estimated the density values of d_i . Furthermore, we can use other distance measures as we used L_1 and L_2 to measure dissimilarities. Therefore the SEM method is a very interesting method for clustering of product density LISA functions in point process statistics and we can continue the SEM method for future research studies to bring out more fascinating results.

Bibliography

- [1] California redwoods point pattern.
- [2] Chorley-ribble cancer data in spatstat. *Chorley and South Ribble Health Authority of Lancashire (England) between 1974 and 1983..*
- [3] Gorilla nesting sites data in spatstat. *Wildlife Conservation Society Takamanda-Mone Landscape Project (WCS-TMLP)..*
- [4] Hierarchical clustering. https://en.wikipedia.org/wiki/Hierarchical_clustering Wikipedia contributors.
- [5] Kernel density estimation. https://en.wikipedia.org/wiki/Kernel_density_estimation Wikipedia contributors.
- [6] Linear interpolation. https://en.wikipedia.org/wiki/Linear_interpolation Wikipedia contributors.
- [7] nnclean function. <https://www.rdocumentation.org/packages/spatstat/RDocumentation>.
- [8] Norm. [https://en.wikipedia.org/wiki/Norm_\(mathematics\)](https://en.wikipedia.org/wiki/Norm_(mathematics)) Wikipedia contributors.
- [9] Probability space. https://en.wikipedia.org/wiki/Probability_space Wikipedia contributors.
- [10] A.BADDELEY, E.RUBUK, AND R.TURNER. Spatial point patterns methodology and applications with r. *Chapman and Hall/CRC, Interdisciplinary Statistics Series 13:978-1-4822-1021-7(eBook-PDF)* (2016).
- [11] A.L.PATTERSON. A fourier series method for the determination of the components of interatomic distances in crystals.

- [12] J.ILLIAN, A.PENTTIINEN, H.STOYAN, AND D.STOYAN. Statistical analysis and modelling of spatial point patterns. *John Wiley and Sons,Ltd Page 76* (2008).
- [13] J.MATEU, G.LORENZO, AND E.PORCU. Detecting features in spatial point processes with clutter via local indicators of spatial association. *American Statistical Association, Institute of Mathematical Statistics and Interface Foundation of North America Journal of Computational and Graphical Statistics, Volume 16,Number 4,Pages 968-990* (2007).
- [14] N.CRESSIE, AND COLLINS, L. Analysis of spatial point patterns using bundles of product density lisa functions. *American Statistical Association and the International Biometric Society Journal of Agricultural,Biological and Environmental Statistics, Volume 6,Number 1,Pages 118-135* (2001).
- [15] WIKIPEDIA. Expectation maximization algorithm.

Appendix A

A.1 List of Acronyms and Abbreviations

2D Two dimensional

3D Three dimensional

CSR Complete spatial random

DR Detection rate

EM Expectation maximization

HCA Hierarchical cluster analysis

LISA Local Indicators of Spatial Association

MC Monte Carlo

MLE maximum likelihood estimate

NN Nearest neighbour

SEM Stochastic expectation maximization

Appendix B

B.1 List of Terms

W Observation window of a point process

X A point process

v Volume of a 3D point process

x A point pattern (a realisation from a point process)

ϵ Bandwidth

Σ Covariance matrix

σ Variance

λ The intensity of a point process

δ Classification parameter

τ Mixing probability

$\rho(r)$ product density function

e_{ij} Translation edge correction weights between points x_i and x_j

f_ϵ Epanechnikov kernel function

$|W|$ The area of the observation window

$\|x_i - x_j\|$ Euclidean distance between points x_i and x_j

K(r) K-function

a area of the region of a 2D point process

cov Covariance

g(r) Pair correlation function

n The number of points of a point process

n(x) The number of points of the point pattern **x**

r Distance

var Variance