

Optimum Average Silhouette Width Clustering Methods

Fatima Batool

Department of Statistical Science
University College London



*A dissertation submitted for the degree of
Doctor of Philosophy*

Declaration

This dissertation entitled “Optimum Average Silhouette Width Clustering Methods”, submitted by me is my own composition and has not been submitted to any other university for any other degree. All the materials and findings are original and include nothing which is outcome of a work done previously or in collaboration except otherwise indicated through reference. The thesis has been formatted according to University College London guidelines.

The work in this dissertation has been compiled as the scientific articles, a list to which is given below.

The work in Chapter 3 and Appendix B is presented for publication in the following article:

- Batool, F., & Hennig, C. (2019). *A new hierarchical clustering algorithm based on optimization of ASW linkage criterion*, under review.

Most of the materials in Chapters 4, 5, Appendix C, and D are part of the following three articles:

- Batool, F., & Hennig, C. (2019). *Clustering by optimizing average silhouette width*, under review.
- Batool, F., & Hennig, C. (2019). *Characterization of average silhouette width clustering*, under review.
- Batool, F. (2019). ASW index and OASW clustering where to use and not to use, to be submitted shortly for peer reviewing after the submission of the dissertation.

Finally, an R package source code named *OASW clustering* comprised of all the proposed algorithms in this work is available upon request from author.

Acknowledgement

I feel profound gratitude while expressing my sincere thanks to my supervisor Christian Hennig for his guidance and support throughout my PhD. I am immensely grateful to him for offering critical feedbacks on my ideas, and thorough discussions. I am grateful to him for improving my conception of cluster analysis through his unique perspective and for his patience and willingness to explain overwhelming things more than once. I am also indebted to him for reading the final draft of the thesis and improving it linguistically.

I would like to acknowledge the discussions with Margareta Ackerman on theoretical clustering, which improved my understanding in this area of research. I benefited from her guidance in this area. I also gratefully acknowledge Catalina Vallejos for directing me towards scRNA-seq data analysis workflow. I am also grateful to Philippe Naveau for providing the software to reproduce their results, both software and results were used in this work. I especially thank to Codina Cotar for giving her munificent time and advise.

I am also thankful to Ioanna Manolopoulou for kindly serving as my subsidiary supervisor. I am grateful to Ricardo Silva for kindly serving as the upgrade examiner. I would like to thank graduate research tutor Paul Northrop for his help with official proceedings of PhD. My warm gratitude goes to Matina Rassias for providing teaching related support.

I am thankful to the Commonwealth Commission for funding my PhD studies without which this work would not have been possible. I'm grateful to my programme officers' Juliette Hargreave and James Goldsmith at CSC for taking care of all funding affairs and providing timely access to research grants every year. I am also grateful to the department of Statistical Science for offering me a Teaching Assistantship near the end of my PhD which has enhanced my statistical understanding further. I want to acknowledge the excellent resources for teaching and research at the department of Statistical Science and University College. I am also thankful to Deepti Jayawardena for providing an excellent administrative support during my PhD.

I would like to thank all the members of the department of Statistical Science at University College and The Alan Turing Institute for making these places delightful and rewarding to work. I feel lucky to be a part of these excellent stimulating intellectual environments. Graduate school has been joyful and tremendous learning experience for me. A special gratitude goes out to all those who made it so.

I am thankful to my brother for reading an earlier draft of thesis and for upgrading the language. I have always immensely profited from the love, joy and support of my parents and siblings. I express my profound gratitude to my parents' moral support and compassionate behaviour which has always been a continuous source of comfort and strength for me.

Optimum Average Silhouette Width Clustering Methods

Fatima Batool

Abstract

Cluster analysis is the search for groups of alike instances in the data. The two major problems in cluster analysis are: how many clusters are present in the data? And how can the actual clustering solution be found? We have developed a unified approach to estimate number of clusters and clustering solution mutually. This work is about theory, methodology and algorithm developed of newly proposed approach.

Average silhouette width (ASW) is a well-known index for measuring the clustering quality and for the estimation of the number of clusters. The index is in wide use across disciplines as standard practice for these tasks. In this work the clustering methodologies is proposed that can itself estimate number of clusters on the fly, as well as produce the clustering against this estimated number by optimizing the ASW index. The performance of the ASW index for these two tasks are meticulously investigated.

ASW based clustering functions are proposed for the two most popular clustering domains i.e., hierarchical and non-hierarchical. The performance comparison for clustering solutions obtained from the proposed methods with a range of clustering methods has been done for the quality evaluation.

The performance comparison for the estimation of the number of clusters of the proposed methods has been made using a wide spectrum of cluster estimation indices and methods. For this, large scale studies for the estimation of the number of clusters have been conducted with well-reputed clustering methods to find out each method's estimation performance with different indices/methods for various kinds of clustering structures.

Developing mathematical and theoretical aspects for clustering is a relatively new and challenging avenue. Recently this research domain has received considerable attention due to the present need and importance of theory of clustering. The purpose behind the theory development for clustering is to make the general nature of clustering more understandable without assuming particular data generating structures and independently from any clustering algorithm/functions. Lastly, a considerable amount of attention has been drawn towards the theory development of the ASW index in the latter part of the thesis.

Impact Statement

Cluster analysis is a field where problems posed are very diverse and challenging in one way or the other. Although there is a vast majority of clustering methods present already, the best solutions to these problems have yet to come. While there are a few clustering methods that can estimate the number of clusters themselves, a majority of the present clustering methods usually deal with these two problems separately. The researchers are trying to find more satisfactory answers to the major underlying questions in the field and are coming up with the solutions that can cover vast majority of applications. The proposed methods in this work also seek answers to the fundamental questions in the field. The new methods will have impact in at least three ways:

Simplicity The methods proposed in this work can be seen as a unified approach to clustering with an aim of making their use straightforward. The users don't have to provide the number of clusters beforehand or choose other parameters. Only a data set is required as an input to produce the clustering and optimal number of clusters. The algorithms proposed here work with distances between observations, which has made them applicable to data application from any discipline, provided that the distance calculation is possible.

Wide applicability The two types of input commonly available for clustering are the original data values or the distances between the data points. The algorithms are implemented in a way that they work with both types of inputs. Several distance measures are provided to match various data requirements. This enables the proposed algorithms to work for almost all kinds of clustering applications.

In particular we have considered novel data applications from single cell RNA sequencing data clustering problems, and clustering of weather stations based on rainfall data. The proposed algorithms have performed better or on par with the competitors, due to their extra advantages of simplicity, less computational time in some instances, and capability of estimation of number of clusters.

Better clustering quality The proposed algorithms have always improved the quality of clustering for all the data settings included in the studies as compared to the ASW index. Therefore, the use of the proposed methods will always guarantee improved clustering quality.

Contents

1	Introduction	1
1.1	Basic concepts related to validation indices	2
1.2	Definition of a cluster:	3
1.3	Objectives and contributions of the thesis	7
1.4	Outline of the thesis	9
2	Clustering Overview	11
2.1	Some clustering applications	11
2.2	Preliminary definitions and concepts	13
2.2.1	Types and format of data	14
2.2.2	Data clustering/partitioning	15
2.2.3	Crisp versus fuzzy clustering	16
2.2.4	Partitional versus hierarchical clustering methods	18
2.2.5	Deterministic vs. stochastic clustering methods	18
2.3	Challenges in cluster analysis	18
2.3.1	Selection of the proximity measure	19
2.3.2	Estimation of the optimal number of clusters	19
2.3.3	Validation issues	20
2.4	Clustering methods review	21
2.4.1	Hierarchical clustering methods	22
2.4.2	Partitioning clustering methods	28
2.4.3	Model-based clustering methods	33
2.4.4	Spectral clustering methods	35
2.5	Definition of validation indices	38
2.5.1	Methods for estimation of the number of clusters	39
2.6	The average silhouette width	48
2.7	The PAMSIL algorithm	49
2.8	Clustering comparison measures	50
2.8.1	Adjusted rand index	50

3 The Optimum ASW Based Linkage Criterion	53
3.1 Preliminary notations	53
3.2 HOSil algorithm and description	54
3.2.1 Algorithm's description	54
3.2.2 Some notes on implementation	62
3.3 Characteristics of interest for clustering	64
3.4 Definition of data generating processes	65
3.5 Simulation design	71
3.5.1 Discovering the true clustering	73
3.5.2 Performance for the estimation of k	81
3.6 Further exploration	85
3.7 Complexity Analysis	89
3.7.1 Runtime complexity	90
3.8 A faster approximation	90
3.9 Applications	93
3.9.1 Tetragonula bee data clustering	93
3.9.2 French rainfall data clustering	96
3.10 Closing remarks	98
4 The Optimum ASW Partitioning Clustering Method	100
4.1 Background and preliminary notations	100
4.2 OASW clustering	101
4.3 Implementation of the OASW method	103
4.4 Simulation setup	104
4.4.1 Definitions of data generating processes	105
4.5 Simulation I: Fix k case	109
4.5.1 Results discussion	110
4.5.2 Summary	122
4.6 Simulation II: Estimation of k case	127
4.6.1 ASW Results	129
4.6.2 Comparison with other indices	130
4.6.3 Summary	146
4.6.4 Some general comments	154
4.7 Simulation III: Overlapping data structure	155
4.7.1 Results discussion	159
4.8 Runtime complexity	166
4.9 Best OASW algorithm selection	167
4.10 Fast version	171
4.10.1 FOSil algorithm	171
4.11 OSil and FOSil comparison	180
4.12 Closing remarks for simulations	183

4.13 OSil ₁ complexity	185
4.14 FOSil ₂ complexity	186
4.15 OASW clustering: Compact and well-separated clusters	187
4.16 Distance metric comparison	191
4.16.1 Simulation scenario	192
4.16.2 Results	192
4.17 Applications	194
4.17.1 Tetragonula bee's data revisited	194
4.17.2 France rainfall data revisited	195
4.17.3 Genetics background	198
4.17.4 Introduction to scRNA-seq technique	199
4.17.5 Identification of cell population	200
4.17.6 scRNA-seq data analysis workflow	200
4.17.7 Clustering scRNA-seq	201
5 Theoretical foundation	210
5.1 Background discussion	210
5.2 Existing literature	212
5.3 Preliminaries	214
5.4 Characterization of the ASW	218
6 Future aspects	228
6.1 An alternative HOSil algorithm suggestion	229
6.2 OSil further improvement and extensions	229
6.3 Future theory development	232
Appendix A Statistical distributions for data generation	235
Appendix B HOSil Algorithm results	239
B.1 HOSil clustering visualization	240
B.2 HOSil estimation of number of clusters	261
Appendix C OSil simulations results	288
C.1 Simulation I: Known k case	289
C.2 Simulation II: Estimation of k Case	299
Appendix D Numerical example for Richness proof for ASW	335
Bibliography	338

List of Notations

Notation	Description
\mathbb{N}_n	Set of natural numbers of size n excluding 0. $\mathbb{N}_n = \{1, \dots, n\}$
\mathbb{N}_k	Set of natural numbers of size k excluding 0. $\mathbb{N}_k = \{1, \dots, k\}$
\mathbb{R}^+	Set of positive real numbers
$\ \cdot\ ^1$	Manhattan norm or l_1 -distance
$\ \cdot\ ^2$	Euclidean norm or l_2 -distance
$\ \cdot\ ^q$	Minkowski distance or l_q -distance
$ W $	determinant of a matrix W
$tr(W)$	trace of matrix W
I_p	Identity matrix of order p
\emptyset	Empty Set
W^t	transpose of matrix W
k	Number of clusters
K	Maximum number of clusters used for estimation
p	Number of dimensions
n	Number of observations
\mathcal{X}	Set of n values of p observations
\mathcal{C}_k	clustering of size k : $\mathcal{C}_k = \{C_1, \dots, C_k\}$
$x_i \sim_{\mathcal{C}} x_j$	Observations $x_i, x_j \in \mathcal{X}$ in same cluster of a clustering \mathcal{C}
$\eta \cdot d$	dot product of scalar η with a function d
\max	maximum value of a function or a vector (in whichever context used)
\min	minimum value of a function or a vector (in whichever context used)
$\arg \max$	function maximized over all the arguments of a domain
$\arg \min$	function minimized over all the arguments of a domain

List of Abbreviations

Abbreviation	Full Name
SW	Silhouette Width
ASW	Average Silhouette Width
OASW	Optimum Average Silhouette Width clustering
HOSil	Hierarchical Optimum average Silhouette width algorithm
OSil	Optimum average silhouette width algorithm
FOSil	Fast OSil algorithm
PAM	partitioning around medoids algorithm
BIC	Bayesian Information Criterion
mb	Model-based clustering using BIC
EM	Expectation–maximization algorithm
DGP	Data Generating Process
PPR	Percentage Performance Rate
CH	Caliński and Harabasz (1974) index
H	Hartigan (1975) index
Krzanowski and Lai (KL)	Krzanowski and Lai (1988) index
Gap	Tibshirani et al. (2001) method
Gamma	Baker and Hubert (1975) index
C	Hubert and Schultz (1976) index
Jump	Sugar and James (2003a) method
PS	Prediction strength method by Tibshirani and Walther (2005)
BI	Bootstrap instability method by Fang and Wang (2012)
CVNN	Liu et al. (2013) index
ARI	Adjusted Rand Index Hubert and Arabie (1985)

Chapter 1

Introduction

Clustering is a powerful tool to find underlying grouping patterns in data. A wide range of clustering methods and algorithms has been proposed in literature. The clustering methods can be broadly classify as partitioning and hierarchical methods. The partitioning methods are based on minimizing or maximizing a numerical function. They usually utilize the concepts of separation and homogeneity to perform clustering [Everitt et al. \(2011\)](#), i.e., objects within a group are closely located (intra - cluster compactness) and have cohesive structure, and they are well separated from the objects in other clusters (inter - cluster separation). There are a few major challenges while performing cluster analysis. The two major challenges among these are how many clusters are present in the data and which clustering algorithm is suitable to retain the clustering structure for the data application at hand. A lot of clustering algorithms need the number of clusters to be provided as a parameter. The process of determining the number of clusters is not straightforward and neither is the selection of the clustering algorithm. Another vital concern while performing cluster analysis is to validate the clustering results using some external criterion. The tasks of validation of clustering results and estimation of number of clusters are closely related. In this study two unified clustering methodologies are introduced using the clustering quality index for validation. This is based on the idea of optimizing the ASW index proposed by [Rousseeuw \(1987\)](#).

The task of cluster validation can be be broadly classify into two categories, namely, the internal validation indices that do not require any external information on clustering or the complementary to this that require external information to validate the clustering results such as true clustering or even the clustering computed from some other method than the one under evaluation. There are various cluster validation indices both internal and external proposed in literature. These indices are usually based on some criterion meaningful for clustering for instance within cluster compactness or between cluster separation.

1.1 Basic concepts related to validation indices

Diameter: The diameter of a cluster is defined by the distance between its two farthest objects ([Hartigan \(1975\)](#), [Han et al. \(2011\)](#), [Hennig et al. \(2015\)](#)). For a cluster C , belonging to a clustering \mathcal{C} on \mathcal{X} , the diameter can be defined as:

$$Diam(C) = \max_{x_1, x_2 \in C} d(x_1, x_2).$$

Separation: The separation of a cluster tells the degree to which a cluster is distinct from other clusters. There are various definitions for separation ([Dunn \(1974\)](#), [Davies and Bouldin \(1979\)](#), [Milligan \(1981\)](#), [Halkidi et al. \(2000\)](#), [Halkidi and Vazirgiannis \(2001\)](#), [Liu et al. \(2013\)](#)). For instance one way of defining it is: take the minimum distance out of all the pairwise distances between the observations of clusters in a clustering. Mathematically, for any two clusters $C, C^* \in \mathcal{C}$ it can be written as follows:

$$Sep(C) = \min_{x_1 \in C, x_2 \in C^*} d(x_1, x_2).$$

Compactness: The compactness or homogeneity of a cluster is defined by the intra-cluster variation. There are numerous ways of defining compactness. For instance the sum of squared deviation from mean can be used, as implied in k -means. For other definitions one can consult [Halkidi and Vazirgiannis \(2001\)](#), [Hennig et al. \(2015\)](#).

Generally the uniform separation and diameter across clusters are expected to ensure balanced clusters. For the compact clusters the small diameter value and high separation value are desirable.

Isolation: Another desirable property for a cluster is isolation. Usually, it is based on the concepts of diameter and separation. A well-isolated cluster is the one whose internal differentiation is lesser than external differentiation. There can be slightly different ways to apply this practically. For instance, a simple way to ensure this is by keeping the diameter of a cluster smaller than its separation. For other definitions of isolation see [Gordon \(1982\)](#) and [Fred and Leitão \(2003\)](#).

Connectedness/Cohesion: This concept ensures to what extend observations are connected within a cluster, i.e., cohesion within a cluster. It is based on observing local densities and checking whether the neighbouring items are in similar clusters or not. As the clusters should be well connected, similar/uniform densities within clusters are desirable. They are good in finding a wide range of clustering shapes. Many density based clustering indices have been proposed, for instance [Halkidi and Vazirgiannis \(2001\)](#), [Halkidi and Vazirgiannis \(2008\)](#) and [Moulavi et al. \(2014\)](#).

The selection of the indices for cluster validation depends upon the application at hand (see [Hennig \(2017\)](#) and [Hennig \(2015b\)](#)). If for a cluster application small

within cluster distances are required to define clusters k -means algorithm ([Hartigan and Wong \(1979\)](#)) can be used to produce clustering because its object function tries to minimize the distance of cluster points from the cluster mean. This clustering can be validate by the index based on the same objective, for instance, the Calinski and Harabasz index ([Caliński and Harabasz, 1974](#)). However, if there are more than one objectives required for clustering, for instance, the cluster should be well separated or as far as possible from each other as well as the cluster should be compact and must be represented by a centroid then the primary object can be used while clustering and the complementary criteria can be involved to validate clustering results. The numerical measure of how much of the other objectives has been achieved by this clustering can be measured using validation indices. It is particularly, useful to bring in the additional requirements for the situations when more than one criterion is needed that no clustering methods offer together yet. The aims of fulfilling several objects while clustering single data can be alternatively achieved by optimizing more than one objective functions in a weighted settings known as multi-objective or ensemble techniques ([Handl and Knowles, 2007](#)). To validate this clustering one can adopt similar approach in which one can use several criterion of interest in weighted setting as introduced in ([Hennig, 2017](#)). In this thesis we have taken an alternative approach to clustering from several criteria and focus on one criterion that is a combination of two objectives i.e., within cluster compactness and between cluster separation and developed a straight forward approach for clustering.

1.2 Definition of a cluster:

[Hennig \(2015b\)](#) argues that there is no general true clustering definition. The “true” cluster definition purely depends upon the aim of clustering and desirable characteristics. While defining clusters one need some intuitive assumptions and concepts. For instance, a cluster is a set of points that shares some characteristics or dissimilarity between neighboring points within a cluster should be smaller than dissimilarity between points between clusters. Different clustering methods aim at finding different kinds of clusters and no method is universally suitable for all problems. Usually, it depends in what domain clustering is required and what characteristic make sense in a given application, then these characteristics are matched with the clustering methods. For instance, the required characteristic can be translated into clustering language as within-cluster dissimilarities should be small, clusters should be of equal sizes, cluster can be represented by centroids, clusters should be well-separated, clusters should be of certain shapes (like elliptical), the number of clusters should be low or high, clusters should be stable or features within clusters should be independent etc. Each clustering methods have their own definition of clusters. Some methods take one value from the clusters like a centroid, a medoid or a clusteroid like k -means ([Lloyd \(1982\)](#)) or PAM

([Kaufman and Rousseeuw \(1990\)](#)) to represent a cluster. For some other methods a set of points from each clusters represents cluster, for instance, CURE ([Guha et al. \(1998\)](#)). Sometimes the clustering is done through probability models and the clusters are defined by the densities, for instance, clusters may be coming from multivariate normal distributions. In general, the formalisation of clusters is not straightforward, and for real life data it is usually unknown what the true clustering is, and there may not be any clearly identifiable clusters present. Finding a true clustering is usually not a problem if clustering is only required for administrative reasons but if the further analysis is based on the clustering results, like in image analysis or pattern recognition, it is imperative to find a clustering that depicts the real phenomenon as closely as possible.

If the number of clusters is not known in advance they can be first separately estimated by using some cluster quality index. The idea here is to choose the number of clusters that give the best clustering quality. The clustering solution is obtained for various numbers of clusters and a clustering quality index is calculated against all of these numbers to choose the best according to the criterion. Usually, different number of clusters are tried first with a clustering method and then one best among these clusterings is chosen using some index for measuring clustering quality in a relative comparison setup.

There are some problems in this approach. Firstly, the index used to access the quality of clustering is based on different statistical theories and concepts than the algorithm used to find clustering solution (for instance use of ASW with k -means algorithm or hierarchical clustering methods —[Chen et al. \(2002\)](#), [Bolshakova and Azuaje \(2003\)](#), [Reynolds et al. \(2006\)](#), [Saitta et al. \(2007\)](#), [Ganesan and Sukanesh \(2008\)](#), [Nguyen et al. \(2015\)](#)). This also applies, if first some formal method for the estimation of number of clusters is separately applied before finding the clustering against this estimated number. Secondly, performing these two tasks separately in real life problems is not straightforward and convenient for the users as highlighted in [Jain and Law \(2005\)](#). These problems are discussed in somewhat more detailed in the following paragraphs to motivate the need of the work conducted in this thesis.

The clustering algorithms, clustering quality measures and methods for the estimation of the number of clusters are based on some objective function. These objective functions are based on some criterion, for instance, homogeneity or compactness of clusters. Often in practice the users utilize one criterion to estimate the number of clusters, for instance, a criterion based on cluster separation, and use another clustering method for instance, based on cluster compactness to perform clustering. In reality, true cluster are not known and in some disciplines it is essential to apply cluster analysis before the actual data analysis, for instance functional magnetic resonance imaging (fMRI) data analysis, single cell RNA sequencing data analysis and analysis of the data simulated from climate models. The analysis to follow afterwards will rely on the clusters found in the beginning.

In situations where it is not known which kind of clusters to look for, the selection of method to estimate number of cluster and task of clustering become more challenging. From literature it is not hard to find the examples where optimization function that is used to estimate the number of clusters and the one that is used to get final data clustering differs in their functioning ([Liu et al. \(2003\)](#), [Ganesan and Sukanesh \(2008\)](#)). For instance k -means [Lloyd \(1982\)](#) (has its own notion of clustering which is to minimize the with-in cluster sum of squares, and this is different from the notion of ASW which is cluster compactness and separation) has been used with ASW index extensively to estimate number of clusters.

In this work we have a focus on developing clustering methods that are based on the same criterion to estimate the number of clusters and to define the objective function to get the final clustering solutions. The idea is if a criterion is acceptable for the estimation of number of clusters, then the clustering solution formed by this should also be acceptable. The advantage of this is that it will make the task of clustering somewhat simpler, and the users don't have to deal with the two tasks separately.

In this study we have defined a coherent framework to estimate the number of clusters and a clustering solution using the average silhouette width (ASW) proposed by [Rousseeuw \(1987\)](#). A clustering method can be defined by optimizing the objective function based on the ASW index. This index measures the clustering quality to estimate the number of clusters and has shown good performance for the estimation of the number of clusters. The motivation is that if an index is really good in estimating number of clusters then it should also be good in getting the final clustering solution.

The ASW is a well-reputed and trusted clustering quality measure. The index has been well received by the research community and is widely used for the estimation of the number of clusters. The index has been used across disciplines in various clustering applications for the estimation of the number of clusters and for comparing the clustering quality obtained from different clustering algorithms. There have been comparisons in the literature with other existing methods that validate the good performance of the index as compared to other indices. ASW was top performing index in [Arbelaitz et al. \(2013\)](#). In the next paragraph several references are provided from literature that conclude the good performance of index.

The ASW has been extensively used to estimate the optimal number of clusters (with a combination of various clustering methods), to compare the performance of clustering methods and for the quality assessment of clustering obtained from many clustering methods. Some empirical studies have also been designed to evaluate performance of the ASW in comparison with other famous indices. The index has been used for cluster analysis in a diverse range of data clustering problems and setups across disciplines, for instance geo-spatial analysis: [Ng and Han \(1994\)](#), clustering of time series: [Kalpakis et al. \(2001\)](#), for document clustering: [Recupero \(2007\)](#), for micro-array analysis: [Kennedy et al. \(2003\)](#), [Bandyopadhyay et al. \(2007\)](#), [Cho et al. \(2010\)](#), for

genotype assesment Lovmar et al. (2005) and for brain analysis: Craddock et al. (2012), for image segmentation: Hruschka and Ebecken (2003), Ganesan and Sukanesh (2008), Kannan et al. (2010) to mention a few. For clustering quality measures, and clustering method comparisons see Chen et al. (2002), Liu et al. (2003), Reynolds et al. (2006), Kannan (2008), Ignaccolo et al. (2008) and Arbelaitz et al. (2013).

A somewhat different use of the ASW index appeared in Lleti et al. (2004), where a method to determine the noise variables from the data was proposed. The authors have introduced various noise variables generated from the uniform distribution to the data and then tried to retrieve the original number of variables present in the data by optimizing the ASW index for k -means clustering. Campello and Hruschka (2006) have extended ASW to a fuzzy clustering regime. Some interesting variations and modifications have also been proposed, for instance, density based ASW by Menardi (2011) and the slope statistics by Fujita et al. (2014).

It is important to understand the behaviour or functioning of the index not only to propose a clustering method based on it but also to understand how the index works for the estimation of number of clusters.

For this work, we don't define in advance, what is the definition of clusters we are looking at, because this is not clear from the ASW definition what it actually tries to achieve. One can only roughly understand what ASW is aiming for. The definition does not fully specify what the shape of clusters are. For ASW one can roughly define what will be the characteristics of the resulting clustering based on ASW. It has its own notion that looks for homogeneous clusters, and separation from the closest cluster. There can be various ways of defining or achieving homogeneity and separation in a mathematical formula and ASW is one of them. There are also various ways of defining both within cluster homogeneity and between cluster separation. For instance ASW measures homogeneity by within cluster distances, thus for ASW it means small within-clusters distances. It is not so clear how separated and homogeneous clusters ASW can deliver, and in what situations it fails.

A few other things are however also understandable from the definition of ASW. For instance it is clearly different from the criterion like single or complete linkage that does not try to find compromise between these two aims of homogeneity and separation. Complete linkage looks for homogeneity and ignores separation and single linkage ignores homogeneity and delivers separated clusters.

There are other methods in the literature that roughly try to achieve the same goals i.e., homogeneity and separation of clusters, for instance k -means algorithm Slonim et al. (2013), or CH index Caliński and Harabasz (1974) or index proposed in Halkidi and Vazirgiannis (2001). The exploration of the similarity between ASW and other methods is not the primary focus of this thesis i.e., in what sense they are same and in what sense they are not. However, comments has been made if they deliver different results and if the reason behind this is so apparent.

It can not be said in advance what will be the shapes of clusters produced by ASW. Also, we can't make statements like the clusters produced by ASW will be, for instance, elliptical, symmetric in a sense that they are of equal sizes, or are defined by distributions such that the densities goes down very quickly and there are very few points in the tails of the distribution curve. ASW is also not a parametric measure of defining clusters for instance take mixture of Gaussian distributions as an example such that it can't be expected that the resulting clusters will look like them. This is something yet to explore that what kind of mixtures ASW is good in identifying and for what it is not so good. It is also worthwhile to explore what exactly happens in cases where it fails.

To develop more understanding about the ASW index we want to explore in this work in what situations this index can deliver true clusterings? As in real life application the purpose of clustering can be very different so to simulate these cases various data generating processes were considered. In reality clusters can be from Gaussian, Uniform, exponential distributions or other arbitrary shapes. To include various possibilities in this work, we generate data against many scenarios of real interest, to see, how ASW based clustering methods perform in each of these circumstances. Therefore, for us the true clusters mean the cluster generated from these cluster generating models. For the evaluation of clustering results obtained we use ARI to compare clustering using the true data generating clustering labels.

It is also worthwhile to explore if the ASW based clustering methods do not deliver the clusters as defined by the data generating process then what do they get and how do they make sense. Its worthwhile to explore, for what kind of data generating processes ASW is fine, and for what its not good and why? What exactly happens in the situations where it is problematic i.e., what characterises these situations, and why does it go wrong, and is there a possibility to argue that even where it goes wrong in terms of recovering true clustering i.e., if the proposed method gives a larger ASW as compared to existing methods but gives low ARI, does it still do something that may be useful in some sense. Thus we are interested in exploring to what extent and where does ASW deliver good ARI for recovering clusters as defined by the data generating processes, and where it doesn't, and how can it be characterised what it delivers instead?

1.3 Objectives and contributions of the thesis

The current dissertation covers practical and theoretical aspects of clustering. The primary objective of the current thesis is the investigation of the ASW based clustering methods. Are they good in finding any sensible clustering and if so what kinds of clusters they can retrieve. Are there some data structures that only these methods can find and existing clustering methods fail at finding? If this is so then a unified clustering method can be proposed based on this. The word unified is used here in a sense that the method can not only produce clustering but can also estimate the number

of clusters. For this kind of exploration the first question one face even before this is the standalone exploration of the ASW index. The goal in this thesis is to compare the performance of proposed and existing methods without favouring any of them. The thesis embodies the results of extensive simulations for the investigation of the newly proposed clustering function that is based on the idea of optimisation of the ASW.

- (i) **To learn which existing method can work well with the ASW index?** In principle ASW can be used with any clustering method to estimate number of clusters. However, some clustering methods can better capture certain kind of pattern in the data as compared to others, therefore, the performance of ASW will vary with the clustering methods. Goal here is to evaluate the performance of ASW with different clustering methods for two aims defined as **(a) performance of ASW for finding clustering solution, and (b) performance of ASW for the estimation of number of clusters.**
- (ii) ASW based clustering functions have been proposed in two most popular clustering domains, i.e., hierarchical and non-hierarchical. These algorithms are named as HOSil, OSil and fast versions of OSil. The performance comparisons of the proposed methods have been done with a range of clustering methods through simulations. The motivation for setting up these simulations was to illustrate the characteristics, and types of clusters the proposed algorithms can capture and identify. For this we define the two aims: **(a) performance of the proposed algorithms for finding clustering solution, and (b) performance of the proposed algorithms for the estimation of number of clusters.**
- (iii) **To find out the best way to initialization OSil** since initialization can effect the algorithm's output greatly ([Arthur and Vassilvitskii, 2007](#)). The algorithm proposed in the non-hierarchical settings (OSil) needs an initial clustering solution. For this several initialization methods were compared and evaluated against both aims.
- (iv) **To extensively evaluate the performances of other indices in comparative setting for the estimation of number of clusters in combination of various clustering methods.** [Milligan and Cooper \(1985\)](#), [Brun et al. \(2007\)](#), [Arbelaitz et al. \(2013\)](#) have conducted studies of empirical comparisons of validation indices and clustering methods. [Milligan and Cooper \(1985\)](#) covered 30 indices proposed during the period of 1965 to 1983 with 4 hierarchical clustering methods. [Brun et al. \(2007\)](#) covered 8 validation indices with 8 clustering methods. Lastly, [Arbelaitz et al. \(2013\)](#) covered 30 validation indices with three clustering methods. These or other such studies can not be fully generalized due to the availability of vast majority of clustering methods, validation indices, synthetic data sets, real data sets and characteristics of interests. To the best of our knowledge the

Gap Tibshirani et al. (2001), prediction strength Tibshirani and Walther (2005), bootstrap instability Fang and Wang (2012) , and CVNN Liu et al. (2013) indices never appeared in a comparative study together with other indices in a systematic setup with such a wide range of clustering methods considered in this work.

- (v) **Development of the axomatic theory for the ASW index.** There are different ways to investigate the quality and characteristics of clustering methods such as validation indices using simulations and real data experiments, model-based theory (Edwards and Cavalli-Sforza (1965), Binder (1978), Banfield and Raftery (1993), Bock (1996), and Fraley and Raftery (2002)), and non-model-based theory known as the axiomatic theory (Jardine and Sibson (1968), Fisher and Ness (1971), Kleinberg (2003) and Ben-David and Ackerman (2009)). In this work we have not only taken the empirical approach of validation of clustering results while simulations but also focus on the development of the axiomatic theory.

1.4 Outline of the thesis

The work conducted in this thesis has three broad divisions, which are as follows: The hierarchical clustering schemes, the non-hierarchical clustering schemes and finally the underlying fundamental theory development for the ASW index. In Chapter 2 basic concepts of clustering are introduced, terminologies and clustering methods and cluster estimation methods are defined to be used later in this thesis. The major challenges in cluster analysis are also discussed in there. In Chapter 3 a new linkage method is defined based on ASW optimization for the hierarchical clustering. We begin with the hierarchical clustering algorithm due to the advantage that it doesn't need any initialisation and the number of clusters are not needed to be specified beforehand. A major concern here is to explore the characteristics of the ASW index standalone in the various clustering settings together with the existing clustering methods and cluster estimation methods. We have then extended this approach to the non-hierarchical clustering. In Chapter 4 clustering methods based on optimization of ASW are proposed for the non-hierarchical clustering domain. A major part of this chapter is devoted for the exploration of the various initialization methods to find the optimal one for the proposed method. The behaviour of ASW is explored extensively. The runtime complexity issue for the algorithm is also taken into account for the proposed methods, and fast versions are introduced together with their exploration. A fairly large portion of Chapter 4 consists of a discussion of the simulations and summary of the results. Chapter 5 is a step towards the theoretical foundation for ASW. Chapters 3, 4 and 5 are independent from each other. The work done here points out towards various further challenges, issues and problems to be addressed, which points towards a vast possibilities for the future research. These directions are discussed in Chapter 6. The sim-

ulations conducted in this thesis are based on the probability distributions which are defined in Appendix A. Some additional results for the simulations in Chapter 3 and 4 are given in Appendix B and Appendix C, respectively. A numeric example related to the proof of Chapter 5 is presented in Appendix D.

Chapter 2

Clustering Overview

2.1 Some clustering applications

Clustering is a widely used multivariate data analysis procedure enjoying popularity in diverse scientific disciplines, for instance artificial intelligence, machine learning, computer vision, natural language processing, text analytics, sequence analysis, crime analysis, web searching and evaluation, archaeology, climatology, taxonomy, genetics, neuroscience and medicine, to name a few. The main objective is to classify similar objects into sensible groups called clusters according to some (dis)similarity criterion. Every clustering problem is unique, therefore there does not exist any universally acceptable definition of clustering. It is also possible that within a domain, there are several possible purposes to do grouping, and often there are several possible ways for grouping the subjects.

In genomics, clustering algorithms are used to automatically assign genotypes or to find biologically important subsets of gene from gene expression to infer population structures. Many clustering algorithms have been applied to find mutated genes i.e., to identify different diseases like cancer or diabetes. Some reference includes [Eisen et al. \(1998\)](#), [Alon et al. \(1999\)](#), [Van't Veer et al. \(2002\)](#), [Sturn et al. \(2002\)](#) and [Jiang et al. \(2004\)](#).

Clustering helps neuroscientists in understanding brain functioning by looking at activation levels of different parts of the the brain. Clustering techniques are used to define regions of homogeneity in the brain volume or on the cortical surface with respect to information provided by one or several images or task related activities. Clustering is used to provide the labels for voxels according to their similarity to identify the regions of interest for connectivity analysis for Functional Magnetic Resonance Imaging (fMRI) time series to isolate zones with similar activation or to investigate two voxels having similar behaviour. For instance see [Heller et al. \(2006\)](#), [Craddock et al. \(2012\)](#) and [Thirion et al. \(2014\)](#).

Clustering has been used for many environmental data sets for instance, real time storm detection and flood forecasting. Climate models are developed based on physical relationship between and within ocean and atmosphere. Many researchers have tried to implement clustering methods to climate data sets in order to understand different environmental factors and to classify climate zones. This is a progressive field where climatologists are trying to improve the climate models by investigating climate phenomena. For instance, El Nino Southern Oscillation (ENSO) is a very popular global climate signal phenomenon that affects the land temperature across the globe and is used to extract climate indices for different time slices at a location. In the process of extraction of important climate indices clustering is a vital step. The data are first clustered according to potential predictors such as continent or area of ocean or patterns in atmospheric pressure, etc., to identify these indices. For insight on the importance of selection of appropriate clustering method and their contribution towards the deeper understanding of climate process, see [Steinhaeuser et al. \(2011\)](#).

Galaxies are not randomly distributed in space, and they are of different colours, sizes and shapes. Their different properties make different types of clusters. Some of them are spiral shaped, for instance the milky way galaxy. Other observed shapes are elliptical, lenticular (disc shaped), rings, toothpick like shape and irregular galaxies. Clusters in galaxies are found based on their colours, shapes, locations and density. Recently, density based clustering algorithms have been modified for galaxy detection and classification for instance, see [Tramacere et al. \(2016\)](#) where they modify the DENCLUE¹ algorithm for identifying structures in galaxies. This is also a very active and flourishing research area.

Clustering algorithms are widely used for text mining and information retrieval by web search engines for quickly finding the nearest neighbours of a document to fulfil web search queries. Documents can be clustered on the basis of terms they contain or co-occurring citations to retrieve similar documents from the large set of documents efficiently. For instance see critical reviews and a collection of studies in this area in [Cooley et al. \(1997\)](#), [Willett \(1988\)](#), [Srivastava et al. \(2000\)](#), and [Berkhin \(2006\)](#).

In human motion analysis, it is important to know when the distribution of human pose changes ([Zhou et al. \(2008\)](#)). Human movements can be seen as multidimensional time series where clustering is used to segment these actions which helps in revealing unusual activities in videos. Similar applications are in decomposition of stream of facial behaviour into facial gesture where unusual facial expressions can be detected through the analysis of outlying temporal patterns. In forensic science and biometric systems ([Uludag et al. \(2004\)](#)), record databases are massive which requires rapid and efficient searching methods. Clustering makes this process of identification and verification of feature sets efficient by partitioning large biometric databases into most homogeneous groups, for instance, fingerprints, iris patterns, facial features, sig-

¹DENCLUE is a DENsity CLUstEring algorithm proposed by [Hinneburg and Gabriel \(2007\)](#)

natures etc.

Clustering can be used for dimensionality reduction, as the two are closely related. Dimensionality reduction methods use the closeness and correlation between dimensions to find a new set of reduced dimensions. The purpose of clustering in this case is to group the similar dimensions/variables based on some correlation measure instead of putting different instances across the dimensions in similar groups, which gives a reduced number of features in the original data set by keeping the loss of information as small as possible. Also, clustering provides labels for each object in the data, therefore it can also be used as a pre-processing step for supervised classification if the classes are not known a priori.

Clustering analysis has solved many machine learning problems. For instance, problems related to pattern recognition and image processing. Image segmentation is a typical clustering problem where the task is to partition pixels in such a way that pixels belonging to a region are similar to each other in order to identify objects, text or digits in an image (for instance see [Shi and Malik \(2000\)](#) and [Chuang et al. \(2006\)](#)). A challenging research area is photo OCR (optical character recognition) problem. Application includes car navigation systems where the car can read the street signs to navigate to the required destination.

Clustering methods are also been used for community detection ([Ye et al. \(2008\)](#), [Leskovec et al. \(2010\)](#), [Malliaros and Vazirgiannis \(2013\)](#)), market segmentation, understanding customers' behaviour and anomaly detection (for instance credit card fraud detection). There has been an increased trend in using clustering algorithms for temporal data mining in recent decades. Clustering time series have applications in economics, business, demography and medicine. For instance, clustering can be done to find countries sharing similar economic indicators. Forecasting can then be done by just predicting future trends for representative time series only from each group. This helps in forecasting a large number of time series by saving significant amounts of time and cost. For a literature survey on time series clustering see [Liao \(2005\)](#).

In the rest of this chapter a very brief overview of some basic concepts for cluster analysis are presented. This is then followed by essential definitions related to this work. Finally, the clustering methods, algorithms and cluster validation indices used later in this work are reviewed.

2.2 Preliminary definitions and concepts

Cluster analysis depends upon many concepts. Clustering algorithms differ according to certain properties like crisp versus fuzzy or deterministic versus stochastic clustering methods. The following subsections are designed to give all the important definitions and concepts to understand the background of this work. Here a purpose is also to setup general notations. However, only the elementary notations are defined here,

and more notations will appear as and when needed throughout the thesis. For the ease of readings the major notations are always briefly recalled before use.

2.2.1 Types and format of data

Clustering algorithms have been proposed for a variety of data types, for instance ordinal, nominal, ratio and mixed type data. Examples of these data include text data (social media, web, social networks), multimedia data (images, audio, video from Facebook, YouTube etc), time series data, sequence data (from web blogs, biological sequences) and stream data. Generally, the algorithms take data in two formats, namely variable/pattern/data matrix or proximity/affinity matrix. Depending upon the domains different names are more commonly used.

Definition 2.2.1. *Variable matrix:* Let n objects have p measurements then the variable matrix $X_{n \times p}$ is the one that displays a variable in each row against objects in columns. Formally, we can write

$$X_{n \times p} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1j} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2j} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \dots & \vdots \\ x_{i1} & x_{i2} & \dots & x_{ij} & \dots & x_{ip} \\ \vdots & \vdots & \vdots & \ddots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nj} & \dots & x_{np} \end{bmatrix},$$

where x_{ij} , $i = 1, \dots, n$; $j = 1, \dots, p$ represent the value of the j^{th} variable on the i^{th} object. Across various domains different names are used for the rows of the above matrix such as an object or a pattern or instances/individuals/cases/sample that are considered for clustering, whereas the columns are known as features or variables of interest. The data matrix can be simply seen as cases times variables. Thus we have observations from n objects over the p variables of interest.

The data is often represented in another format which is mostly used in this work. Let the n observations be $\mathcal{X} = \{x_1, \dots, x_n\}$, where each $x_i \in \mathcal{X}$, $i = 1, \dots, n$ is a column vector of length p representing the p observed variables on the i^{th} observation. Moreover, X will be used only to represent the variable matrix instead of $X_{n \times p}$.

Proximity matrix:

From the variable matrix some index of proximity or affinity can be established between pairs of patterns. This proximity index can be a similarity or dissimilarity. A similarity index indicates how similar the objects are to each other whereas a dissimilarity index is complementary to it. A dissimilarity index otherwise known as the distance function d , measures the pairwise distances between objects in \mathcal{X} .

Let \mathbb{R} be set of real numbers, \mathbb{R}^+ be the set of positive real numbers and \mathbb{N}_n the set of natural numbers excluding 0, up to n . Let \mathbb{R}^p represents a p -dimensional Euclidean space.

Definition 2.2.2. *Distance function:* A function $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$, is called a distance on \mathcal{X} if, it satisfies the following properties.

- (i) *Non-negativity:* $\forall x_i, x_h \in \mathcal{X}$, where $i, h \in \mathbb{N}_n$, $d(x_i, x_h) \geq 0$
- (ii) *Reflexivity:* $\forall x_i \in \mathcal{X}$, $d(x_i, x_i) = 0$, distance of an object to itself is zero
- (iii) *Symmetry:* for $x_i, x_h \in \mathcal{X}$, $d(x_i, x_h) = d(x_h, x_i)$

A function $d' : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$ is called a metric on \mathcal{X} , if in addition to above properties, the following properties also hold,

- (i) *Identity:* $\forall x_i, x_h \in \mathcal{X}$ $d(x_i, x_h) = 0 \iff x_i = x_h$
- (ii) *Triangle inequality:* $\forall x_i, x_h, x_r \in \mathcal{X}$, where $i, h, r \in \mathbb{N}_n$, $d(x_i, x_r) \leq d(x_i, x_h) + d(x_h, x_r)$.

Definition 2.2.3. *Minkowski distance:* The Minkowski distance of order q between two objects x_i and x_h in $\mathcal{X}, \mathcal{X} \in \mathbb{R}^p$ is defined as follows

$$d_q(x_i, x_h) = \left(\sum_{j=1}^p |x_{ij} - x_{hj}|^q \right)^{(1/q)}.$$

This distance is mostly used with $q = 1$ (Manhattan distance) or $q = 2$ (Euclidean distance).

2.2.2 Data clustering/partitioning

The purpose of clustering or partitioning is to split the data into k groups called clusters. The task is to divide the data into coherent structures. More formally, denote a partitioning of \mathcal{X} into k groups by the set \mathcal{C}_k . The members of \mathcal{C}_k will be called clusters. Let k be an integer such that $1 \leq k \leq n$ be the number of subsets in a particular partitioning \mathcal{C}_k .

Definition 2.2.4. A k -clustering $\mathcal{C}_k = \{C_1, \dots, C_k\}$ of a data \mathcal{X} is a partition of \mathcal{X} into k disjoint subsets of \mathcal{X} .

Clustering function/criterion:

A clustering function f_k takes as an input a set \mathcal{X} and a distance function d , and a pre-known fixed number of clusters, say k , to return a clustering of \mathcal{X} .

Definition 2.2.5. A partitioning function $f_k(\mathcal{X}, d) = \mathcal{C}_k$; $\mathcal{C}_k = \{C_1, \dots, C_k\}$ provides k - partitions such that, $\emptyset \neq C_i \subset \mathcal{X}$ for $i \in \mathbb{N}_k$, $C_i \cap C_j = \emptyset$ for $i \neq j$, and $\cup_{i=1}^k C_i = \mathcal{X}$.

Note that according to the above definition f_k needs some definition of distance function d and number of clusters k as an input to get the clustering for \mathcal{X} . The above definition ensures that the clustering functions f_k gives a clustering such that no member C_k in a k -clustering is empty. All the members are pairwise disjoint and all the members are collectively exhaustive. Examples of the clustering criterion functions that operate like this are partitioning around medoids (PAM) or hierarchical clustering methods formally defined in Section 2.4. Note that not all clustering criterion function provide k disjoint (see Section 2.2.3) subsets of \mathcal{X} .

Alternatively, there are some clustering methods that take as an input the data matrix as given in Definition 2.2.1 instead of the proximity matrix to cluster the data and a known number of clusters beforehand such as the k -means clustering method. This partitioning function can be defined by $f_k^*(\mathcal{X}) = \mathcal{C}_k$. Finally, there are also some clustering methods that take only a data matrix as an input and they can estimate number of clusters, say \hat{k} , themselves to return a clustering. In this situation the clustering function can be defined as $f'(\mathcal{X}) = \mathcal{C}_{\hat{k}}$. More details on these types of functions will come in the section where these methods are reviewed. Various clustering algorithms has been studies in this work. Not all of them required same kind of input. We will use both variable matrices and proximity matrices as an input to clustering algorithms in this work and also the three kinds of clustering functions just defined in the study. Finally, the two special clustering cases defined as $k = 1$, when all the data forms a single cluster and $k = n$, when each point forms its own cluster are not of interest for the work done here.

Definition 2.2.6. *Clustering labels:* For a given partition, the clustering label set defines the cluster memberships of all the observations in the variable matrix. For $x_i \in C_r$, $r \in \mathbb{N}_k$, the label of x_i for $i \in \mathbb{N}_n$ is $c_i = r$. Therefore, a complete labels' vector for a partition is (c_1, \dots, c_n) , where c_i represents a label for object 'i' and each of $c_i \in \mathbb{N}_k$.

Clustering label vector is an integer vector which has values between 1 and k . The length of this vector is equal to the number of observations n in the data. For each index i , $i \in \mathbb{N}_n$, the coordinate c_i is equal to the number r , $r \in \mathbb{N}_k$ representing the cluster number observation x_i belongs to.

2.2.3 Crisp versus fuzzy clustering

Clustering is often differentiated into crisp or fuzzy. Crisp, also known as hard clustering, is the one for which all objects in the data just belong to exactly one cluster.

Mathematically, we can write the cluster labels returned from a hard clustering algorithm as $k \times n$ matrix H as followed:

$$H = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \dots & c_{kn}, \end{bmatrix}$$

where n denotes number of objects in a data set, k denotes the number of clusters and c_{ri} satisfies

$$c_{ri} \in \{0, 1\}, \quad 1 \leq r \leq k, 1 \leq i \leq n, \quad (2.1a)$$

$$\sum_{r=1}^k c_{ri} = 1, \quad 1 \leq i \leq n, \quad (2.1b)$$

$$\sum_{i=1}^n c_{ri} > 0, \quad 1 \leq r \leq k. \quad (2.1c)$$

Each row of matrix H is a binary vector. The rows represents cluster membership and column represents the observations. Constraint (2.1a) ensures that each object either belongs to a cluster or not, Constraint (2.1b) shows that each object just belongs to one cluster, i.e., only one element in each column of matrix H will be one and all others will be zero. $c_{ri} = 1$ means that observation ‘ i ’ belongs to cluster r . Constraint (2.1c) is because each cluster should contain at least one object. Note that the label vector (c_1, \dots, c_n) defined in Definition 2.2.6 represents hard clustering and can be converted to a matrix H defined above. Whereas, in soft or fuzzy clustering, each object belongs to each cluster with a certain cluster membership score also called degree based on how similar the object is to other objects in that cluster. For instance, for each data point x_i , a certain degree for each cluster in a clustering can be calculated depending upon the closeness of the point to the center of a clusters. For fuzzy clustering constraint (2.1a) changes to

$$c_{ri} \in [0, 1], \quad 1 \leq r \leq k, 1 \leq i \leq n.$$

Note that the description above for fuzzy clustering is very basic and is just intended to give an idea to differentiation between these two types of clustering domains to identify the present interest of this work. In fuzzy clustering the membership scores are not necessarily probabilities i.e., they don’t always fulfil (2.1b) and can be defined in other ways. In this thesis only hard clustering methods are used. As fuzzy clustering is not required for this work hence not discussed in further detail. To summarize, a crisp clustering generates hard assignments of objects to clusters as for any two clusters C_r and $C_{r'}$, we have, $C_r \cap C_{r'} = \emptyset$ such that $\cup_{i=1}^k C_i = \mathcal{X}$. This also agrees with the definition of a partitioning function given in Definition 2.2.5.

2.2.4 Partitional versus hierarchical clustering methods

All partitioning based clustering methods give flat clustering , meaning that they partition the data into non-overlapping clusters and treat these clusters at the same level in clustering i.e., clusters are not nested inside clusters. They return a single partitioning of a data instead of series of partitions, and no further structures are seen within clusters. If a cluster is further sub-divided into different clusters, then the formed clusters are at different levels. The hierarchical clustering, gives the clusters in a hierarchy. Smaller clusters are merged into the bigger clusters, or bigger clusters could be further partitioned such that the hierarchy of nested clusters is generated. There are some key differences between the two methods in terms of assumptions, clustering results, input parameters and runtime. Hierarchical clustering does not require the number of clusters or a set of initial points to begin with as many partitioning clustering methods, instead just the similarity or dissimilarity measure is needed to perform the clustering. Partitioning methods are typically faster than hierarchical clustering methods.

2.2.5 Deterministic vs. stochastic clustering methods

Deterministic clustering methods always arrive at the same clustering result for the given data for instance see [Chi and Lange \(2015\)](#) and [Everitt et al. \(2011\)](#). Some examples are average linkage, complete linkage, Ward's method and McQuitty's methods. All these methods are hierarchical clustering methods, and are defined in Section [2.4.1](#). Examples of other deterministic clustering methods include the PAM algorithm reviewed in Section [2.4.2](#), spectral clustering reviewed in Section [2.4.4](#) or the dbscan algorithm. In contrast, stochastic methods do not reach at same clustering solution if run more than once with the same parametric choices on the same data, for instance, the standard k-means method reviewed in Section [2.4.2.1](#) for a different set of initial points chosen as starting centres can reach different local minima resulting in different clustering solutions. Similarly, model-based clustering defined in Section [2.4.3](#) is also a stochastic clustering method. Also, both k -means and model-based clustering can be run in deterministic fashion. In this work we have used both deterministic and stochastic clustering methods. The new proposed algorithms in this work, can be classified as stochastic clustering methods.

2.3 Challenges in cluster analysis

There are various challenges in clustering, for example, what is the suitable distance measure for the given data? How many clusters are present in the data? How do we validate the clustering results once found? Do we need to standardize the data before clustering? Is the clustering method robust to outliers? Are the clustering results sta-

ble and reproducible? Is the clustering method scalable to large databases? Are the clustering results obtained generalizable to new features or to new observations of the same features? Are the results interpretable? Some of these issues relevant to this work are discussed in some detail in the following subsections.

2.3.1 Selection of the proximity measure

The clustering methods are sensitive to the distance measure used. The selection or definition of a suitable distance measure is an important concern in clustering. What is a meaningful similarity between the objects for a particular application and how variables should be weighted in construction of such measures is an interesting question. There is a whole range of different similarity and dissimilarity measures suitable for different applications and for different types of spaces, see for instance [Deza and Deza \(2009\)](#). The most commonly used ones are Euclidean, Minkowski, Mahalanobis, Manhattan, Canberra, Geodesic, correlation, cosine angle, hamming, simple matching coefficient, Jaccard and its variations known as Tanimoto distances. Note that the last two mentioned in this list are similarity measures. For a detailed description of some of these for binary, categorical, continuous and mixed data types the reader are referred to Chapter 3 of [Everitt et al. \(2011\)](#). The choice of proximity measure depends upon various factors such as meaning of closeness in a certain application, what types of clusters are to be discovered, the type of the data (e.g., binary, continuous or mixed), the space of the data (e.g., Euclidean or non Euclidean), nature of the analysis and the clustering method to use. For hierarchical clustering methods one also has to decide what linkage method to use, which will also influence the shape of the clusters formed.

2.3.2 Estimation of the optimal number of clusters

It is crucial to decide the appropriate number of clusters for the data set. This is hard to identify, as there is no single operational definition of a cluster and objects are cluster with different purposes in mind. A careful analysis is needed to find the appropriate number of clusters in the data. There are many methods available, for instance see [Milligan and Cooper \(1985\)](#) for the relative comparison of some most often used methods. Some methods to estimate the number of clusters includes average silhouette width ([Kaufman and Rousseeuw \(1990\)](#)), gap statistics ([Tibshirani et al. \(2001\)](#)), prediction strength ([Tibshirani and Walther \(2005\)](#)), distortion curve ([Sugar and James \(2003b\)](#)), Calinski and Harabasz's index ([Caliński and Harabasz \(1974\)](#)), Dunn's Index ([Dunn \(1974\)](#)), Hartigan's rule ([Hartigan \(1975\)](#)), Kranowski and Lai criterion ([Krzanowski and Lai \(1988\)](#)) and Bayesian information criterion ([Fraley and Raftery \(1998\)](#)). One should keep in mind that different representations of the same data may give different numbers of clusters from the same clustering method. The relationship between the variables is vital information to identify an appropriate number of clusters. Thus, the user

has to choose a suitable representation of data and features/variables to use in the analysis, and the number of clusters found will depend upon these choices.

2.3.3 Validation issues

Clustering validation is the evaluation of accuracy, quality and goodness of clustering results. It is an essential task to determine the usefulness of clustering obtained. Most clustering algorithms will give some partition of the data even if there are no inherent clusters present. Therefore, it is important to assess whether the dataset contains any meaningful clusters or not in the first place. This problem is known as clusterability ([Ackerman and Ben-David \(2008\)](#)), that is, to check the clustering tendency and to validate whether some non-random structure is present in the data or not, and whether cluster analysis is sensible to perform. Another issue is that, the different clustering algorithms (or even the same algorithm run twice) may yield different partitions of a data set leaving the decision to the users to chose the one that is most meaningful for them. Clustering validation is an exploration process to help users to find out which among the candidate partitions make most sense for the given application.

The task of clustering quality validation and estimation of number of clusters are closely related. Many of the clustering quality indices are used for the estimation of number of clusters. The number of clusters can be chosen by optimizing a quality index. Keep in mind not all the indices will work for the estimation of number of clusters. Some indices will systematically give higher (or lower) values for larger (or smaller) numbers of clusters so that they are hard to compare across different numbers of clusters. For example, criteria such as within cluster sum of squares will become smaller and smaller if the number of clusters is increased, and so optimising them cannot be done for estimating the number of clusters.

Clustering quality measures can be classified as external and internal validation measures ([Handl et al. \(2005\)](#), [Halkidi et al. \(2001\)](#)). External validation methods take knowledge of known class labels to validate clustering algorithms on data sets to learn about the performance of a method and to prepare them for more challenging real data sets. They are also used to compare the clustering results coming from different methods. In situations where external labels are not known, internal validation measures can be used to validate the clustering. Internal validation measures explore how well the clustering fits the data set using some criterion. A special case of the latter is the evaluation of clustering consistency or stability through resampling methods, see [Fridlyand and Dudoit \(2001\)](#). Beside these, there are relative measures that compare different clusterings on a given data set resulting from various parametric choices for an algorithm. This is often used to decide the optimal number of clusters for the data sets. For an introduction about validation procedures, [Jain and Dubes \(1988a\)](#) and [Hennig et al. \(2015\)](#) are good references to start. For comparison between different clustering validation methods see [Halkidi et al. \(2001\)](#), [Bolshakova and Azuaje](#)

(2003), Handl et al. (2005), Brun et al. (2007), Arbelaitz et al. (2013), Xiong and Li (2013, chap. 23) and Lei et al. (2017).

There are different clustering aims, and what the "best" clustering is and how this is measured depend on these. There is no unique best clustering in a dataset and there is no unique best criterion or index to measure this. Likewise, different indices are based on different objectives as they measure different characteristics of clusterings, most of which are legitimate and of interest in at least some applications, but some of them may contradict some others. It is, however, not well understood and investigated how to exactly characterise what different indexes do and how they differ, so that in a given situation a user could have a clear idea which one to choose. No literature exists, that explains as precisely as possible, for what specific situation an index is good and for which it is not so appropriate.

The relationship between the clustering method and the index is not clear and cannot be generalized. Therefore, not all the validation indices can be used with all clustering methods. The properties associated with validation measures are important when selecting the index. Like clustering algorithms, clustering validation measures are also based on certain concepts. For instance, some internal validation measure may be strongly connected to the concepts of separation, isolation, or compactness. Due to these properties some of the indices can give a higher value of index for a clustering method with which they share their criterion. A validation criterion based on within cluster sum of squares like CH, H, Gap etc are appropriate to validate clustering that aims at minimizing average within cluster distances like k -means. On the other hand, if a criterion which is not based on within cluster sum of squares is not useful for assessing clustering obtained from a method based on within cluster distances criterion.

Another important challenge in cluster analysis is the definition of clusters. i.e., what truth the researchers are trying to recover. This is discussed together with what is the definition of clusters for this work after reviewing ASW index. Therefore, this is provided in Section 1.2 after ASW is reviewed.

2.4 Clustering methods review

Several clustering methods have been proposed in literature. The classification of the clustering methods is not straightforward and different authors classify them in different categories based on different factors. For instance some based their classification on input requirement of methods and on outputs. Yet some classify based on clustering criterion e.g., distance-based, search-based, density based. A slightly different classifications can be found across literature, see for instance Murtagh (1983), Handl et al. (2005), Berkhin (2006), Han et al. (2011, chap. 10), and Hennig et al. (2015). The methods can be broadly classified as distance based methods, density based methods, grid based methods, graph or network clustering methods, kernel clustering methods,

constraint based methods, Bayesian parametric and non-parametric clustering methods. Many clustering methods have specially been designed by keeping in mind the complexity issues of the data or the existing clustering methods have been scaled up for big data sets. Many hybrid clustering methods have also been introduced using combinations of these methods. Among the distance based methods we have partitioning and hierarchical methods. The following sections provide a review to the clustering methods that are relevant to this work.

2.4.1 Hierarchical clustering methods

Hierarchical clustering is based on a concept that builds clusters gradually and gives a series of partitions from an individual cluster to n clusters or vice versa. The number of clusters are not needed to be fixed in advance, however, if the desired number of clusters k is known already the partitioning can be stopped when the required clusters are obtained. The standard hierarchical clustering methods are deterministic and further classified in literature as agglomerative (bottom-up) and divisive (top-down) methods (see, for instance, [Everitt et al. \(2011\)](#)). The classification may be represented by a tree-like diagram called dendrogram which shows the data partitioning at each stage. This helps in evaluating the data structure and deciding the appropriate number of clusters. Only the agglomerative hierarchical clustering (AHC) methods are used in this thesis.

At level 0, a root represents an individual data point to be clustered, thus all the roots are the data points to be clustered (see [Hennig et al., 2015](#), chap. 6). These roots form "singleton" clusters in the beginning. At every level, the root nodes are further merged. Every level in the hierarchy corresponds to a set of clusters, thus, returning a sequence of clusterings. This sequence of clusterings is usually represented by a tree diagram. The tree can be cut at a certain height to get a single partitioning of the data. More formally, let \mathcal{X} be the data to be clustered, where x_{ij} represents the j^{th} measurement taken on the i^{th} object. A dissimilarity d on \mathcal{X} is required to calculate a pairwise distances between data points. Let a single partition of data be represented by $\mathcal{C} = \{C_1, \dots, C_k\}$. Let a sequence of partitions be represented by $\mathcal{P} = \{\mathcal{C}_1, \dots, \mathcal{C}_L\}$, where $L \leq n$ and $n \in \mathbb{N}$. Let k_1, \dots, k_L be the cardinalities of these partitions, then at a particular hierarchy level a single partition can be denoted by $\mathcal{C}_l = \{C_1, \dots, C_{k_l}\}$, for $l = 1, \dots, L$. Note that $k_1 = n$ and $k_L = 1$. Hierarchical clustering will satisfy the following conditions on a sequence of partitions \mathcal{P} ,

- (i) $k_1 > k_2 > \dots > k_L$, where $k_l = |\mathcal{C}_l|$, $l = 1, \dots, L$ and $|\mathcal{C}_l|$ is the cardinality of \mathcal{C}_l .
- (ii) $C_i \cap C_j = C_i$ or $C_i \cap C_j = C_j$ or $C_i \cap C_j = \emptyset$ for all $C_i \in \mathcal{C}_i$ and for all $C_j \in \mathcal{C}_j$ with $i < j$.

Hierarchical clustering methods need to measure proximities between clusters to merge or split the clusters in agglomerative and divisive clustering respectively, com-

monly known as linkage methods. These proximities can be measured in form of similarities or dissimilarities. Hierarchical clustering can be done with any similarity or distance measure and for any type of variables, for instance categorical or numeric as long as a distance measure can be defined. A linkage is a measure of proximity between two clusters [Berkhin \(2006\)](#). A distance measure for the inter-cluster pair of observations and the linkage criterion which are the functions of these pairwise distances are needed to amalgamate clusters. After each split or merge the total number of clusters increases or decreases by one.

There are many linkage methods, all of which can be derived from the [Lance and Williams \(1967\)](#)'s dissimilarity formula. A list consists of standard linkage methods together with their definitions are given in Table 4.1, Page 79 of "Cluster Analysis" by [Everitt et al. \(2011\)](#). Before presenting the AHC's algorithm some linkage methods are recalled below to be used later in this work.

2.4.1.1 The average linkage

The average linkage by [Sokal and Michener \(1958\)](#) takes the average of pairwise distances between all the points in the two clusters considered for merging. Those two clusters at the same level in hierarchy are merged that give the minimum average. Consider any two clusters $C, C' \in \mathcal{C}$, and let n_C and $n_{C'}$ denotes the number of objects in these clusters then average linkage can be defined as

$$D_a(C, C') = \frac{1}{n_C n_{C'}} \sum_{x_i \in C} \sum_{x_j \in C'} d(x_i, x_j),$$

where D_a is the distance between two clusters and any distance measure can be used to calculate the distance $d(x_i, x_j)$ between pairs of points. $D_a(C, C')$ is calculated for all clusters in a partition at a certain level and those two clusters are merged that give the minimum average linkage.

2.4.1.2 The single linkage

Single linkage is among the oldest clustering methods, see [Graham and Hell \(1985\)](#) for the history of its development. It was developed by [McQuitty \(1957\)](#) independently among others. It takes the minimum pairwise distances between the members of two clusters, i.e., it joins the two closest clusters to form a new cluster. Mathematically, single linkage can be defined as follows:

$$D_s(C, C') = \min_{x_i \in C, x_j \in C'} d(x_i, x_j).$$

The two clusters are combined that give the minimum $D_s(C, C')$ over all clusters.

2.4.1.3 The complete linkage

Complete linkage ([Sorensen \(1948\)](#)), joins the least similar pairs in clusters together, i.e., it takes the maximum distance between two objects in clusters to form a new cluster. Mathematically, complete linkage can be defined as follows:

$$D_c(C, C') = \max_{x_i \in C, x_j \in C'} d(x_i, x_j).$$

The $D_s(C, C')$ is calculated between all the clusters in a partition at a certain level and those two clusters are combined that give the minimum D_s .

2.4.1.4 The McQuitty method

[McQuitty \(1966\)](#) has proposed a method to calculate the proximity between clusters using both, the measure of distance or similarity. This method is also known as McQuitty similarity analysis. Some measure of association between objects has to be calculated first, for instance, it can be the correlation between multiple responses of the objects as used in [McQuitty \(1966\)](#). For the n observations taken over p variables compute correlation between each pair of object. These similarities are stored in the matrix. To start off, as per hierarchical clustering rule construct the first hierarchy by putting each observations in its own cluster. There will be C_1, \dots, C_n clusters. Next, note the first maximum and the second maximum values from the similarity tables and combine the corresponding clusters to them. Let C_1 and C_2 be the clusters that were just merged into one cluster, denote this newly formed cluster as $C_1 \cup C_2$. After merging these two clusters the new matrix of similarity is to be calculated. For this delete two individual rows C_1 and C_2 from the previous similarity matrix and add a combined row for $C_1 \cup C_2$. For this the similarity of $C_1 \cup C_2$ needs to be calculated to all other rows (clusters). This is done through the McQuitty similarity. Let C_r represent any other cluster from \mathcal{C}_l , where \mathcal{C}_l denotes a partition at hierarchy level l . Let the entries in the similarity matrix (for instance similarity between C_1 and C_2) is denoted by $a(C_1, C_2)$. Then the McQuitty similarity for a cluster formed by joining two clusters C_1 and C_2 , now denoted as cluster $C_1 \cup C_2$ with another cluster C_r is calculated as follows. For all $C_r \in \mathcal{C}_l$ calculate

$$Sim_{Mc}(C_1 \cup C_2, C_r) = \frac{a(C_1, C_r) + a(C_2, C_r)}{2}.$$

After construction of the new matrix, the maximum entry of the whole matrix is noted again to merge the corresponding clusters. The process is continued in the analogous ways until every object is in one cluster. Thus this measure joins those two clusters whose average similarity value is highest.

Note that as described in [McQuitty \(1966\)](#) the McQuitty similarity will give higher

weight to those two clusters that have fewer pairs of objects in them as compared to the pair of clusters that have higher numbers of pair of objects in them. For instance let $n_{C_1} = 2$, $n_{C_2} = 3$ and $n_{C_r} = 5$ then $\frac{1}{n_{C_1} n_{C_r}} > \frac{1}{n_{C_2} n_{C_r}}$. Due to this unequal weighting this method is often criticized. We have decided to include this method in the analysis as for the ASW it is still not clear and well understood how it deals with different sizes of clusters or how these differences in the numbers of pairs in different clusters affect its performance, so we hope to learn something by including this method for ASW.

2.4.1.5 The Ward's minimum variance method

Proposed by [Ward Jr \(1963\)](#), this criterion looks at how much the total within clusters sum of squares increases when two clusters are combined. In agglomerative hierarchical clustering, since every object forms a cluster on its own in the beginning, we have a zero sum of squares within clusters. Ward's linkage will combine two clusters if the sum of squares deviation from the mean of the newly formed cluster is minimum of all combinations of clusters. Let SSE denotes the sum of squared errors for clustering, then mathematically we can define it as follows:

$$SSE(\mathcal{C}) = \sum_{r=1}^k \sum_{i=1}^{n_r} (x_{ir} - \bar{c}_r)^2,$$

where d is the Euclidean distances, x_{ir} is the i^{th} object in the r^{th} cluster, n_r is the number of objects in the r^{th} cluster and \bar{c}_r is the mean of the r^{th} cluster.

There are many versions of agglomerative hierarchical clustering algorithms depending upon different linkage metrics. There are some methods that represent clusters with some kind of statistical averages also called prototypes like the centroids or medoids, and the decision about merging the clusters is based on the distance between the centroids or medoids of the clusters. Centroid, medoids or mode are the commonly used averages, all of these are the most representative points of a cluster. A centroid is calculated by taking the average of all the points in the clusters. A medoid is restricted to the actual data point from the cluster, usually a medoid is that data point, which gives the minimum average dissimilarity from all the other data points in a cluster. In this work we will not use these kinds of linkage measures that require clusters to be represented by some averages, as the proposed method in this work is free of cluster averages.

The choice of the linkage metric will significantly influence the clustering solution, as they are based on a particular concept of propinquity. Hierarchical clustering can capture clusters of arbitrary shape. The choice of metric influences the shape of clusters strongly. There are certain properties attached to each linkage method. Each of these linkages and others not defined here, are good at performing well in certain situations and fail in others. For instance, single linkage is relatively good at handling

AHC algorithm

Initialize

Set $l = 1$. Initialize n singleton clusters i.e., every object forms its own cluster,

$$\mathcal{C}_1 = \left\{ C_1, \dots, C_{k_1} \right\} = \left\{ \{x_1\}, \{x_2\}, \dots, \{x_n\} \right\}, \quad k_1 = n.$$

Repeat

- (i) Calculate the pairwise distances between all the pairs of clusters in partition \mathcal{C}_l , i.e., $D(C_i, C_j)$ for $C_i, C_j \in \mathcal{C}_l$.
- (ii) Merge the pair (C_i, C_j) that minimizes $D(C_i, C_j)$ for $C_i, C_j \in \mathcal{C}_l$, i.e., merge the least dissimilar pair, such that at hierarchy level “ l ”;

$$\mathcal{C}_{l+1} = \mathcal{C}_l \cup (C_i \cup C_j) \setminus \{C_i, C_j\}, \quad k_{l+1} = k_l - 1,$$

$$l = l + 1.$$

- (iii) Stop when $l = n$, i.e., $k_l = 1$ and $\mathcal{C}_l = \{\mathcal{X}\}$.
-

Note: $C_i \cup C_j$ means that cluster C_i and C_j are amalgamated and \ represent the set division. This means that the two individual clusters C_i and C_j from \mathcal{C}_l are excluded, and one cluster which is formed by merging these two clusters is added to \mathcal{C}_l .

In case a similarity method is used to merge two clusters, the pairwise similarities between all the pairs of clusters is maximized.

complex shapes as it joins two clusters if even one pair of points is close enough. Like many other methods, it is also good in identifying outliers as singletons if they are sufficiently far away from clusters. Single linkage ensures separation, it doesn't care about compactness and produces chain-like clusters. Complete linkage usually forms homogeneous clusters of almost equal sizes. Single and complete linkage are invariant to the ordering of objects.

Ward's method assumes the points to be in Euclidean space and tries to find spherical and equally sized clusters. The McQuitty similarity is good in finding uneven sized clusters. Also its performance is often close to the single linkage method. For a detailed discussion of these methods the readers are referred to Chapters 4 & 6 of [Everitt et al. \(2011\)](#) and [Dunn and Everitt \(2004\)](#), and the references therein.

Hierarchical clustering has a close connection with graph or network clustering ([Karypis et al. \(1999\)](#), [Schaeffer \(2007\)](#)). It can be viewed as multilevel decomposition of graphs in a tree structure. The observations in the data set can be seen as nodes or vertices and these nodes can connect by weighted edges/links based on some similarity measure such that the original graph is divided into clusters. A point's neighbours are those that are similar to it in some sense, and some similarity function can be used

to measure this similarity, where the similarity function can be metric or non-metric.

Zhang et al. (1996) proposed BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), which recodes the statistical summary of clusters as a clustering feature tree of a dense data portion and stores the summary information about each cluster to update clusters through scanning data. It pre-clusters the data and then uses a centroid based hierarchical algorithm to give a final clustering of the pre-clustered data. BIRCH can only find spherical clusters.

CURE (Clustering Using REpresentatives) proposed by Guha et al. (1998) tries to take advantage of the good properties of both partitioning and hierarchical methods. CURE first chooses a constant number of representative points that can capture possible clusters present in the data. These points are then moved towards the centroid of the cluster by a shrinking fraction to reduce the effect of outliers on clustering. Two clusters are merged by measuring the closeness between the pair of representative points in these clusters.

ROCK (RObust Clustering using linKs) proposed by Guha et al. (1999) is an agglomerative hierarchical algorithm for boolean and categorical variables. It establishes links between data points instead of distances or the Jaccard coefficient to measure closeness. A link is the number of common neighbours between two points. The objective function maximizes the sum of links between all the pairs of points in a single cluster.

Karypis et al. (1999) proposed CHAMELEON, a two-phase algorithm using a dynamic model to account for special characteristics of individual clusters. In the first phase using a K - nearest neighbour (KNN) graph, it clusters data into a large number of small sized clusters. Then in the second phase it uses an agglomerative hierarchical algorithm to combine these clusters to get the final clustering. Weighted edges were used as the similarity between the nodes. For merging two clusters CHAMELEON takes advantages of both CURE and ROCK. It combines two clusters if inter-connectivity and closeness between two clusters (ROCK) are high as compared to the internal inter-connectivity of the clusters and the closeness of items within the clusters (CURE).

All the algorithms mentioned here suffer from some kind of limitation. Either their time complexity is too high for even a few thousands data points, or they impose restrictions on clusters' shapes and are designed only for specific data sets. Linkage metric based methods have non-linear time complexity, $O(n^2 \log(n))$ and $O(2^n)$ (see for instance Sibson (1973), Defays (1977)) where n is the number of observations, so they are not efficient for large input data.

Some general limitations also apply to hierarchical clustering methods. For instance, if a split/merge has been made at an earlier stage it cannot be undone, i.e., two individuals that have been joined in a cluster will remain in the same cluster forever and cannot be subsequently separated at later joining stages. The hierarchical clustering methods are based on local search and have no clear global optimization goal, and it is non-trivial to choose an appropriate distance measure.

2.4.2 Partitioning clustering methods

The partitioning methods are based on minimizing or maximizing a numerical function. They usually utilize the concepts of separation and homogeneity to perform clustering ([Han et al. \(2011\)](#)), i.e., objects within a group are closely located (intra - cluster compactness) and have cohesive structure, and they are well separated from the objects in other clusters (inter - cluster separation). These methods attempt to divide the data into k non-overlapping groups by optimizing a criterion function. Many clustering criterion functions are prototype-based, meaning that they try to capture the closeness of the data to some particular point or set of points known as prototype(s) of the cluster. A prototype defines a cluster and can be a centroid for instance, the mean, the medoid, the median, or the mode. Other hybrid ideas, for instance, CURE based on centroid based and hierarchical clustering can also be used.

k -means is one of the most popular and widely used clustering methods. It is also recognized as widely cited and used algorithm, and is considered among the most influential data mining algorithms. Some kind of implementation of the algorithm exists in almost all well-known and widely used statistical software systems. The origin of k -means and its various algorithms can be found in [Bock \(2008\)](#). Perhaps the idea was for the first time proposed by a Polish mathematician ([Steinhaus \(1956\)](#)²) in French. Though the paper did not use the term k -means, it yet defines its principle for continuous data in finite dimensions. However, no data application was presented.

Lloyd in 1957 essentially worked on the same idea which was published much later in [Lloyd \(1982\)](#). [Forgy \(1965\)](#) also came up with the same method except the difference from Lloyd was that he considered a continuous distribution instead of using the discrete distribution for the data. They have exactly the same procedure apart from this difference. [MacQueen et al. \(1967\)](#) for the first time used the term k -means and proposed an algorithm that is efficient for large data sets compared to the earlier two just mentioned. The [Hartigan and Wong \(1979\)](#) algorithm is widely applicable and is currently the default algorithmic choice in k -means' R implementation, available through the function "kmeans()" in the base package "stats".

2.4.2.1 The k -means algorithm

Let \mathcal{X} be n observations to cluster. Let the clustering produced by k -means be denoted by $\mathcal{C}_k = \{C_1, \dots, C_k\}$. Let m_1, \dots, m_k be the centroids of k clusters and $c_i = r$ be the label of an observation $x_i \in C_r$, for $i \in \mathbb{N}_n$ and $r \in \mathbb{N}_k$. Let n_r , $r = \{1, \dots, k\}$ denotes the number of observations in r -th cluster. Before running the k -means algorithm the users have to decide the distance measure, number of clusters, the method to chose the initial centroids, method to assign the points to centroids, and the iterative way in

²The English translation of the title of the paper is "On the division of material bodies" by Laurent Duval.

which the centres are to be updated. These issues are discussed after the algorithm is presented. The general steps of the k -means algorithm are given as follows:

The k -means algorithm

Initialize

Decide an initial set of centroids, i.e., input m_1, \dots, m_k .

Iterate

- (i) Assign cluster labels to each observation by selecting the closest centroid to it, i.e., chose c_1, \dots, c_n that minimize the following objective function:

$$\mathcal{M}(\mathcal{C}, m_1, \dots, m_k) = \sum_{i=1}^n \|x_i - m_{c_i}\|^2,$$

such that

$$c_i = \arg \min_{r \in \{1, \dots, k\}} \|x_i - m_r\|, \quad i \in \mathbb{N}_k.$$

- (ii) Compute the new cluster centroids by computing means $m_k = \frac{1}{n_r} \sum_{c_i=r} x_i$.

Stop iterating if (i) and (ii) do not change.

k -means algorithm needs an initial set of centroids to begin. It then iteratively improves the initial solution by recalculating centers after every cluster assignment. The final solution depends upon the initialization. That's why this method does not guarantee a unique solution. Various suggestions for efficient initializations methods have been suggested, for instance, see [Fayyad et al. \(1998\)](#) and [Celebi et al. \(2013\)](#). One such widely used method is random sampling of set of centroids from the data by [Forgy \(1965\)](#). To get well distributed centres across the data, the chosen centres should be located as far as possible. If data is divided into random parts equal to the desired number of centroids and then the means of these sampled parts are taken as initial points, the risk is that initial points will almost coincide. Therefore, it is recommended to do several random initializations. All such suggestions are capable of finding local optima only. If the initial points are not carefully chosen it can affect the stability of the algorithm, can take longer time to converge, and the solution may not be good.

For the k -means' algorithm the users have to decide the number of clusters to use. There are many k -means' algorithms. These algorithms differ in method used for the initialization of the starting clusters, and how the cluster centres should be updated. The clustering objective is to assign points to the clusters in such a way that the total within clusters sum of squares is minimum. Since mathematically the total within clusters sum of squares is minimum when means are used as centres, therefore each algorithm updates the centres by calculating arithmetic mean. In Lloyd/Forgy algorithm, the k centroids are decided by choosing k observations randomly from the data set. Each observation in the data is then assigned to closest centres to form clusters.

The cluster centres are recalculated once all the observations are assigned to the closest centres (see [Kaufman and Rousseeuw \(1987\)](#)). The selection of initial centriods for the MacQueen algorithm is same as that to the Lloyd/Forgy algorithm but the centroid update process is different.

MacQueen chooses a random sample of k points from the data to decide the initial centroids. It starts with the k groups each having one point in it and then starts adding observations from data to clusters. Each time an observation is added to a cluster, the center is updated. Thus here, the points are allocated to their closest centroid and the centres are updated each time a new assignment is made unlike the Lloyds/Forgy algorithm where the centroids are updated only once and that is at the end of assignment of all points to centres. The algorithm is stopped if centres do not change or if no point changes the cluster membership.

For the Hartigan and Wong algorithm, to decide an initial set of centres, the authors suggested the following method. Let k be the number of clusters to be found. Sort the data according to their distance from the overall mean of the data. They then take the sample of k observations from the ordered data, by applying k times, the sampling rule: $((j - 1)n/k + 1)$, where $j \in \{1, \dots, k\}$ and n is the total number of observation in the data. Assign each observation to their closest cluster center. Calculate the new centroids by taking mean of the assigned points. If a centroid is updated then the membership for the data points is decided by calculating the within-cluster sum of squares error (SEE).

k -means has certain limitations and some drawbacks. As the mean is sensitive to outliers so is k -means. It is limited to Euclidean distances only. It is also sensitive to the initialization method. The k -means optimization problem tries to optimize the variance of clusters globally but often terminates at a local optimum ([MacQueen et al., 1967](#)). However, the majority of the algorithms designed to optimized the k -means objective function will always converge at least to a local minimum and have linear time complexity.

Generally, clustering algorithms based on the partitioning principle will give different clustering solutions with several different initializations. These algorithms need to pre-specify the number of clusters k and the final solution depends upon k . Partitioning techniques might perform bad if points in a cluster are not close to the centroids/medoids of their own cluster but rather to the center of another cluster.

There are various modifications and extensions to k -means, as some people try to improve the limitations, others propose alternative methods. Some modifications aim at finding the better initilization methods, for instance, kmean++ proposed by [Arthur and Vassilvitskii \(2007\)](#). Some try to find the best number of clusters k , for instance ISODATA by [Ball and Hall \(1967\)](#). Some try to make k -means robust ([Cuesta-Albertos et al. \(1997\)](#)). k -medians is a simple modification of k -means where medians are used as centroids instead of means. K-modes by [Huang \(1997\)](#) provides an extension of k -means to categorical data by introducing a dissimilarity measure “simple matching

co-efficient” for categorical objects and a frequency - based method to update modes instead of means. Other attempts to extend similar ideas based on k-means to other clustering methods are fuzzy c-mean ([Ruspini \(1969\)](#)) for the soft cluster assignments rather than the hard ones.

2.4.2.2 The partitioning around medoids algorithm

The k-medoids clustering method was proposed by [Kaufman and Rousseeuw \(1987\)](#). It tries to find k representative members from the data set to reflect the structure of the data. To apply this method the authors have designed the program PAM (Partitioning Around Medoids) that consists of two phases, build and swap, available through the R package “cluster” ([Maechler et al. \(2017\)](#)). The build phase aims at obtaining the initial set of medoids by minimizing the average distance of objects from their representative object. The first representative is the most centrally located object in the data and the rest are selected iteratively. In the swapping phase this set of initial medoids is improved by replacing each selected medoid by a non selected object and minimizing the average distance for all potential medoids.

Partitioning around medoids (PAM) gives a partitioning of n objects in k clusters. Let \mathcal{X} be the n objects to be clustered and $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$ be some dissimilarity measure. PAM chooses a set of k medoids $\{m_1, \dots, m_k\} \in \mathcal{X}$ and assigns each object a cluster label $c_i^r; i \in \mathbb{N} = 1, \dots, n; r \in \mathbb{N} = 1, \dots, k$ for objects in \mathcal{X} in such a way that the following objective function is minimized:

$$\mathcal{P}(\mathcal{C}, m_1, \dots, m_k) = \sum_{i=1}^n d(x_i - m_{c_i^r}).$$

The [Kaufman and Rousseeuw \(1990\)](#)’s implementation of PAM algorithm has been used in this work. PAM works with both a data matrix and a proximity matrix. It is flexible in terms of defining closeness, as it is not based on any specific distance metric. The computational complexity of PAM is $O(k(n-k)^2)$ which makes it very slow for large data sets. To make PAM efficient for large data sets much effort has been done, for instance, CLARA (Clustering LARge Applications) by the authors of PAM. CLARA works for Euclidean data and takes several small samples of data and computes medoids. The data set is applied to the resulting medoids for each sample. The best solution among these is picked according to the objective function. Lucasius et al. (1993) have noticed that CLARA performs poorly as the numbers of clusters increases.

PAM has gained immense popularity and has been applied to many clustering problems in various fields. Some studies have compared the performance of PAM with other methods, see [Reynolds et al. \(2006\)](#). Various attempts have been made towards making PAM faster by modifying initilization of PAM in a k-means manner, for instance, see [Park and Jun \(2009\)](#). Some other attempts include, but are not limited to,

The PAM algorithm

Let \mathcal{X} be set of n objects to cluster into k groups. Let M is set of tentative selected medoids. The *Build* phase choose the members m_1, \dots, m_k of the set M . Let $C = \mathcal{X} - M$ be the set of unselected objects which contains candidate for inclusion in M . Let D_j be the dissimilarity between object x_j and the closest object in M .

Build (Choose k representative points for k clusters)

- (i) Calculate for $i \neq h$, $d(x_i, x_h), (i, h) \in \mathbb{N}_n$.
- (ii) Select the object as the first medoid that is most centrally located in the data i.e., choose that point as m_1 that gives
$$m_1 = \arg \min_{x_h \in \mathcal{X}} \sum_{i=1}^n d(x_i - x_h).$$
- (iii) Set $q = 2$. Consider $i \in C$ as a candidate for the next medoid to be included in M .
- (iv) For an object $j \in C - \{i\}$, compute D_j .
- (v) Let $E_{ji} = \max\{D_j - d(j, i), 0\}$.
- (vi) Compute the total gain obtained by adding i to M as $G_i = \sum_{j \in C} E_{ji}$.
- (vii) Choose the object i that maximizes G_i .
- (viii) Let $M = M \cup \{i\}$, $C = C - \{i\}$, and $q = q + 1$.
- (ix) Stop when $q = k$.

Swap (Improve the value of objective function, if possible)

- (i) $q = 1$. Replace each of $\{m_1, \dots, m_k\}$ with a non-selected medoid x_i (i.e., now consider every object in the data as a medoid other than $\{m_1, \dots, m_k\}$) and denote the new set of medoids by $\{m_1^*, \dots, m_k^*\}$.
 - (ii) For each of the pairs (x_i, m_r) , where $x_i \notin \{m_1^*, \dots, m_k^*\}$, compute $\mathcal{P}_{(i,r)}^* = \mathcal{P}(\mathcal{C}_{(i,r)}, m_1^*, \dots, m_k^*)$, where $\mathcal{C}_{(i,r)}$ represents the optimal assignment of x_i to the closest medoid m_r^* .
 - (iii) $\mathcal{P}^{(q)} = \arg \min_{(i,r)} \mathcal{P}_{(i,r)}^*$.
 - (iv) If $\mathcal{P}_{(i,r)}^{(q)} \leq \mathcal{P}_{(i,r)}^{(q-1)}$, it means that the objective function can be further improved with this swap. Replace m_r^* with object x_i and go to (i). If $\mathcal{P}_{(i,r)}^{(q)} \geq \mathcal{P}_{(i,r)}^{(q-1)}$, no further improvement in the objective function can be made. Stop.
-

application of PAM to time series, financial, medical or spatial data sets for instance, see a survey by [Rani and Sikka \(2012\)](#).

2.4.3 Model-based clustering methods

Let us have n observations which consist of k different sub-populations. Let the observations be p dimensional. Let the i^{th} observation belong to the r^{th} sub-population which has the density: $f_r^*(x; \theta_r)$, where θ_r are the unknown parameters of interest. Let λ_r for $r = 1, \dots, k$ be the mixture parameters. Let $\varphi(x; \tau)$ be the mixture density model for this population, where $\tau = (\lambda, \theta)$ are the parameters of the mixture model. The multivariate mixture density can be written as:

$$\varphi(x; \tau) = \sum_{r=1}^k \lambda_r f_r^*(x; \theta_r), \quad (2.2)$$

where $\lambda_r \in [0, 1]$ and $\sum_{r=1}^k \lambda_r = 1$, f_r^* is the density of the r^{th} component in the mixture. The likelihood for the mixture model given in (2.2) with k components for x_1, \dots, x_n observations in \mathbb{R}^p can be written as follows:

$$\ell(\tau|x) = \prod_{i=1}^n \sum_{r=1}^k \lambda_r f_r^*(x_i; \theta_r).$$

$f_r^*(x; \theta)$ are often taken as k multivariate Gaussian densities g_r with $\theta = (\mu_r, \Sigma_r)$, given as follows:

$$g_r(x|\mu_r, \Sigma_r) = \frac{\exp\{-\frac{1}{2}(x_i - \mu_r)^t \Sigma_r^{-1} (x_i - \mu_r)\}}{(2\pi)^{p/2} |\Sigma_r|^{1/2}}. \quad (2.3)$$

The parametric vector τ can now be fully written as

$\tau = (\lambda_1, \dots, \lambda_{k-1}, \mu_1, \dots, \mu_k, \Sigma_1, \dots, \Sigma_k)$. The maximum likelihood estimation of the parametric set τ is usually obtained through the expectation maximization (EM) algorithm under some constraint for covariance matrix Σ_r . In case of multivariate Gaussian densities the clusters are centered at μ_r with ellipsoidal densities. These ellipsoidal structures of the clusters are determined by three geometric features shape, volume and orientation defined by the covariance matrix Σ_r . Various parametrization for Σ_r exist in literature allowing to vary only one, two or all of three features. One way to allow each cluster's shape, volume and orientation to differ is $\Sigma_r = \alpha_r O_r W_r O_r^T$ where α_r is a scalar and determines the volume of r^{th} ellipsoid, O_r is the orthogonal matrix consisting of the eigenvectors of Σ_r which determines the orientation and W_r is the diagonal matrix with elements proportional to the eigenvalues of Σ_r , which determines the shapes of the ellipsoids (density contours). For more detailed decomposition and geometric interpretation of a list of options available for Σ_r see [Scrucca et al. \(2017\)](#), where the authors have currently implemented ten covariance structures in the R package “mclust” ([Scrucca et al. \(2017\)](#)).

How many mixture components should be used to fit the data and which of the covariance structures fit the data best can be handled as a model selection problem

([Dasgupta and Raftery \(1998\)](#)). This can be done either by using Bayes factors or using the Bayesian information criterion, see for instance [Jeffreys \(1935\)](#) and [Kass and Raftery \(1995\)](#).

Without having the cluster membership of the observations, the data is assumed to be incomplete. Let x_i be in one of the k clusters represented by z_{ir} such that

$$z_{ir} = \begin{cases} 1, & x_i \in r^{th} \text{cluster} \\ 0, & \text{otherwise.} \end{cases}$$

z_{ir} is known as missing data and needs to be initialized first while estimation. Let $y_i = (x_i, z_i)$ represent the complete data. The *log* likelihood for the complete data denoted by ℓ_c can be written as follows:

$$\ell_c(\theta_r, \lambda_r | z_{ir}, y) = \sum_{i=1}^n \sum_{r=1}^k z_{ir} \log(\lambda_r f_r^*(x_i; \theta_r)). \quad (2.4)$$

After choosing the constraint on the covariance matrix and by fixing the number of components, the maximum likelihood estimation of the mixture model defined in (2.4) using the Gaussian mixture model given in (2.3) for the incomplete data is usually done by the EM algorithm ([Dempster et al. \(1977\)](#)). The EM algorithm approximates the model parameters when some data is missing. It iterates between the E and the M step. It first takes the initial values for z_{ir} and performs the M step. The EM algorithm for maximum likelihood estimation for multivariate Gaussian mixtures is given as follows:

The EM algorithm for model-based clustering

Set $q = 0$

Initialize $z_{ir}^{(q)}$

Iterate

M-step:

Given $z_{ir}^{(q)}$, compute

$$n_r^0 = \sum_{i=1}^n z_{ir}, \lambda_r^{(0)} = \frac{n_r}{n}, \mu_r^{(0)} = \frac{1}{n_r} \sum_{i=1}^n z_{ir} x_i,$$

where estimation of Σ_r depends upon the model, see ([Celeux and Govaert \(1995\)](#)).

E-step:

Given the MLEs of parameters, compute $z_{ir}^{(q)}$

$$z_{ir}^{(q)} = \frac{\lambda_r f_r^*(x_i | \mu_r, \Sigma_r)}{\sum_{j=1}^k \lambda_j f_j^*(x_i | \mu_j, \Sigma_j)},$$

where $f_r^*(x_i | \mu_r, \Sigma_r)$ is replaced by g_r as defined in (2.3).

$q = q + 1$

Stop

when convergence criteria are met.

For the initialization of z_{ir} one can use the model-based AHC suggested by [Fraley and Raftery \(1998\)](#) and implemented in R package “mclust” ([Scrucca et al. \(2017\)](#)). The model-based AHC starts with every object in its own cluster. From n clusters it then starts to merge the two closest clusters in one cluster, based on some similarity among them by optimizing the classification likelihood function. Let $c_n^r = (c_1, \dots, c_n)$ be the labels for x_i and the r -th sub-population is $f_{c_i}^*(x_i|\theta)$. Then θ and c_r is choosen to maximize $\ell_{cl}(\theta, c|x_i)$ defined as follows:

$$\ell_{cl}(\theta, c|x_i) = \prod_{i=1}^n f_{c_i}^*(x_i|\theta_{c_i}), \quad (2.5)$$

where $c_i = r$ (if x_i is assigned to r^{th} cluster), is the classification of the data and $f_r(x_i)$ is defined in (2.3). To terminate the EM algorithm some criterion is needed. For instance, the algorithm can be stopped if there is no significant increase in the likelihood function.

Presented here is just the main idea, very briefly of a very vast topic. Note that there are many other criteria available for model selection, techniques for mixture model estimation and various versions of EM algorithms. Model-based clustering has been well explored by many authors and been applied in various disciplines, for instance, see [Banfield and Raftery \(1993\)](#), [Celeux and Govaert \(1995\)](#), [Fraley and Raftery \(1998\)](#), [Yeung et al. \(2001\)](#), [Dortet-Bernadet and Wicker \(2007\)](#), [Neumann et al. \(2008\)](#) and [Scrucca and Raftery \(2015\)](#). For a review on model-based clustering methods in context of their limitations for the high-dimensional data sets readers are refer to see [?](#).

2.4.4 Spectral clustering methods

Spectral clustering can be viewed as an approach to partition similarity based graphs. It is based on a connectivity concept to cluster data, rather than on compactness. It clusters the data with the notion that intra-cluster similarity should be high and inter-cluster similarity should be low. It does not apply any specific assumptions on the clusters a priori and can find non-convex clusters. To have a comprehensive overview on this method, [Von Luxburg \(2007\)](#), [Filippone et al. \(2008\)](#) and ([Meila, 2015, §7](#)) can serve as good reference points to start.

Let \mathcal{X} be the n data points to be clustered. From the data matrix an undirected,³ weighted⁴ graph is constructed. Let $\mathcal{G} = (V, E, A)$ be the similarity graph (also known as network) where V represents set of vertices (also known as nodes) of the graph and each vertex v_i in V represents a data point x_i . The edges (or links) between two vertices v_i and v_j represent similarity between them and are represented by set E . Thus, the graph \mathcal{G} is also called the similarity graph. The adjacency is an $n \times n$ matrix A , whose

³In an undirected graph the edges between each pair of vertices are bidirectional.

⁴In a weighted graph each edge has an associated weight attached to it.

entries contain the weights of the edges of the graphs connecting vertices i and j . If a_{ij} represents the elements of matrix A then the adjacency between two points can be defined as

$$a_{ij} = \begin{cases} \varphi(x_i, x_j), & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases}$$

The function φ can be any similarity measure between pairs of data points. To built the adjacency matrix, several options for φ are available such as the Gaussian kernel $\exp\left(-\|x_i - x_j\|^2/(2\sigma^2)\right)$ where $\|x_i - x_j\|^2$ is the Euclidean distance between the observations i and j . There are also different types of similarity graphs available, for instance, the ε -neighbourhood graph or the n -nearest neighbour graphs. The similarities a_{ij} are used as the weights of edges between pairs of vertex i and j . The elements of A are non-negative i.e., we assume each edge between two vertices has a non-negative weight $a_{ij} \geq 0$. If $a_{ij} = 0$ this means the vertices v_i and v_j are not connected through an edge. To perform clustering the aims is to partition the similarity graph such that the edges between different groups have low weights and edges within a group have high weights. The partitioning of set V of a graph into k subsets will define clusters and is known as graph cut. To separate a subset C from the set of vertices V from the complementary set $\bar{C} = V \setminus C$ an optimization function is needed to perform a graph cut. The value of cut between two subsets C and \bar{C} of V is defined as follows

$$\text{cut}(C, \bar{C}) = \sum_{i \in C} \sum_{j \in \bar{C}} a_{ij}.$$

The optimal partitioning into two subsets is determined by minimizing the cut value. The cut can be extended to k subsets. To obtain a k groups partition $\mathcal{C}_k = (C_1, \dots, C_k)$, where $C_i \cap C_j = \emptyset$ and $\cup_{r=1}^k C_r = V$, one can use the following cut.

$$\text{cut}(C) = \frac{1}{2} \sum_{r=1}^k \text{cut}(C_r, \bar{C}_r).$$

This is the simplest function to optimize the cut and is known as minmax cut ([Ding et al. \(2001\)](#)). The minmax cut is prone to find singleton clusters or prefers to cut small sets of isolated nodes in a graph more often, see Section 5 in [Von Luxburg \(2007\)](#), therefore other definitions can be used to avoid this problem. For instance normalized cut ($Ncut$) by [Shi and Malik \(2000\)](#) can be used. For formally defining this cut we need to define the degree matrix and the volume of a cluster. The degree of a vertex $v_i \in V$ is defined as:

$$\deg(v_i) = \sum_{j \in V} a_{ij}.$$

From here a diagonal matrix having degrees $\deg(v_1), \dots, \deg(v_n)$ on diagonals can be defined as the degree matrix \tilde{D} .

One among the other possible definitions to measure size of C can be defined by adding the weights of all the edges associated to the vertices in subset C . Symbolically we can write it as follows:

$$vol(C) \equiv \sum_{i \in C} \deg(v_i).$$

The normalized cut ($Ncut$) is defined as follows:

$$Ncut = \sum_{r=1}^k \frac{cut(C_r, \overline{C}_r)}{vol(C_r)}, \text{ where } \overline{C}_r = V \setminus C_r,$$

or the ratio cut by [Hagen and Kahng \(1992\)](#) can be used with the difference from the above in the definition of size of the subset of V i.e., replace the volume of subset $vol(C_r)$ by its size. Ratio cut uses $|C_r|$ in the denominator instead of $vol(C_r)$, where $|C|$ represents the number of vertices in C . Several other functions are available to perform graph cut for instance see [Dhillon et al. \(2004\)](#).

Spectral clustering can be divided into two main streams. One that is just defined above which takes the approach of first separating the whole large graph into two small pieces by removing the connected edges in graph. It then recursively minimize cut on exiting segments to extend the bi-partitioning to k-partitions. A limitation of this method is that the minimizing task for the $Ncut$ or ratio cut is the NP-hard⁵ discrete graph partitioning problem⁶. Alternatively, a relaxation to NP-hard can be made by defining the semi-optimal cut using the graph's Laplacian. As mentioned in [Von Luxburg \(2007\)](#), relaxing $Ncut$ leads to normalized spectral clustering while relaxing ratio cut leads to un-normalized spectral clustering.

One can use the Laplacian matrix instead of using the adjacency matrix A directly. A Laplacian matrix is a matrix defined on relationships of the adjacency matrix, and clustering can be performed using the eigenvectors of this matrix. There are many forms of this matrix. For instance two are given as follows:

The normalized Laplacian: $L = I - \tilde{D}^{-\frac{1}{2}} A \tilde{D}^{-\frac{1}{2}}$

⁵Different decision tasks have different complexity classes. These are P, NP, NP-complete and NP-hard. P in language of algorithm complexity means that the problem is solvable in “polynomial time” on a Turing machine whereas NP stands for “non-deterministically polynomial time” meaning that the problem can't be solved in polynomial time but can be verified in polynomial time on a Turing machine. Many NP problems are solvable by transforming them to P class, however, it is not known whether each problem in the class of NP can be solved, i.e., are reducible into P class. A problem is NP-complete if it can be transformed into polynomial time (P) problem. A problem is NP-hard when it can be polynomially reduced from an NP-complete problem, but it is not known whether it belongs to NP.

⁶A discrete partitioning problem is one where the solution vector can only take one out of two possible values. In the relaxation of ratio cut this restriction is waived off such that the solution vector can take any value. This relaxation leads to the un-normalized graph Laplacian.

The un-normalized Laplacian: $L = \tilde{D} - A$.

Their properties and discussions on them can be found in [Mohar et al. \(1991\)](#) and [Chung \(1997\)](#). The algorithm by [Ng et al. \(2001\)](#) has been used in this work.

The Normalized Laplacian algorithm

- (i) Construct the Adjacency matrix $A \in \mathbb{R}^{n \times n}$ using the Gaussian similarity function.
 - (ii) Construct the degree matrix \tilde{D} and the normalized Laplacian matrix $L = I - \tilde{D}^{-\frac{1}{2}} A \tilde{D}^{-\frac{1}{2}}$.
 - (iii) Compute the first k eigenvectors w_1, \dots, w_n of L and define a new matrix $W \in \mathbb{R}^{n \times k}$ by storing the vectors w_1, \dots, w_n in the columns of W .
 - (iv) Construct a matrix Y from matrix W by normalizing rows of W to norm 1, i.e., apply the transformation $y_{ij} = w_{ij}/(\sum_j w_{ij}^2)^{1/2}$.
 - (v) For $i = 1, \dots, n$, let rows of Y : $y_1, \dots, y_n \in \mathbb{R}^k$. Cluster the rows of Y into k groups by applying k -means algorithm.
 - (vi) Assign the cluster memberships to observations $i \in \mathcal{X}$ corresponding to the cluster membership of row i of matrix Y .
-

The computational complexity of the spectral clustering algorithm is high, which makes it unsuitable for datasets having values in the thousands. The complexity for the construction of similarity graph is $O(n^2)$, computation of the eigenvalues of Laplacian matrix is $O(n^3)$ and the k -means application to rows of normalized matrix W for the eigenvalues decomposition is $O(npks)$, where n is number of data points, p is number of variables, k is number of clusters and s is number of iteration taken by k -means to converge.

For other views and algorithms of spectral clustering see [Meila and Shi \(2000\)](#) and [Shi and Malik \(2000\)](#). There have been some attempts in the literature to automatically find the number of groups in spectral clustering by analysis the eigenvalues of the adjacency matrix, see for instance [Zelnik-Manor and Perona \(2004\)](#).

2.5 Definition of validation indices

In this work we use validation indices in three ways. Firstly, to check how well the clustering fits the data. For this we will use external indices. These indices measure how well the two clusterings match. Since for the synthetic datasets the data generating models are known, we can use these labels to validate clustering obtained from the proposed method. Also there are many real data sets for which there exists a consensus of researchers or experts from the fields on the correct data partitioning, which can

serve as a benchmark. Secondly, internal validation indices are used to categorize the proposed algorithms. For instance, if an index is based on the within clusters variation concept (or other definitions like separation or compactness), calculating this index for the clustering can inform about which among the considered clustering methods can give best separated or compact or homogeneous clusters. Lastly, we have used several existing indices for the estimation of the number of clusters to compare the estimated number of clusters from the proposed methods.

All the validation indices and methods for the estimation of number of clusters used in this work are reviewed in the following subsection.

2.5.1 Methods for estimation of the number of clusters

The problem of finding structure in the data go in hand with the problem of the estimation of the number of groups in the data. Various approaches have been proposed in the literature for the estimation of number of clusters, but it is still a difficult problem, as it heavily relies on the clustering methods used. In this section we will review the methods used in this study for the estimation of the number of clusters.

To estimate the number of clusters a randomness hypothesis can be tested against k clusters are present in the data ([Jain and Dubes \(1988b\)](#)). The null hypothesis in this case is that there is no structure present in the data, or these data is drawn completely at random from a certain distribution i.e., $k = 1$. Among the used null hypotheses the most common is the random position hypothesis. In other words this hypothesis tests for the spatial randomness of the data points. A way to ensure this is to state that the data is a random sample from a p -dimensional Gaussian distribution (see [Jain and Dubes \(1988b\)](#) for more details on this).

It is expected that the null hypothesis of randomness will be rejected with high probability if the data contain clear clusters or non-random patterns. The evidence against the null hypothesis can be obtained by constructing test statistics based on information of the data (see [Jain and Dubes \(1988b\)](#)). Statistics based on characteristics of clustering like compactness, connectedness etc., formally known as internal indices, can also be used, for instance, within and between cluster sum of squares. One crucial thing in developing these statistics is to define a rejection criterion i.e., to set a threshold that decides whether the statistics/index is large or small enough to reject a hypothesis.

A common approach in this regard is to not formally test a null hypothesis, but rather to look for the optimal value of the statistics/index used. Let K denote the maximum number of clusters allowed while estimation. Then one way of estimating the number of clusters is to find \hat{k} , $1 \leq \hat{k} \leq K$ for $1 \leq K \leq n$ such that the optimum value, maximum or minimum depending upon the index used, is attained. We now define the indices used in our study to estimate the number of clusters, namely the Calinski and Harabasz index, the Hartigan index, the Gamma index, the C index, the Krzanowski

and Lai index, the Gap method, the Jump method, the prediction strength method, the bootstrap instability method, the CVNN index and the average silhouette width index. Another approach is based on estimating the number of clusters by modelling the data from Gaussian mixtures and then it estimates the number of components in a mixture through the BIC.

Let $\mathcal{X} = x_1, \dots, x_n$, be the n observations where each of the x_i is a column vector of length ' p ' representing p variables on the i^{th} observation, that is, $x_i = (x_{i1}, \dots, x_{ip})$. The data is assumed to be partition into k clusters. Let the integer vector c represent a partition of the data with values between 1 to k . The length of c is equal to the number of observations n . For each index $i \in \mathcal{X}$, the i^{th} entry in the vector c is such that $c_i = r$, ($1 \leq r \leq k$) indicates the cluster number to which each observation x_i is allocated. Let the clustering solution of the data be represented as \mathcal{C}_k . For each $C_r \in \mathcal{C}_k$, let n_r , $r = \{1, \dots, k\}$ be the number of observations in these clusters. Let μ me the overall mean vector of data of length p , where x_i are Euclidean is assumed. Let μ^r be the mean vectors of clusters C_r , $r = \{1, \dots, k\}$. Let Ω_r stand for the set of indexes corresponding to the observations which are belonging to cluster C_r .

The total dispersion matrix for clustering \mathcal{C}_k is $T_{\mathcal{C}_k}$ can be defined as follows:

$$T_{\mathcal{C}_k} = \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^t.$$

Since each x_i is a column vector as $(x_{i1}, \dots, x_{ip})_{p \times 1}$ and μ is $(\mu_1, \dots, \mu_p)_{p \times 1}$, therefore,

$$(x_i - \mu) = \left((x_{i1}, \dots, x_{ip}) - (\mu_1, \dots, \mu_p) \right) = (x_{i1} - \mu_1, \dots, x_{ip} - \mu_p)_{p \times 1},$$

such that

$$(x_i - \mu)(x_i - \mu)^t = \begin{bmatrix} x_{i1} - \mu_1 \\ \vdots \\ x_{ip} - \mu_p \end{bmatrix} \begin{bmatrix} x_{i1} - \mu_1 & \dots & x_{ip} - \mu_p \end{bmatrix}. \quad (2.6)$$

Note that the dimension of matrix given in (2.6) is $(p \times p)$ and there will be n of such matrices one for each $x_i \in \mathcal{X}$ to sum up to get a final $(p \times p)$ matrix for $T_{\mathcal{C}_k}$. The diagonal entries in this matrix is variances whereas the off-diagonals are co-variances such that the matrix is symmetrical. The total sum of squares can be obtained by taking the trace of this matrix. The trace of a matrix is calculated by taking the sum of the diagonal entries of a square matrix. The total dispersion is splitted as $T_{\mathcal{C}_k} = W_{\mathcal{C}_k} + B_{\mathcal{C}_k}$, where $W_{\mathcal{C}_k}$ and $B_{\mathcal{C}_k}$ are within and between clusters dispersions respectively.

The within clusters dispersion for a clustering \mathcal{C}_k represented by $W_{\mathcal{C}_k}$ can be defined as:

$$W_{\mathcal{C}_k} = \sum_{r=1}^k \sum_{i \in \Omega_r} (x_i - \mu^r)(x_i - \mu^r)^t. \quad (2.7)$$

Since $x_i = (x_{i1}, \dots, x_{ip})$ and $\mu^r = (\mu_1^r, \dots, \mu_p^r)$, such that

$$(x_i - \mu^r)(x_i - \mu^r)^t = \begin{bmatrix} x_{i1} - \mu_1^r \\ \vdots \\ x_{ip} - \mu_p^r \end{bmatrix} \begin{bmatrix} x_{i1} - \mu_1^r, \dots, x_{ip} - \mu_p^r \end{bmatrix}. \quad (2.8)$$

For a cluster $C_r \in \mathcal{C}_k$ having the number of observations n_r , there will be n_r matrices like the matrix given in (2.8) due to the sum $\sum_{i \in \Omega_r}$ giving dispersion for each cluster and finally there will be k of such matrices of dimensionality $(p \times p)$ due to the outer sum $\sum_{r=1}^k$ in (2.7). The within cluster sum of squares is obtained by taking trace of $W_{\mathcal{C}_k}$. The between clusters dispersion, represented by $B_{\mathcal{C}_k}$ can be defined as:

$$B_{\mathcal{C}_k} = \sum_{r=1}^k n_r (\mu^r - \mu) (\mu^r - \mu)^t.$$

Note that $B_{\mathcal{C}_k}$ is also $p \times p$ matrix.

2.5.1.1 The Calinski and Harabasz index

The [Caliński and Harabasz \(1974\)](#) index is defined as follows

$$CH_k = \frac{\text{tr}(B_{\mathcal{C}_k})/(k-1)}{\text{tr}(W_{\mathcal{C}_k})/(n-k)},$$

where “ tr ” represents trace of a matrix. Note that CH_1 is not defined and the index should be maximized for k .

2.5.1.2 The Hartigan index

The [Hartigan \(1975\)](#) index is defined as follows:

$$H_k = \left(\frac{\text{tr}(W_{\mathcal{C}_k})}{\text{tr}(W_{\mathcal{C}_{k+1}})} - 1 \right) / (n - k + 1).$$

The estimated number of cluster is the smallest $k \geq 1$ such that $H_k \leq 10$.

2.5.1.3 The Gamma index

There are $n(n-1)/2$ number of distinct pairs of points in the data \mathcal{X} . Similarly, for a single cluster C_r in a clustering \mathcal{C}_k there are $n_r(n_r-1)/2$ pairs of points where n_r represents the number of points in C_r . Let n_w represent the total number of pairs of

points within k clusters of \mathcal{C}_k , then

$$n_w = \sum_{r=1}^k \frac{n_r(n_r - 1)}{2}.$$

Let Ω_w represent the indices for the pairs of points that are in the same clusters and Ω_b represent the indices for the pairs of points that are not in the same cluster. Let $x_i \sim_{\mathcal{C}} x_{i'}$ represent the two points x_i and $x_{i'}$ are in the same cluster and $x_i \not\sim_{\mathcal{C}} x_{i'}$ otherwise. Let n_+ represent the count for the distance between two points in the same cluster ($d(x_i, x_{i'})$ such that $x_i \sim_{\mathcal{C}} x_{i'}$) is smaller than the distance between the two points in different clusters ($d(x_i, x_{i'})$ such that $x_i \not\sim_{\mathcal{C}} x_{i'}$). Let n_- represent an opposite situation to n_+ , i.e., it gives the count for the total number of times for which the distance between the points belonging to same cluster is higher than the pair of points belong to different clusters. We can write these as follows:

$$n_+ = \sum_{(h, h') \in \Omega_b} \sum_{(i, i') \in \Omega_w} 1_{\{d(x_i, x_{i'}) < d(x_h, x_{h'})\}},$$

and

$$n_- = \sum_{(h, h') \in \Omega_b} \sum_{(i, i') \in \Omega_w} 1_{\{d(x_i, x_{i'}) > d(x_h, x_{h'})\}}.$$

Note that the ties are ignored, i.e., the case $d(x_i, x_{i'}) = d(x_h, x_{h'})$ where $x_i \sim_{\mathcal{C}} x_{i'}$ and $x_h \not\sim_{\mathcal{C}} x_{h'}$ is excluded. The Gamma index proposed by [Baker and Hubert \(1975\)](#) for a clustering \mathcal{C}_k is defined as

$$\text{Gamma}_k = \frac{n_+ - n_-}{n_+ + n_-}.$$

The index value ranges between 0 and 1 and should be maximized over k .

2.5.1.4 The C index

The C index proposed by [Hubert and Schultz \(1976\)](#) depends upon the distances between the pairs of points within each cluster. Let D_w denote the sum of within cluster distances, more formally as follows:

$$D_w = \sum_{x_i \sim_{\mathcal{C}} x_{i'}} d(x_i, x_{i'})$$

Let D_{min} denote the sum of the smallest distance between pair of points from each cluster. Note that if there are k clusters this will be sum of k distances, one from each cluster, such that a distance is chosen between two objects that is smallest in a cluster. Let D_{max} denote the complementary of D_{min} i.e., the sum of maximum distances between a pair of objects (one from each cluster) within each cluster. The C index can be

defined as follows:

$$C_k = \frac{D_w - D_{min}}{D_{max} - D_{min}}.$$

The index lies between $[0, 1]$ and the best k is the one that gave minimum value of the index.

2.5.1.5 The Krzanowski and Lai index

The Krzanowski and Lai (1988) index is defined as

$$KL_k = \left| \frac{diff_k}{diff_{k+1}} \right|,$$

where

$$diff_k = (k-1)^{2/p} W_{\mathcal{C}_{k-1}} - k^{2/p} W_{\mathcal{C}_k}.$$

The index is not defined for $k = 1$ and should be maximized over k .

2.5.1.6 The Gap method

Tibshirani et al. (2001) take the approach of testing a null hypothesis of no clustering versus the alternative hypothesis of k clusters in the data. They do so for $k = \{2, \dots, K\}$ clusters to find the optimal k for the data. For the observed data $\mathcal{X}_{n \times p}$, starting from $k = 2$ clusters, perform the clustering using any clustering method and calculate the within cluster sum of squares $tr(W_{\mathcal{C}_k})$ as defined in (2.8). The Gap statistics is defined as under:

$$\text{Gap}_k = \mathbb{E}_n \left(\log(tr(W_{\mathcal{C}_k})) \right) - \log(tr(W_{\mathcal{C}_k})),$$

where \mathbb{E}_n is the expectation of sample of size n under the reference distribution.

For the null hypothesis of no clusters present in the data, a null reference distribution is needed. For this, generate the data for p variables in the data over the range of the observed p variables. Let X'_1, \dots, X'_p denote the p variables in \mathcal{X} . Note that each of X'_j , $j = \{1, \dots, p\}$ is a vector of length n . Let $min_{\{j\}}$ and $max_{\{j\}}$ denote the minimum and maximum values of X'_j , $j = \{1, \dots, p\}$ respectively. Then draw p reference variables over the range of observed variables as $R_j \sim \mathbb{U}(min_{\{j\}}, max_{\{j\}})$ of size n , where \mathbb{U} is the continuous Uniform distribution. Let the reference data be drawn from the reference distribution be denoted as $\mathcal{R} = (R_1, \dots, R_p)$. The next step is to generate several reference data sets say M . For each of the M reference data sets perform a clustering from any clustering method (denote the resulting clustering by $\tilde{\mathcal{C}}_{km}$) and compute the within cluster dispersion matrix. Let $W_{\tilde{\mathcal{C}}_{km}}$ denote the within cluster dispersion matrix, i.e., the trace of this matrix will give within cluster sum of squares for each replicate of the reference data i.e., $m = \{1, \dots, M\}$, and for $k = \{2, \dots, K\}$. Compute the estimated

Gap statistic as follows:

$$\widehat{\text{Gap}}_k = \frac{1}{M} \sum_{m=1}^M \log(\text{tr}(W_{\tilde{\mathcal{C}}_{km}})) - \log(\text{tr}(W_{\mathcal{C}_k})).$$

The purpose is to estimate $\mathbb{E}_n \left(\log(\text{tr}(W_{\mathcal{C}_k})) \right)$ by an average $\log(\text{tr}(W_{\tilde{\mathcal{C}}_k}))$ over the M replicates of datasets from the reference distribution.

To evaluate the sampling distribution of the Gap statistics, the standard error is needed over the M replicates, which is $(M+1)/M$ times the standard deviation of $W_{\tilde{\mathcal{C}}_{km}}$ calculated from the $m = \{1, \dots, M\}$ replicates for fix k . Let the standard deviation of $\log(W_{\tilde{\mathcal{C}}_k})$ from the M replicates be denoted by sd_k , then the standard error (se) is given by $se_k = sd_k \sqrt{1 + 1/M}$. Choose the number of clusters \hat{k} as the smallest of $k \geq 1$ such that $\widehat{\text{Gap}}_k \geq \widehat{\text{Gap}}_{k+1} - se_{k+1}$.

2.5.1.7 The Jump method

[Sugar and James \(2003a\)](#) introduced the jump method based on distortion, idealized from information theory approach used for the compression of data in engineering. The idea is to minimize the Mahalanobis distance between cluster centres and the data points allocated to these centres. Let $x_i, i \in \mathbb{N}_n$ be a p -dimensional random variable. The distortion can be defined by assuming k components Gaussian mixture on data or a non-Gaussian distribution such as the uniform distribution on p -dimensional clusters to generalize the approach for other data settings. Let the data be from a k component Gaussian mixture each with covariance matrix Σ , and we want to fit a k cluster model with candidate cluster centres v_1, \dots, v_k . Let v_{x_i} be the closest cluster center to x_i . The distortion for the k -component model w.r.t. the data can be calculated as follows:

$$\delta_k = \frac{1}{p} \underset{v_1, \dots, v_k}{\arg \min} \mathbb{E}[(x_i - v_{x_i})^t \Sigma^{-1} (x_i - v_{x_i})].$$

Let K be the maximum number of clusters to fit the data. For $1 \leq k \leq K$. To estimate δ_k [Sugar and James \(2003a\)](#) proposed to apply the k -means clustering algorithm to the data to calculate the estimated distortion $\hat{\delta}_k$. Calculate the ‘jump’ as $Jump_k = \hat{\delta}_k^{-Y} - \hat{\delta}_{k-1}^{-Y}$, where Y is an transformation power for dataset that captures the relationship between clusters and distortion and should be greater than zero. It is hard to decide the power and it is recommended to check various values. [Sugar and James \(2003a\)](#) provide a somewhat detailed discussion on this. Choose the number of clusters such that $\hat{k} = \text{argmax}_{k \geq 1} Jump_k$.

The setting of $Jump_k$ is such that if there are no clusters in the data, the ‘jump’ should choose $k = 1$. Another way of choosing the number of clusters is to see the point where the distortion curve (formed by plotting $\hat{\delta}_k$ verse k) flattens off. The curve

is expected to decrease monotonically and one should expect this decrease to be very slow for k greater than the true number of clusters.

2.5.1.8 The prediction strength

Proposed by [Tibshirani and Walther \(2005\)](#) the prediction strength is a way to estimate the optimal number of clusters based on clustering stability. Let the data to be clustered be called the training data \mathcal{X}_{tr} having n independent observations over p variables of interest. Cluster this data by some clustering function into k clusters., and get the clustering solution denoted as $\mathcal{C}_{(tr,k)}$. Call the cluster centres obtained from training set clustering as k training set cluster centres. From this clustering define a 0-1 matrix Ψ that determines whether or not each pair of observations in \mathcal{X}_{tr} is together in a cluster or not in $\mathcal{C}_{(tr,k)}$. Let $C_{(tr,r)}$ represents the r^{th} cluster for $r \in \{1, \dots, k\} = \mathbb{N}_k$ in clustering $\mathcal{C}_{(tr,k)}$. For two distinct observations $(x_i, x_{i'}) \in \mathcal{X}_{tr}$, let $x_i \sim_{\mathcal{C}_{(tr,k)}} x_{i'}$ refer to the case that x_i and $x_{i'}$ are in same cluster $C_{(tr,r)}$ of a clustering $\mathcal{C}_{(tr,k)}$. Let $x_i \not\sim_{\mathcal{C}_{(tr,r)}} x_{i'}$ represent the complementary case. For all $(x_i, x_{i'}) \in \mathcal{X}_{tr}$ the matrix Ψ of dimensions $n \times n$ can then be defined as follows:

$$\Psi_{ii'}[\mathcal{C}_{(tr,k)}, \mathcal{X}_{tr}] = \begin{cases} 1, & \text{if } x_i \sim_{\mathcal{C}_{(tr,k)}} x_{i'}, \\ 0, & \text{otherwise.} \end{cases}$$

[Tibshirani and Walther \(2005\)](#) refer to the entries of Ψ as co-memberships. Let an independent sample of size m from p variables taken from the same population from which the training sample is drawn is also available. Let this sample be called the test data denoted by \mathcal{X}_{te} . Cluster the test data into k clusters and denote the clusterings as $\mathcal{C}_{(te,k)}$. Call these clusters as the k test clusters. Let $n_r = |C_{(te,r)}|$ for $r \in \{1, \dots, k\}$. Let $\Omega_{(te,r)}$ for $r \in \{1, \dots, k\}$ be the k sets of indices for the observations in k test clusters. Now classify the test data to the clusters of training data. Assign the observations of test data to one of the closest k cluster centres of training data. Call this classification of \mathcal{X}_{te} using the training centres $C^*(\mathcal{X}_{tr}, k)$. The idea for the prediction strength is to cluster the training data into k clusters and using these cluster centres classify the observations of test data to clusters again. Then measure for each pair of observations that are together in test clusters, determine whether were the training centres also assign them to same training centre.

The prediction strength for a k -clustering $ps(k)$ is defined as follows:

$$ps(k) = \min_{1 \leq r \leq k} \frac{1}{n_r(n_r - 1)} \sum_{\substack{i \neq i' \\ i, i' \in \Omega_{(te,r)}}} 1_{\{\Psi_{ii'}^*[\mathcal{C}^*(\mathcal{X}_{tr}, k), \mathcal{X}_{te}] = 1\}}.$$

One should choose the optimal k by maximizing the $ps(k)$ over k . The ps mea-

sures how well the k cluster centres obtained from the training set can predict the co-memberships for the observations in the test set.

In practice the population of the data is not known and hence the test set is not available. In this situation the authors have suggested one can split the data into two disjoint halves (2-folds) several times randomly and use first halves as a training set and other as the test set. Note that in this situation the final ps will be obtained by averaging the ps obtained from these several random splits of the data. The idea is if almost similar clustering is obtained every time from the samples of the data then the clustering is stable. For the stable clustering the training set and test set will be similar and ps for the clustering will be high.

2.5.1.9 The bootstrap instability method

Proposed by [Fang and Wang \(2012\)](#) the bootstrap instability method also defines the optimal number of clusters based on an instability measure calculated by taking several bootstrap samples from the data. Let we have a data \mathcal{X} having p -dimensional n observations. Generate 2 independent bootstrap samples of size n from \mathcal{X} , M times. Let the 2 samples is denoted as \mathcal{X}'_m and \mathcal{X}^*_m , $m = \{1, \dots, M\}$. Cluster the two sample data sets into k clusters and denote the resulting clusterings as $\mathcal{C}_{(\mathcal{X}'_m, k)}$ and $\mathcal{C}_{(\mathcal{X}^*_m, k)}$, for all m . The next step is to calculate the distance between the two clusterings obtained from \mathcal{X}'_m and \mathcal{X}^*_m . Before we formally define a distance between two clustering we need to define an indication function for a clustering. Let π be an indication function which determines the two observations are in same clusters or not i.e., it can either return the value 0 or 1. For any two distinct observations i and i' in \mathcal{X}' , π is defined as under:

$$\pi_{ii'} = \begin{cases} 1, & \text{if } c'_i = c'_{i'} \\ 0, & \text{if } c'_i \neq c'_{i'} \end{cases}$$

Let c' and c^* be two clustering vectors of length n representing clusterings on two samples \mathcal{X}' and \mathcal{X}^* respectively. Calculate the difference between the two clusterings $\mathcal{C}_{(\mathcal{X}'_m, k)}$ and $\mathcal{C}_{(\mathcal{X}^*_m, k)}$ as follows:

$$D(\mathcal{C}_{(\mathcal{X}'_m, k)}, \mathcal{C}_{(\mathcal{X}^*_m, k)}) = \frac{1}{n^2} \sum_{i=1}^n \sum_{i'=1}^n |\pi_{ii'}(c'_i = c'_{i'}) - \pi_{ii'}(c^*_i = c^*_{i'})|.$$

Note that the difference between the two indicator functions can return -2, -1, 0 or 1. The clustering instability measure is defined as follow:

$$\hat{S}_M(k) = \frac{1}{M} \sum_{m=1}^M D(\mathcal{C}_{(\mathcal{X}'_m, k)}, \mathcal{C}_{(\mathcal{X}^*_m, k)}).$$

The clustering stability is calculated for a range of numbers of clusters. Lets the maximum number of clusters tried were K . A smaller value of $\hat{S}_M(k)$ indicates a stable clustering. Chose the optimal number of clusters as: $\hat{k} = \arg \min_{2 \leq k \leq K} \hat{S}_M(k)$.

2.5.1.10 The cvnn index

The cvnn index proposed by Liu et al. (2013) is based on some kind of normalized intra-cluster compactness and inter-cluster separation, where both of these measures are defined differently then the others introduced previously. Let $\mathcal{C}_k = \{C_1, \dots, C_k\}$ represent a clustering on data $\mathcal{X}_{n \times p}$. Let n_r for $r \in \{1, \dots, k\}$ represent the number of observations in a cluster C_r from the clustering \mathcal{C}_k . Let the $Sep^*(\mathcal{C}_k, \phi)$ denote the inter cluster separation and $Com^*(\mathcal{C}_k)$ denotes the intra cluster compactness. Let i and i' be two distinct observations from a cluster C_r . The intra cluster compactness is defined as follows:

$$Com^*(\mathcal{C}_k) = \sum_{r=1}^k \left[\left(\frac{2}{n_r(n_r - 1)} \right) \sum_{i, i' \in C_r} d(x_i, x_{i'}) \right],$$

i.e., the compactness for a clustering can be defined as averages of the pairwise distances within clusters, and then a final sum over all clusters in a clustering.

Further assume that $\phi \in \mathbb{N}$ denotes the nearest neibours of a point, let v_i denotes the count for the nearest neighbors of an observation $x_i \in C_r$ that are outside of the cluster C_r , let n_r be the number of observations in C_r . Let i be an index representing the observations in cluster C_r then the inter cluster separation can be defined as:

$$Sep^*(\mathcal{C}_k, \phi) = \max_{r=\{1, \dots, k\}} \frac{1}{n_r} \sum_{i=1}^{n_r} \frac{v_i}{\phi}.$$

thus the inter cluster separation for a clustering is defined as choosing the one maximum value from the average of v_i over all clusters. Let $Sep(\mathcal{C}_k, \phi)$ and $Com(\mathcal{C}_k)$ be the normalized Sep^* and Com^* respectively, defined as follows:

$$Com(\mathcal{C}_k) = \frac{Com^*(\mathcal{C}_k)}{\max_k Com^*(\mathcal{C}_k)}, \quad Sep(\mathcal{C}_k, \phi) = \frac{Sep^*(\mathcal{C}_k, \phi)}{\max_k Sep^*(\mathcal{C}_k, \phi)}.$$

The cvnn index for a clustering \mathcal{C}_k , denoted by $cvnn(\mathcal{C}_k, \phi)$, is defined as follows:

$$cvnn(\mathcal{C}_k, \phi) = Sep(\mathcal{C}_k, \phi) + Com(\mathcal{C}_k).$$

The index should be minimized to find the optimal number of clusters i.e., $\hat{k} = \arg \max_{k=\{2, \dots, K\}} cvnn(\mathcal{C}_k, \phi)$.

The indices reviewed above have been used later in this work for comparison purposes to illicit the performance of the newly proposed methods, and also to compare

the performances of the already existing methods. A large number of indices proposed in literature is based on within and between clusters sum of square measures. Among these some are better known for their good performance as identified by comparison studies in the literature, for instance in [Milligan and Cooper \(1985\)](#), CH, Gamma and H index were the top performing indices. [Milligan and Cooper \(1985\)](#) did a study for the comparisons of the existing indices, but since then many other methods have been proposed in literature, especially after 2000, that are based on somewhat different ideas than the within and between cluster dispersions, like re-sampling using Monte Carlo or boot strapping strategies. Since [Milligan and Cooper \(1985\)](#), there has been no such big scale study to compare these new methods together with the previously proposed methods based on cluster dispersion idea. In this thesis a vast spectrum of indices have been included that are known for their good performance in the literature.

2.6 The average silhouette width

[Kaufman and Rousseeuw \(1990\)](#) proposed the Average Silhouette Width (ASW) to estimate the number of clusters using PAM. The silhouette width (SW) for an object in data represents how well the object fits in its present cluster. Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be the data set of size n and d be a distance function over \mathcal{X} and $\mathcal{C}_k = \{C_1, \dots, C_k\}$ a clustering identified by some clustering function f_k on \mathcal{X} . Let i represents the index for observations $x_i \in \mathcal{X}$. Let the clustering labels be represented in the standard column vector denoted by $(l(1), \dots, l(n)) \in \mathbb{N}_k$ determined by $l(i) = r$, $r \in \mathbb{N}_k$, $i \in \mathbb{N}_n$, where \mathbb{N}_k and \mathbb{N}_n set of natural numbers upto n and k . Let the cluster sizes are determined by $n_r = \sum_{i=1}^n 1(l(i) = r)$, $r \in \mathbb{N}_k$. For each objects $i \in \mathbb{N}_n$ calculate,

$$a(i) = \frac{1}{n_{l(i)} - 1} \sum_{\substack{l(j)=l(h) \\ i \neq h}} d(x_i, x_h), \quad (2.9)$$

and,

$$b(i) = \min_{r \neq l(i)} \frac{1}{n_r} \sum_{l(h)=r} d(x_i, x_h). \quad (2.10)$$

For a given clustering \mathcal{C}_k , the silhouette width for a data object having index i , $i \in \mathbb{N}_n$, is

$$S_i(\mathcal{C}_k, d) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad (2.11)$$

such that $-1 \leq S_i(\mathcal{C}_k, d) \leq 1$.

In other words, $a(i)$ is the average dissimilarity of object x_i from all the other objects in the cluster C_r , to which x_i belongs. $b(i)$ defines the minimum average distance of object x_i from all the objects in another cluster, except the cluster C_r of which x_i is

a member. $a(i)$ is not defined for singletons and to calculate $b(i)$ there should be at least two clusters. Therefore SW is only defined for $k > 1$. For a good clustering the “within” clusters dissimilarity should be less than the “between” clusters dissimilarity. Therefore, if $a(i)$ is much smaller than the smallest between clusters dissimilarity $b(i)$ we get evidence (larger $s(i)$, close to 1 is better in this case), that object x_i is in the appropriate cluster. On the other hand, $s(i)$ close to -1, points towards the wrong cluster assignment for object x_i . In this case $a(i) > b(i)$, meaning that object “ i ” is more close to its neighbouring cluster than to its present cluster. A neutral case occurs when $s(i) \approx 0$, i.e., object x_i is approximately equally distant from both, its present cluster and neighbouring cluster.

The SW averaged over all the members of a cluster can be used as a measure of a cluster’s quality. The ASW averages SW over all members of a data set \mathcal{X} . It is a global quality measure for clustering. Formally, for the clustering \mathcal{C}_k it can be written as follows:

$$\bar{S}(\mathcal{C}_k, d) = \frac{1}{n} \sum_{i=1}^n S_i(\mathcal{C}_k, d). \quad (2.12)$$

The best k can be selected by maximising $\bar{S}(\mathcal{C}_k, d)$ over k .

The ASW can be thought of as a combinational index because it is based on the two concepts which are separation and compactness that define a unified concept of isolation. It is a ratio of inter cluster variation and intra cluster variation. It measures how homogeneous the cluster are, and what the separation between them. Thus the ASW tells us about the coherent structure of clustering.

2.7 The PAMSIL algorithm

[Van der Laan et al. \(2003\)](#) have proposed a clustering method by optimizing the ASW based on medoids. They first run the PAM algorithm to get a clustering, i.e., they first choose a set of k medoids using the PAM build phase, and then consider all possible swaps (see swap phase of the PAM algorithm in Section 2.4.2.2) to further improve the values of objective function obtained in the build phase. At the end of the PAM algorithm, one gets a set of medoids and a clustering. Based on these medoids they then define an algorithm called PAMSIL to maximize the ASW. This algorithm is the same as the PAM swap phase, except that it does an extra step. After swapping each non-medoid to a medoid and assigning all the data points to these medoids based on minimum distance to define clustering, they then use this clustering to calculate the ASW value as well. At the end a set of medoid is chosen which gave the maximum value of ASW. Thus PAMSIL tries to find a clustering that maximize the ASW based on medoids.

PAMSIL algorithm

bwasp

Get a set of medoids from the PAM algorithm. Denote the medoids by $M_k = \{m_1, \dots, m_k\}$.

Set $q = 1$.

silswap

- (i) Replace every non-medoid object $i, i \in \mathbb{N}_n$ with one of the medoids, say $m_r \in M_k$. Represent the new set of medoids by $M_{(i,r)}^* = \{m_1^*, \dots, m_k^*\}$.
 - (ii) Assign each object to the closest medoid, i.e., $\arg \min_{i \in \mathbb{N}_n} d(x_i, M_k^*)$. Denote the resulting clustering by \mathcal{C}_k^* .
 - (iii) Calculate $f_{(i,r)} = \bar{S}(\mathcal{C}_k^*, d)$.
 - (iv) Assign $(h, s) = \arg \max_{(i,r)} f_{(i,r)}$, $f^{(q)} = f_{(h,s)}$ and $\mathcal{C}^{(q)} = \mathcal{C}^*$.
 - (v) Stop if $f^{(q)} \geq f^{(q-1)}$, else $q = q + 1$, repeat (i) - (v).
-

2.8 Clustering comparison measures

Clustering comparison measures are used to compare the different clusterings of the same data set. They are extensively in use to measure the agreement between two clusterings, usually as an external validation when the ground truth (usually the true labels corresponding to which the data has been generated) is known. The adjusted rand index (ARI) is one such measure to evaluate a clustering obtained from a method/algorith against a ground truth. In this work we have used ARI for the clusterings' evaluation. To calculate the ARI two clusterings are required. Usually, one of them is the external ground truth and other is the clustering result obtained from an algorithm one wishes to evaluate against ground truth. The use of ARI is not limited to this scenario and can be used in other ways for instance, not to compare a clustering with ground truth but instead to calculate the similarity for two clustering on a data set obtained from two different clustering methods.

2.8.1 Adjusted rand index

Let we have two clusterings as \mathcal{C}_k and \mathcal{C}'_q having k and q clusters respectively on the data set \mathcal{X} having n observations. Let $\{C_1, \dots, C_k\}$ and $\{C'_1, \dots, C'_q\}$ be the disjoint sets representing clusters for \mathcal{C}_k and \mathcal{C}'_q , respectively. Let $|C_r| = n_r$ for $r = 1, \dots, k$ and $|C'_h| = n'_h$ for $h = 1, \dots, q$ denotes the number of objects in clusters of the two clusterings \mathcal{C}_k and \mathcal{C}'_q , respectively. Naturally, $\sum_{r=1}^k n_r = n = \sum_{h=1}^q n'_h$. The agreement or similarity between two clusterings \mathcal{C}_k and \mathcal{C}'_q can be measure by counting the pair of

points of \mathcal{X} that has been assigned to same clusters in both clusterings. For a pair of points from \mathcal{X} one of the following four cases will always hold:

- n'_{11} : The number of pairs of points that are in the same clusters in \mathcal{C}_k and in the same cluster in \mathcal{C}'_q
- n'_{10} : The number of pairs of points that are in the same clusters in \mathcal{C}_k and in different cluster in \mathcal{C}'_q
- n'_{01} : The number of pairs of points that are in different clusters in \mathcal{C}_k and in the same cluster in \mathcal{C}'_q
- n'_{00} : The number of pairs of points that are in different clusters in \mathcal{C}_k and in different cluster in \mathcal{C}'_q

Using these four cases several measures for clustering similarity or dissimilarity has been proposed in literature. These four cases can be obtained from a contingency table. The contingency table is a $k \times q$ table for two clusterings \mathcal{C}_k and \mathcal{C}'_q whose rh -th element is the number of points in the intersection of clusters C_r of \mathcal{C}_k and C'_h of \mathcal{C}'_q , i.e., in both clusters C_r and C'_h . Let $n_{rh} = |C_r \cap C'_h|$, i.e., n_{rh} represents the number of object in both clusters C_r and C'_h . Let $n_{r\cdot}$ and $n_{\cdot h}$ represents the row sum and column sum of contingency table respectively. Thus n'_{11} and n'_{00} can be mathematically written as:

$$n'_{11} = \frac{1}{2} \sum_{r=1}^k \sum_{h=1}^q n_{rh}(n_{rh} - 1),$$

and

$$n'_{00} = \frac{1}{2} \left(n^2 + \sum_{r=1}^k \sum_{h=1}^q n_{rh}^2 - \left\{ \sum_{r=1}^k n_{r\cdot}^2 + \sum_{h=1}^q n_{\cdot h}^2 \right\} \right).$$

Note that $n'_{11} + n'_{10} + n'_{01} + n'_{00} = n(n-1)/2$. The Rand index porposed in [Rand \(1971\)](#) is given as:

$$RI(\mathcal{C}_k, \mathcal{C}'_q) = \frac{n'_{11} + n'_{00}}{n(n-1)/2}.$$

The problems with Rand index and other such related indices are well explored for instance refer to [Hubert and Arabie \(1985\)](#). One problem is that the expected value of the index is not constant. Another is, the number of pair of points for which the clusterings \mathcal{C}_k and \mathcal{C}'_q disagrees (n'_{00}) is often as large as $\binom{n}{2}$. An implication of this is that the index will approach its maximum value as number of clusters increases. This gives a false impression of closeness of two clusterings when in fact in reality two clusterings are further from each other. [Hubert and Arabie \(1985\)](#) proposed an adjustment to the Rand index. Let $\mathbb{E}(RI)$ represents the expected value of $RI(\mathcal{C}_k, \mathcal{C}'_q)$, then ARI is given as:

$$ARI(\mathcal{C}_k, \mathcal{C}'_q) = \frac{RI(\mathcal{C}_k, \mathcal{C}'_q) - \mathbb{E}(RI)}{1 - \mathbb{E}(RI)}.$$

Note that the expression $n'_{11} + n'_{00}$ can be simplified as $\sum_{r=1}^k \sum_{h=1}^q \binom{n_{rh}}{2}$ and

$$\mathbb{E}\left[\sum_{r=1}^k \sum_{h=1}^q \binom{n_{rh}}{2}\right] = \left[\sum_{r=1}^k \binom{n_r}{2} \sum_{h=1}^q \binom{n_h}{2}\right] / \binom{n}{2}.$$

Such that the *ARI* can be written as follows:

$$ARI(\mathcal{C}_k, \mathcal{C}'_q) = \frac{\sum_{r=1}^k \sum_{h=1}^q \binom{n_{rh}}{2} - \left[\sum_{r=1}^k \binom{n_r}{2} \sum_{h=1}^q \binom{n_h}{2}\right] / \binom{n}{2}}{\left[\sum_{r=1}^k \binom{n_r}{2} + \sum_{h=1}^q \binom{n_h}{2}\right] / 2 - \left[\sum_{r=1}^k \binom{n_r}{2} \sum_{h=1}^q \binom{n_h}{2}\right] / \binom{n}{2}}.$$

The index lies between 0 and 1 and a larger value indicates greater similarity between clusterings.

Chapter 3

The Optimum ASW Based Linkage Criterion

3.1 Preliminary notations

In this chapter we will introduce a new agglomerative hierarchical clustering method by introducing a new linkage criterion based on optimum average silhouette width.

Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be the data set to be partitioned, where x_i represents the i^{th} observation and each x_i represents a p -dimensional variable. We will here only consider crisp clustering, thus every object will belong to one cluster only and there will be no overlapping between clusters in the hierarchy. There will be n total hierarchy levels. Let k_1, \dots, k_n be the number of clusters in a clustering at each hierarchy level. Let the full hierarchy of \mathcal{X} be given by $\mathcal{P} = \{\mathcal{C}_n^1, \dots, \mathcal{C}_1^n\}$. The superscript in $\mathcal{C}_{k_l}^l \in \mathcal{P}$ represents the hierarchy level, where $l = 1, \dots, n$ and $k_l = n, (n-1), \dots, 2, 1$ denotes the number of clusters at each hierarchy level. In hierarchical clustering we start with n clusters in the beginning. Thus if l represents a particular hierarchy level, then at $l = 1$ we have $k_l = n$ clusters, i.e., each observation forms a separate cluster. The number of clusters subsequently reduce as hierarchy level proceeds. For simplicity assume that only one pair of clusters merges at each hierarchy level.

Let $\mathcal{C}_{k_l}^l = \{C_1^l, \dots, C_{k_l}^l\}$, where $C_r^l \in \mathcal{C}_{k_l}^l$, $r = 1, \dots, k_l$ represents an r -th cluster in a clustering at hierarchy level l . The members of a cluster at hierarchy level $l = 1$ can be further written as $C_1^1 = \{x_1\}, C_2^1 = \{x_2\}, \dots, C_n^1 = \{x_n\}$, thus $\mathcal{C}_1 = \{\{x_1\}, \{x_2\}, \dots, \{x_n\}\}$ and at the (n^{th}) final hierarchy level, $\mathcal{C}_1^n = \{x_1, x_2, \dots, x_n\}$, thus $\mathcal{C}_1^n = \mathcal{X}$. Let $\gamma^l(x_1, r), \dots, \gamma^l(x_n, r)$; $r = 1, \dots, k_l$ represent the clustering label vector at hierarchy level l . At a hierarchy level l , r indicates to which cluster observation x_i has been assigned.

Let **Hierarchical Optimum Silhouette width** agglomerative hierarchical clustering algorithm be called HOSil. The algorithm can't start from $l = 1$. This is because for the calculation of $a(i)$, $i \in C$ there should be at least one cluster in the clustering, with at

least two observations and for calculation of $b(i)$ there should be at least two clusters in a clustering solution. Therefore, for $(l = 1, k_1 = n)$ and $(l = n, k_l = 1)$ calculation of ASW is not possible. So we can start calculating ASW from at least $k_2 = n - 1$ and must stop at $r_{n-1} = 2$. For $l = 1$ the two closest observations are joined to form a cluster.

3.2 HOSil algorithm and description

An agglomerative hierarchical clustering algorithm can be defined based on a linkage criterion that optimizes ASW. Two clusters are merged together to form a single cluster if this gives the maximum ASW after the merge. Each time all possible cluster merges are tried out and those clusters are finally merged that give maximum ASW.

HOSil can also be used to find the best number of clusters (k) for the data. According to this criterion a best k will be the one that gives maximum value of ASW among all hierarchies level.

3.2.1 Algorithm's description

To understand how the algorithm works we take a small data set of 12 instances as an example. The data is plotted in Figure 3.1.

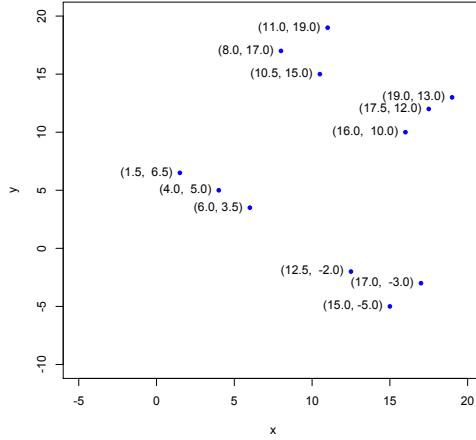


Figure 3.1 An example with 12 instance in two dimensions to illustrate the HOSil algorithm.

HOSil algorithm

Input

Input $n(n - 1)/2$ pairwise distances between points in \mathcal{X} , i.e., $d(x_i, x_j)$ for $x_i, x_j \in \mathcal{X}$.

Initialize

- (i) Set $l = 1$. Start with n clusters i.e., every object forms its own cluster,

$$\mathcal{C}_{k_l}^l = \left\{ C_1^l, \dots, C_{k_l}^l \right\} = \left\{ \{x_1\}, \{x_2\}, \dots, \{x_n\} \right\}, \quad k_l = n.$$

- (ii) Update $l = 2$. Join the two observations into one cluster that have minimum $d(x_i, x_j)$. Denote the resulting clustering as $\mathcal{C}_{k_l}^l = \left\{ C_1^l, \dots, C_{k_l}^l \right\}$, $k_l = (n - 1)$, and clustering labels for this clustering as $\gamma^l(\mathcal{X}, r) = \gamma^l(x_1, r), \dots, \gamma^l(x_n, r)$ where $r = 1, \dots, k_l$.
- (iii) Calculate $f^{(l)} = \bar{S}(\mathcal{C}_{k_l}^l, d)$ where $\bar{S}(\cdot)$ as defined in (2.12).

Repeat

- (i) Combine every cluster i with every other cluster j in the clustering $\mathcal{C}_{k_l}^l$. For all pairs (i, j) of cluster combinations denote a set of labels as $\gamma_{(i,j)}^l(x_1, r), \dots, \gamma_{(i,j)}^l(x_n, r)$, where $r = 1, \dots, (k_l - 1)$ and denote the corresponding clustering as $\mathcal{C}_{(k_l - 1)}^*$.
- (ii) Calculate $f_{(i,j)} = \bar{S}(\mathcal{C}_{(k_l - 1)}^*, d)$, where $\bar{S}(\cdot)$ as defined in (2.12).
- (iii) $(i^*, j^*) = \max f_{(i,j)}$, and denote the the corresponding label vector as $\gamma^l(\mathcal{X}, r)$.
- (iv) Merge the cluster pair $(C_{i^*}^l, C_{j^*}^l)$, such that,

$$\mathcal{C}_{k_{l+1}}^{l+1} = \mathcal{C}_{k_l}^l \cup \{C_{i^*}^l \cup C_{j^*}^l\} \setminus \{C_{i^*}^l, C_{j^*}^l\}, \quad k_{l+1} = k_l - 1.$$

Let $l = l + 1$.

- (v) Assign $f^{(l)} = f_{(i^*, j^*)}$.

Stop

When $l = n - 1$, i.e., $k_l = 2$.

Return

$f^{(l)}$ and $\gamma^l(\mathcal{X}, r)$ for all $l = 2, \dots, (n - 1)$.

Table 3.1 Pairwise Euclidean distances for the example data set

	1	2	3	4	5	6	7	8	9	10	11
2	2.50										
3	5.41	2.92									
4	13.65	12.65	12.35								
5	12.35	11.93	12.38	3.20							
6	16.29	15.65	15.70	3.61	4.03						
7	11.95	13.00	14.92	10.63	7.43	10.30					
8	14.30	15.21	16.92	10.74	7.62	9.55	2.50				
9	16.10	17.00	18.67	11.71	8.73	10.00	4.24	1.80			
10	12.38	14.87	17.73	23.09	20.50	24.33	15.03	17.18	18.44		
11	12.78	15.26	18.18	21.93	19.14	22.80	13.04	15.01	16.13	2.83	
12	8.515	11.011	13.901	19.53	17.12	21.05	12.50	14.87	16.35	3.91	4.61

Start when every object is in its own cluster with the label vector having values from 0 to 11. Calculate the pairwise distance between the clusters to identify the most similar clusters. The Euclidean distances for the example data set are shown in the Table 3.1. The smallest distance is between instances/clusters 8 and 9, bold entry in the table just mentioned. Form a new cluster by joining these two observations together. Thus we get 11 clusters and the resulting labelling set for this clustering can be given as (1, 2, 3, 4, 5, 6, 7, 0, 0, 8, 9, 10). The ASW for this clustering is 0.07116. Now at each hierarchy step we will reduce one cluster by combining those two clusters that give maximum ASW for the resulting clustering. For this purpose each cluster is grouped with every other cluster and ASW is calculated for each possible labels' set. There are 55 possible combinations of 11 clusters combining pairwise. For instance, cluster number "0" can be combined with all the other 10 clusters (0, 1), (0, 2), ..., (0, 10). The cluster labels are generated for each of these 55 unique possibilities and ASW is calculated. Thus, checking out the possibility of attaining the maximum ASW that can be attained at this hierarchy. One best clusters' combination was chosen out of these 55 combinations based on maximum ASW for the clustering.

For the example data set there are ($n - 2 = 9$) possible hierarchy levels. All possible clusters combinations for the example data set are displayed in Table 3.2.1. Some of the clustering labels vector generated for each of these possible combinations at each hierarchy are listed in Table 3.2.1. Similarly other labels can be generated. Table 3.2.1 shows the calculated ASW values for each case. For $l = 1$, 0.149 is the maximum value of the ASW obtained, which is corresponding to combination (0, 7) given in Table 3.2.1. Thus cluster numbers "0" and "7" must be combined at this hierarchy. Table 3.5 gives the best labels selected and corresponding ASW values from all possible combinations at a particular hierarchy level.

At $l = 2$, $k = 10$ and starting label set is (1, 2, 3, 4, 5, 6, 0, 0, 0, 7, 8, 9). Again all

possibilities of combinations are generated which is ${}^{10}C_2 = 45$. The labels at each hierarchy depends upon the labelling set obtained from previous hierarchy. So for the above selected label vector from $l = 1$, combining clusters (0, 1) at $l = 2$ will give (0, 1, 2, 3, 4, 5, 0, 0, 0, 6, 7, 8). But if (0, 1, 2, 3, 4, 5, 6, 7, 0, 0, 8, 9) was selected instead from $l = 1$, then combining cluster (0, 1) will give different labels as (0, 0, 1, 2, 3, 4, 5, 6, 0, 0, 7, 8). Thus we make sure that at each hierarchy level the possibility of combining each pair of clusters is considered, and those clusters who have already been combined in previous hierarchy remain combined at all latter stages. The same observation in different set can get different label. Therefore, the particular label as a cluster label for an observation does not matter but the membership of the cluster for the observation in a cluster matters. The process continues until we reach two clusters clustering.

Table 3.2: Total possibilities of combination of clusters for example data set at each hierarchy

${}^{11}C_2 = 55$	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)	(0, 6)	(0, 7)	(0, 8)	(0, 9)	(0, 10)
$l = 1$	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)	(1, 7)	(1, 8)	(1, 9)	(1, 10)	
	(2, 3)	(2, 4)	(2, 5)	(2, 6)	(2, 7)	(2, 8)	(2, 9)	(2, 10)		
	(3, 4)	(3, 5)	(3, 6)	(3, 7)	(3, 8)	(3, 9)	(2, 10)			
	(4, 5)	(4, 6)	(4, 7)	(4, 8)	(4, 9)	(4, 10)				
	(5, 6)	(5, 7)	(5, 8)	(5, 9)	(5, 10)					
	(6, 7)	(6, 8)	(6, 9)	(6, 10)						
	(7, 8)	(7, 9)	(7, 10)							
	(8, 9)	(8, 10)								
	(9, 10)									
${}^{10}C_2 = 45$	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)	(0, 6)	(0, 7)	(0, 8)	(0, 9)	
$l = 2$	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)	(1, 7)	(1, 8)	(1, 9)		
	(2, 3)	(2, 4)	(2, 5)	(2, 6)	(2, 7)	(2, 8)	(2, 9)			
	(3, 4)	(3, 5)	(3, 6)	(3, 7)	(3, 8)	(3, 9)				
	(4, 5)	(4, 6)	(4, 7)	(4, 8)	(4, 9)					
	(5, 6)	(5, 7)	(5, 8)	(5, 9)						
	(6, 7)	(6, 8)	(6, 9)							
	(7, 8)	(7, 9)								
	(8, 9)									
${}^9C_2 = 36$	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)	(0, 6)	(0, 7)	(0, 8)		
$l = 3$	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)	(1, 7)	(1, 8)			
	(2, 3)	(2, 4)	(2, 5)	(2, 6)	(2, 7)	(2, 8)				
	(3, 4)	(3, 5)	(3, 6)	(3, 7)	(3, 8)					

(4, 5)	(4, 6)	(4, 7)	(4, 8)
(5, 6)	(5, 7)	(5, 8)	
(6, 7)	(6, 8)		
(7, 8)			

⁸ $C_2 = 28$	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)	(0, 6)	(0, 7)
$l=4$	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)	(1, 7)	
	(2, 3)	(2, 4)	(2, 5)	(2, 6)	(2, 7)		
	(3, 4)	(3, 5)	(3, 6)	(3, 7)			
	(4, 5)	(4, 6)	(4, 7)				
	(5, 6)	(5, 7)					
	(6, 7)						

⁷ $C_2 = 21$	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)	(0, 6)
$l=5$	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)	
	(2, 3)	(2, 4)	(2, 5)	(2, 6)		
	(3, 4)	(3, 5)	(3, 6)			
	(4, 5)	(4, 6)				
	(5, 6)					

⁶ $C_2 = 15$	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)
$l=6$	(1, 2)	(1, 3)	(1, 4)	(1, 5)	
	(2, 3)	(2, 4)	(2, 5)		
	(3, 4)	(3, 5)			
	(4, 5)				

⁵ $C_2 = 10$	(0, 1)	(0, 2)	(0, 3)	(0, 4)
$l=7$	(1, 2)	(1, 3)	(1, 4)	
	(2, 3)	(2, 4)		
	(3, 4)			

⁴ $C_2 = 6$	(0, 1)	(0, 2)	(0, 3)
$l=8$	(1, 2)	(1, 3)	
	(2, 3)		

³ $C_2 = 3$	(0, 1)	(0, 2)
$l=9$	(1, 2)	

Table 3.3: Clustering labels for all the combinations

Sr. No.	1	2	3	4	5	6	7	8	9	10	11	12
x	6.0	4.0	1.5	8.0	10.5	11.0	16.0	17.5	19.0	15.0	17.0	12.5
y	3.5	5.0	6.5	17.0	15.0	19.0	10.0	12.0	13.0	-5.0	-3.0	-2.0
Initial Labels												
l=0	1	2	3	4	5	6	7	0	0	8	9	10
l=1												
(0,1)	0	1	2	3	4	5	6	0	0	7	8	9
(0,2)	1	0	2	3	4	5	6	0	0	7	8	9
(0,3)	1	2	0	3	4	5	6	0	0	7	8	9
(0,4)	1	2	3	0	4	5	6	0	0	7	8	9
(0,5)	1	2	3	4	0	5	6	0	0	7	8	9
:	:			...			:		...			:
(8,10)	2	3	4	5	6	7	8	1	1	0	9	0
(9,10)	2	3	4	5	6	7	8	1	1	9	0	0
l=2												
(0,1)	0	1	2	3	4	5	0	0	0	6	7	8
(0,2)	1	0	2	3	4	5	0	0	0	6	7	8
(0,3)	1	2	0	3	4	5	0	0	0	6	7	8
:	:			...			:		...			:
(7,9)	2	3	4	5	6	7	1	1	1	0	8	0
(8,9)	2	3	4	5	6	7	1	1	1	8	0	0
l=3												
(0,1)	1	2	3	4	5	6	0	0	0	0	0	7
(0,2)	0	2	3	4	5	6	1	1	1	0	0	7
(0,3)	2	0	3	4	5	6	1	1	1	0	0	7
:	:			...			:		...			:
(7,8)	3	4	5	6	7	0	2	2	2	1	1	0
l=4												
(0,1)	1	2	3	4	5	6	0	0	0	0	0	0
(0,2)	0	2	3	4	5	6	1	1	1	0	0	0
(0,3)	2	0	3	4	5	6	1	1	1	0	0	0

\vdots	\vdots			\dots			\vdots			\dots		\vdots
(6,7)	3	4	5	6	0	0	2	2	2	1	1	1
$l=5$												
(0,1)	0	0	2	3	4	5	1	1	1	0	0	0
(0,2)	0	0	2	3	4	5	0	0	0	1	1	1
(0,3)	0	0	0	3	4	5	2	2	2	1	1	1
\vdots	\vdots			\dots			\vdots			\dots		\vdots
(5,6)	1	1	4	5	0	0	3	3	3	2	2	2
$l=6$												
(0,1)	0	0	0	2	3	4	1	1	1	0	0	0
(0,2)	0	0	0	2	3	4	0	0	0	1	1	1
(0,3)	0	0	0	0	3	4	2	2	2	1	1	1
\vdots	\vdots			\dots			\vdots			\dots		\vdots
(4,5)	1	1	1	4	0	0	3	3	3	2	2	2
$l=7$												
(0,1)	0	0	0	0	0	3	2	2	2	1	1	1
(0,2)	1	1	1	0	0	3	2	2	2	0	0	0
(0,3)	1	1	1	0	0	3	0	0	0	2	2	2
\vdots	\vdots			\dots			\vdots			\dots		\vdots
(3,4)	2	2	2	1	1	0	0	0	0	3	3	3
$l=8$												
(0,1)	0	0	0	0	0	0	2	2	2	1	1	1
(0,2)	1	1	1	0	0	0	2	2	2	0	0	0
(0,3)	1	1	1	0	0	0	0	0	0	2	2	2
\vdots	\vdots			\dots			\vdots			\dots		\vdots
(2,3)	2	2	2	1	1	1	0	0	0	0	0	0
$l=9$												
(0,1)	0	0	0	0	0	0	0	0	0	1	1	1
(0,2)	1	1	1	0	0	0	0	0	0	0	0	0
(1,2)	0	0	0	1	1	1	1	1	1	0	0	0

Table 3.4: ASW for each set of clustering labels at different hierarchies for example data set. Bold values represent the maximum ASW attained against all possible clustering vectors at a particular hierarchy level.

¹¹	$C_2 = 55$	-0.165	-0.189	-0.195	-0.157	-0.119	-0.133	0.149^a	-0.167	-0.185	-0.181
	$l=1$	0.113	-0.003	-0.061	-0.062	-0.069	-0.002	-0.060	-0.066	-0.038	
		0.095	-0.050	-0.060	-0.068	-0.005	-0.066	-0.072	-0.053		
		-0.059	-0.059	-0.066	-0.007	-0.069	-0.075	-0.061			
		0.093	0.078	0.017	-0.072	-0.079	-0.072				
		0.039	0.024	-0.069	-0.075	-0.067					
		0.005	-0.073	-0.078	-0.072						
		-0.009	0.002	0.012							
		0.127	0.072								
		0.032									
¹⁰	$C_2 = 45$	-0.024	-0.040	-0.059	0.010	0.087	0.039	-0.054	-0.037	-0.027	
	$l=2$	0.214	0.094	0.036	0.052	0.029	0.031	0.031	0.059		
		0.191	0.047	0.056	0.030	0.026	0.024	0.044			
		0.038	0.056	0.032	0.023	0.021	0.036				
		0.204	0.176	0.030	0.019	0.025					
		0.147	0.049	0.042	0.049						
		0.019	0.018	0.024							
		0.217	0.168								
		0.129									
⁹	$C_2 = 36$	-0.136	0.007	0.003	-0.008	-0.023	0.003	-0.023	0.319		
	$l=3$	0.055	0.022	0.001	0.070	0.147	0.099	0.115			
		0.281	0.155	0.096	0.112	0.089	0.177				
		0.256	0.107	0.116	0.090	0.170					
		0.104	0.117	0.093	0.168						
		0.270	0.237	0.174							
		0.214	0.193								
		0.170									
⁸	$C_2 = 28$	-0.038	0.225	0.199	0.174	0.132	0.168	0.124			
	$l=4$	0.161	0.137	0.117	0.186	0.263	0.215				
		0.405	0.288	0.230	0.246	0.223					
		0.366	0.223	0.232	0.206						
		0.213	0.232	0.209							
		0.379	0.352								

		0.335					
⁷	$C_2 = 21$	0.209	0.071	0.518	0.246	0.248	0.210
	$l=5$	0.045	0.313	0.223	0.254	0.210	
		0.269	0.265	0.336	0.287		
		0.356	0.377	0.362			
		0.458	0.425				
		0.408					
⁶	$C_2 = 15$	0.334	0.179	0.406	0.450	0.406	
	$l=6$	0.204	0.350	0.4000	0.361		
		0.398	0.474	0.424			
		0.5889	0.560				
		0.544					
⁵	$C_2 = 10$	0.405	0.295	0.3967	0.720		
	$l=7$	0.406	0.250	0.515			
		0.276	0.476				
		0.553					
⁴	$C_2 = 6$	0.496	0.329	0.640			
	$l=8$	0.545	0.425				
		0.435					
⁴	$C_2 = 3$	0.484	0.319				
	$l=9$	0.507					

Table 3.5 gives the maximum ASW at $l = 7$. An empirical behaviour of ASW for example data set is, it keeps increasing from the hierarchy level 0 until level 7 and after that it starts decreasing. Of course this behaviour can be different for clusters of different natures which can only be explored in simulation. For the example data set the best number of clusters is four based on ASW linkage.

3.2.2 Some notes on implementation

The algorithm first calculates the pairwise distances between all the instances if not provided. It then joins the two most similar observations into a cluster and calculates ASW for this clustering. The algorithm is written in C++ and an interface is provided

Table 3.5 Best labels selected at each hierarchy based on HOSil

No. of clusters	Hierarchy level	Clustering Labels												ASW
$k = 11$	$l = 0$	1	2	3	4	5	6	7	0	0	8	9	10	0.0712
$k = 10$	$l = 1$	1	2	3	4	5	6	0	0	0	7	8	9	0.1487
$k = 9$	$l = 2$	2	3	4	5	6	7	1	1	1	0	0	8	0.2170
$k = 8$	$l = 3$	2	3	4	5	6	7	1	1	1	0	0	0	0.3193
$k = 7$	$l = 4$	0	0	3	4	5	6	2	2	2	1	1	1	0.4054
$k = 6$	$l = 5$	0	0	0	3	4	5	2	2	2	1	1	1	0.5178
$k = 5$	$l = 6$	1	1	1	0	0	4	3	3	3	2	2	2	0.5886
$k = 4$	$l = 7$	1	1	1	0	0	0	3	3	3	2	2	2	0.7200
$k = 3$	$l = 8$	1	1	1	0	0	0	0	0	0	2	2	2	0.6402
$k = 2$	$l = 9$	0	0	0	1	1	1	1	1	1	0	0	0	0.5065

in the R language [R Core Team \(2015\)](#) through the Rcpp ([Eddelbuettel et al. \(2011\)](#)) package. The algorithm needs a special form of input. The lower triangular distance matrix obtained through R function “dist()” is stored in a vector with an additional entry zero stored at first place to pass to C++. Thus this vector will have $n \times (n - 1)/2 + 1$ entries. This is because of the programming logic of the algorithm implementation and users don't have to worry about this input as an additional function called “filldys” is included to prepare the required input.

The algorithm is implemented in such a way that it can be used with data matrix or with the calculated distances obtained from any measure to return a set of dissimilarities as similar to the function “dist()” in the R package “cluster” ([Maechler et al. \(2017\)](#)). The algorithm just needs data to cluster, and no other additional parameter is needed. It can estimate the best number of clusters itself. However, in the experiments in Section (3.5), the clustering results against the true number of clusters were always examined. The output is designed in such a way that it is possible to retrieve clustering results against any k . One advantage of this is that we can handle the two tasks i.e., finding clustering and estimation of number of clusters in the same framework, so that we can also compare them. This can help in understanding whether the algorithm is capable of finding the true clustering or not for the fixed k even if it failed estimating the correct k . It is quite possible for some data sets that the clustering obtained against estimated k is different than the one with fixed known k . In such situations it is worthwhile to explore what this clustering looks like and what makes HOSil to produce such a clustering. How should a clustering look based on optimum ASW criterion? We will try to explore in simulations how likely it is for the proposed method to estimate the true number of clusters k and why it gives the clustering it gives.

3.3 Characteristics of interest for clustering

We have generated various data sets having different clustering characteristics and complexity to test the proposed method. In this section we will introduce all the synthetic data sets used in the experiments. We simulated data sets for several scenarios covering clustering difficulties of various kind. Some of these characteristics are listed below

- (i) Clusters with different variations among observations, i.e., compact and widely spread clusters,
- (ii) Equal and unequal sized clusters,
- (iii) Clusters from different distributions assuming every individual cluster is coming from a single distribution. For instance, clusters from Gaussian, Student's t , Gamma or Beta distributions,
- (iv) Clusters from skewed distributions,
- (v) Different types of clusters for instance, spherical, non-spherical, elongated or arbitrarily shaped clusters,
- (vi) Close and far away clusters, i.e., the distance between the means of clusters are varied,
- (vii) Overlapping and well-separated clusters,
- (viii) Nested clusters,
- (ix) Clusters with correlated variables within clusters,
- (x) Different number of clusters,
- (xi) Different number of variables/dimensions,

and more. Note that a good mixture of most of the above characteristics is made within a single data set to make the clustering task more challenging. In addition, we compared the runtime asymptotic consistency of the algorithm. We checked how many observations the algorithm can handle efficiently to cluster and what will be the efficiency of the algorithm as the number of clusters, the number of observations, and the number of dimensions increases.

3.4 Definition of data generating processes

For the data generating processes (DGPs) several probability distributions have been used. We first define the notations for these distributions. Let $N_p(\mu_p, \Sigma_{p \times p})$ represent the p -variate Gaussian distribution with mean μ_p and covariance matrix $\Sigma_{p \times p}$. Let $SN(\zeta, \omega, \alpha, \tau)$ represent a skew Gaussian univariate distribution with $\zeta, \omega, \alpha, \tau$ as location, scale, shape and hidden mean parameters of the distribution respectively. Let $\mathbb{U}(a, b)$ represent the uniform distribution defined over the continuous interval a and b . Let t_v represent Student's t distribution with v degrees of freedom. Let $t_r(v)$ represent the non-central t distribution with r degrees of freedom and v be the non-centrality parameter. Let $\text{Gam}(\alpha, \beta)$ represent Gamma distribution where α and β are shape and rate parameters, respectively. Let $\text{NBeta}(\nu_1, \nu_2, \lambda)$ represents the non-central Beta distribution of Type-I with ν_1, ν_2 be two shape parameters and λ being the non-centrality parameter. Let $\text{Exp}(\lambda)$ represent the Exponential distribution with λ being the rate parameter. Let $\mathbb{F}_{(\nu_1, \nu_2)}(\lambda)$ represent the non-central F distribution with ν_1, ν_2 degrees of freedom and λ be the non-centrality parameter. Let $\mathbb{W}(\tau, \zeta)$ represent the Weibull distribution with τ, ζ as shape and scale parameter, respectively. The definitions of all distributions used in data generating processes (DGPs) are given in Appendix A. Let I_p be the identity matrix of order p , where p represents the number of dimensions. The DGPs are defined as below.

Model 1:

Two clusters of equal sizes are generated in two dimensions coming from different distributions. 100 observations are generated from the Gaussian distribution with identity covariance matrix centred at $(0, 5)$. 100 observations drawn from $\mathbb{U}(-10, 1)$ independently along both dimensions. The result is one compact spherical cluster located at the corner of a uniformly distributed cluster.

Model 2:

Two Gaussian clusters of unequal sizes and variations were generated in two dimensions independently. The clusters contain 50 and 100 observations centred at $(1.5, 5)$ and $(0, 5)$, respectively, with $0.1I_2$ and $0.5I_2$ covariance matrices, respectively. The result is one small, compact spherical cluster lying close to a bigger widely spread spherical cluster.

Model 3:

Three clusters of unequal sizes and variations were generated in two dimensions. Two clusters were generated from independent bi-variate Gaussian distributions with 50

and 100 observations centred at (0, 5) and (0.5, 5.5), respectively, with covariance matrix as $0.1I_2$ and $0.2I_2$ respectively. The third cluster with 50 observations was generated from a non-central t distribution with $t_{25}(5)$ and $t_{25}(10)$ independently. The clusters are of such nature that the generated non-central t cluster has a wider spread than the two Gaussian clusters which were kept close to each other as compared to the bigger spread cluster.

Model 4:

Three Gaussian clusters in two dimensions of unequal variations. The clusters contains 50, 50, and 100 observations, with covariance matrices as $0.1I_2$, $0.1I_2$ and $0.5I_2$, while the clusters are centred at (-2, 5), (2, 5), (0, 5), respectively. The result is one bigger, spherical widely spread cluster located between two small, spherical and compact clusters.

Model 5:

Three Gaussian clusters of unequal sizes having different variations along dimensions were generated independently in two dimensions. The clusters were randomly chosen to have 25, 50 and 75 observations without replacement such that the total sample size is 150 always. The clusters are centred at (0, 5), (2, 5), (-2, 5) respectively. The first cluster has covariance matrix as $0.5I_2$, the other two clusters have common covariance matrix defined as $\Sigma = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.7 \end{bmatrix}$. The result is one spherical cluster located between two clusters having wider spread across one dimension as compared to the others.

Model 6:

Three Gaussian clusters in two dimensions of equal sizes, different variations and different shapes. 50 observations were generated from (0, 5) with covariance matrix as $0.5I_2$. 50 observations were generated from Gaussian distribution with means (1.5, 5) with covariance matrix as $\Sigma = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.7 \end{bmatrix}$. 50 observations were generated from (1.5, 7) with co-variance matrix as $0.1I_2$. The clusters look like a wider spread spherical cluster located left to the vertical cluster and a smaller compact cluster located just at the top of the vertical cluster.

Model 6.A:

The Gaussian cluster with centre (1.5, 7) in Model 6 is re-centred at (-1.5, 7). The purpose is to move the location of the compact cluster far from the dense region of other

two clusters.

Model 6.B:

A fourth cluster of 50 observations from Gaussian distributions centred at (-1.5, 3) with covariance matrix as $0.1I_2$ is added to Model 6.A.

Model 7:

Four almost touching spherical clusters of equal sizes and variations. The clusters were generated from Gaussian distributions each with 50 observations and $0.2I_2$ covariance matrix. The clusters were centred at (1, 1), (1, 2), (2, 1), and (2, 2).

The touching cluster problem is well known in clustering literature for instance see [Zhong et al. \(2010\)](#). These cluster have slightly different versions in literature. For instance, a pair of cluster is called touching clusters that are joined together by a small neck and removal of this neck produces two separate clusters like in [Zhong et al. \(2010\)](#) or Aggregation (from [Gionis et al. \(2007\)](#)) data set. The term touching clusters is also used in literature for clusters that have very close or joining boundaries with each other for instance the Tetra data set in the Fundamental clustering problem suite (FCPS), see [Ultsch \(2005\)](#). These kind of data sets is of natural interest for ASW based clustering method. It is not known what should be the level of separation between clusters that ASW can determine them as two separate clusters.

Model 8:

Four clusters of equal sizes each having 50 observations were generated in two dimensions. One cluster was generated from independent non-central t distributed variables with parameters $t_7(10)$ and $t_7(30)$. One cluster was generated from $\mathbb{U}(10, 15)$ independently along both dimensions. One cluster was generated from bivariate normal distribution parametrized by mean (2, 2), and covariance I_2 generated independently across both dimensions. The fourth cluster is also from independent bivariate Gaussian distributions parametrized by, mean (20, 80) with covariance matrix $\Sigma = \begin{bmatrix} 0.1 & 0 \\ 0 & 2 \end{bmatrix}$.

Model 9:

Four clusters with correlated variables in 2 dimensions. Two clusters were generated having 25 and 50 observations, and, centred at (7.5, 4) and (-2.5, 3) with a common covariance matrix $\Sigma = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$, respectively. The other two clusters contains 50 and

75 observations centred at (-7.5, 4) and (2.5, 3) with a common covariance matrix $\Sigma = \begin{bmatrix} 1 & -0.7 \\ -0.7 & 1 \end{bmatrix}$, respectively.

Model 10:

Five clusters in two dimensions from different distributions. The clusters are parametrized from Chi-squared, \mathbb{F} , t , Gaussian and skewed Gaussian distributions as $\chi_7^2(50)$ and $\chi_{10}^2(80)$, $\mathbb{F}_{(2,6)}(4)$ and $\mathbb{F}_{(5,5)}(4)$, $t_{40}(100)$ and $t_{35}(150)$, $N((100, 0), 0.9I_2)$, $SN(20, 0.9, 2, 4)$ and $SN(200, 0.8, 3, 6)$. The clusters contains 50 observations each and were generated independently along both dimensions.

Model 11:

Six clusters from different distributions in two dimensions. The clusters are parametrized as $\mathbb{U}(-6, -2)$, $Exp(10)$, $Beta(2, 3, 120)$, $\mathbb{W}(10, 4)$, $Gam(15, 2)$ in both dimensions, whereas one cluster from $SN(5, 0.6, 4, 5)$ along first dimension and $SN(0, 0.6, 4, 5)$ across second dimension. The clusters contains 50 observations each and were generated independently along both dimensions.

Model 12:

Six correlated Gaussian clusters in two dimensions. The clusters are centred at (-10, -10), (5, -2), (20, 0), (-30, -5), (-40, 40), (-50, 30) with covariance matrices as $\Sigma_1 = \Sigma_3 = \begin{bmatrix} 9 & 10.8 \\ 10.8 & 16 \end{bmatrix}$, $\Sigma_2 = \Sigma_4 = \Sigma_6 = \begin{bmatrix} 9 & -10.8 \\ -10.8 & 16 \end{bmatrix}$, respectively. Cluster 5 has covariance matrices as $\Sigma_5 = \begin{bmatrix} 9 & 1.2 \\ 1.2 & 16 \end{bmatrix}$. Cluster 1 to 4 contains 50 observations each whereas cluster 5 and 6 contains 25 observations each.

Model 13:

Fourteen Gaussian clusters in two dimensions. Two clusters were generated with 25 observations each having common co-variance matrix as $0.5I_2$ centred at (0, 2) and (0, -2). Six clusters were generated with 25 observations each having common covariance matrices as $\begin{bmatrix} 0.1 & 0 \\ 0 & 0.7 \end{bmatrix}$. The clusters are centred at (-4, -2), (-3, -2), (-2, -2), (2, -2), (3, -2), and (4, -2). The remaining six clusters have common covariance matrix as $0.1I_2$ and 25 observations each centred at (-4, 2), (-3, 2), (-2, 2), (2, 2), (3, 2), and (4, 2).

Model 14:

Three elongated clusters in three dimensions having 100 observations each. Let $v \sim U(-0.5, 0.5)$. Generate the first cluster along three dimensions namely x , y and z as $x \sim v + N(0, 1)$, $y \sim v + N(0, 1)$ and $z \sim v + N(0, 1)$. Generate cluster 2 along three dimensions as $x \sim v + N(0, 1) + 2$, $y \sim v + N(0, 1) + 2$ and $z \sim v + N(0, 1) + 2$. Generate cluster 3 in the same way by adding value 4 in each dimensions respectively.

Model 15:

Eight Gaussian clusters surrounding a cluster formed by uniformly distributed points in a unit circle. The unit circle contains 33 observations and is centred at [0, 0, 0]. The 8 Gaussian clusters contain 25 observations each. Four clusters are centred at (-7, -0.2, -0.2), (0.2, -4, -4), (0.5, 3, 3), and (7, -1, -1) with a common covariance matrix as $0.1I_3$. Two clusters are centred at (-5.5, 2.5, 2.5) and (4.5, -3, -3) with a common

covariance matrix $\begin{bmatrix} 0.6 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 0.6 \end{bmatrix}$. The remaining two clusters are centred at (-7, -0.2,

-0.2), (0.2, -4, -4), (0.5, 3, 3), and (7, -1, -1) with a common covariance matrix $0.1I_3$. Two clusters are centred at (-4, -2.5, -2.5) and (5, 1.5, 1.5) with a common covariance matrix

$$\begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 0.4 \end{bmatrix}.$$

Model 16:

Ten clusters in 100 dimensions. The clusters are centred at -21, -18, -15, -9, -6, 6, 9, 15, 18, 21. The clusters are in 100 dimensions such that the 100 dimensional mean vectors of these values were generated for all clusters. The number of observations for these ten clusters are 20, 40, 60, 70, and 50 each for six of the remaining clusters. The number of observations for the means of clusters were not fix. Any cluster can take any number of observations from these such that any six clusters have equal number of observations i.e., 50 and the remaining four has different observations each, which is one out of 20, 40, 60, 70 values. The total size of the data is always 490 observations. The covariance matrix for each of these clusters is one out of $0.05I_{100}$, $0.1I_{100}$, $0.15I_{100}$, $0.175I_{100}$, $0.2I_{100}$ matrices. The covariance matrix for each cluster was chosen randomly with replacement out of these, such that as a result, all the clusters can have same covariance matrix, two or more clusters can have same covariance matrix or all of the 10 clusters can have different same covariance matrices.

Model 17:

Three clusters in 1000 dimensions. Each cluster contains 40 observations from standard Gaussian distributions with each of the first 100 coordinates centred at -5, 0 and 5 respectively. The remaining dimensions of all clusters are centred at 0.

A data sets generated from each of the 17 models is displayed in Figure 3.2.

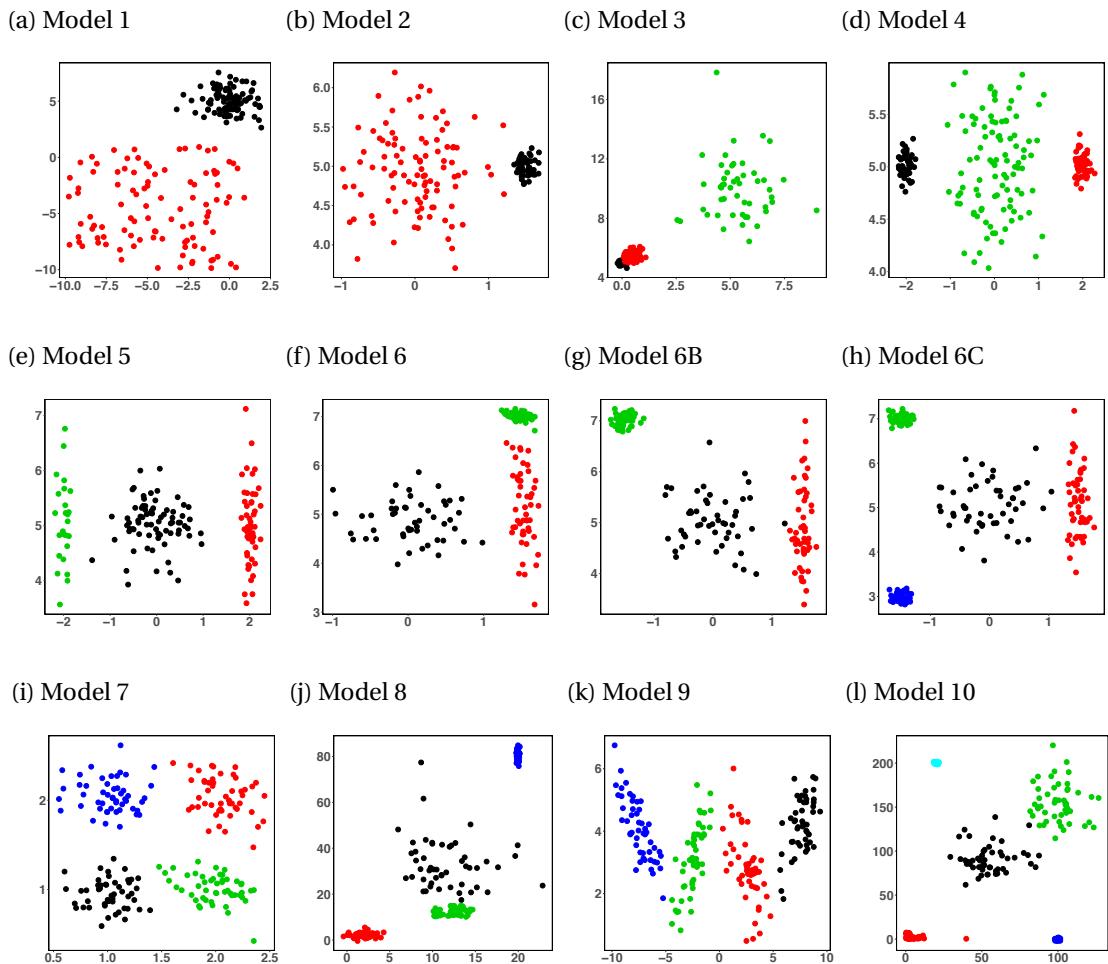


Figure 3.2 Plots of all data sets, each generated from the DGPs included in the study.
(cont.)

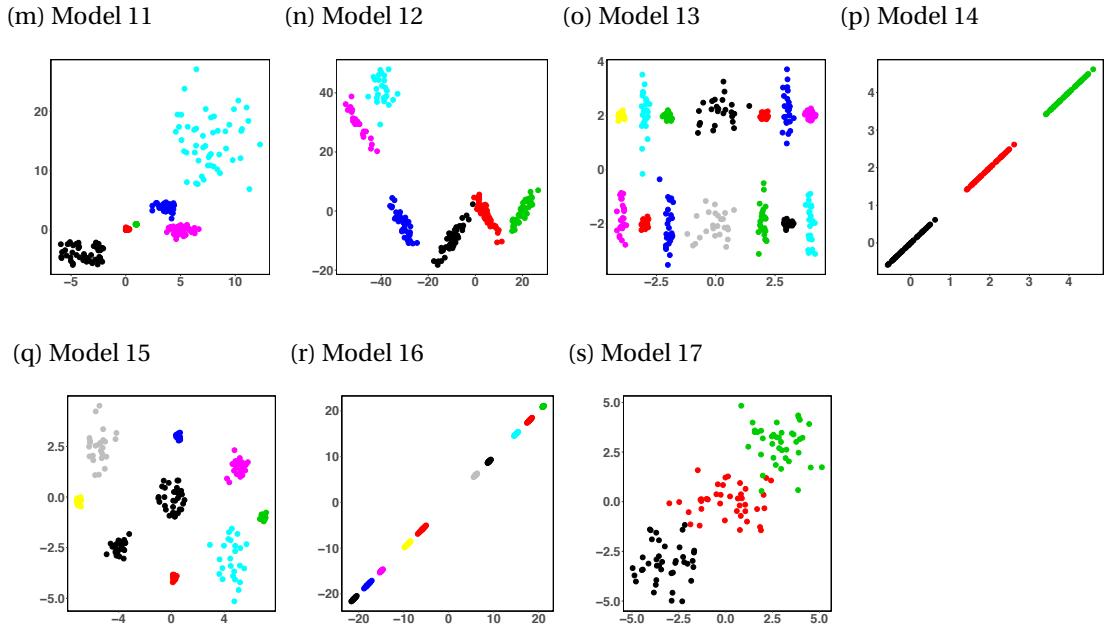


Figure 3.2 Plots of all data sets, each generated from the DGPs included in the study.

3.5 Simulation design

All the data generating processes just defined above were considered for simulation. The simulation was done in R language ([R Core Team \(2015\)](#)). Along with the proposed algorithm we have applied various widely applicable clustering algorithms with Euclidean distances, namely k -means, partitioning around medoids (PAM), hierarchical clustering algorithms (with single, complete, average, Ward, McQuitty methods), spectral clustering, model-based clustering method and PAMSIL clustering algorithm. For all the hierarchical clustering methods we use function ‘`hclust()`’ available with R base package “`stats`”. For Ward’s method we use option “`Wards.D2`” for the method argument of ‘`hclust()`’. For k -means and PAM we use functions ‘`kmeans()`’ (with `nstart = 100`) and ‘`pam()`’ also available through R base “`stats`” and R package “`cluster`” (version: 2.0.6, [Maechler et al. \(2017\)](#)), respectively. For spectral and model-based clustering we have used the R packages “`kernlab`” (version: 0.9.25, [Zeileis et al. \(2004\)](#)) and “`mclust`” (version: 5.2.3, [Scrucca et al. \(2017\)](#)) implementation, available through the ‘`specc()`’ and ‘`Mclust()`’ functions, respectively. For PAMSIL we have used the standalone C function written by [Van der Laan et al. \(2003\)](#). For all the methods, we used the default settings except otherwise stated. Let the number of data sets generated are denoted by B . We have generated $B = 50$ data sets for each DGP.

We have also tried DBSCAN algorithm proposed by [Ester et al. \(1996\)](#). DBSCAN

is a density based clustering algorithm, and an implementation for this is available in R through function ‘`dbscan()`’ in package “`dbSCAN`” ([Hahsler and Piekenbrock \(2018\)](#)). The algorithm has two parameters `epsilon` and `minimum points` (‘`eps`’ and ‘`MinPts`’ arguments in function `dbscan()`). The algorithm finds the density regions which are defined by the number of points within certain region of each point. Let x_i and x_j be two points in the data \mathcal{X} to cluster. The basic idea is that there exists at least a minimum number of points (`MinPts`) in the `eps`-neighborhood of a point say x_i , where the `eps`-neighborhood of the point x_i itself is defined by $ENB(x_i) = \{x_j \in \mathcal{X} | d(x_i, x_j) \leq eps\}$. Although the authors provided some advise to chose the two parameters yet it is hard to fix these in advance as they are purely data driven. The current available implementation of density based clustering method ‘`dbSCAN`’ is not suitable for simulations as it is hard to choose the parameters of the algorithm suitable for the experiments. The performance of the algorithm can be greatly improved by experimenting and visual analysis of the clustering results against the parameters, rather than sticking with the default recommendation of the parameters, as they are not always good. This is hard to do in a simulation setup. For some of the simulated models we have taken several ‘`eps`’ values in a particular range at a constant frequency. Firstly for different models this common range was not useful, secondly, it is hard to decide an appropriate range for each model. Many times the algorithm assigns everything to one cluster and identifies a few points as noise, which makes it unfit for the calculation of further results, for instance, internal/external indices or the estimation of number of clusters considered in the simulations.

We have also considered the estimation of number of clusters from several internal indices with combinations of several clustering methods. We have applied k -means, PAM and agglomerative hierarchical clustering with five linkage methods with 11 different methods of estimation of number of clusters namely, H, CH, KL, ASW, Gap, Jump, PS, BI, BIC, PAMSIL and HOSil. For the estimation of number of clusters with ASW index we have also considered model-based and spectral clustering methods.

Note that BIC was used only with model-based clustering method and we did not use all other internal estimation indices with model-based clustering. In principle all of these internal indices can be used with model-based clustering for the estimation of number of clusters but this is not a standard practice. Since BIC is based on maximum likelihood it’s not logical to use BIC with clustering methods that don’t maximise the likelihood. One can anticipate that maximum likelihood will always be better in terms of BIC such that its a natural choice for model-based clustering. On the other hand, a somewhat similar argument holds for the use of other internal validation indices together with model-based clustering i.e., for some of these combinations it’s either not logical to combine two different or even conflicting aims or it is not quite obvious and well explored in literature that the use of these together make sense or not.

The maximum number of clusters was set to 15 for the estimation of number of

clusters. For H, CH, KL, BI and ASW we have considered the number of clusters from 2-15 and for PS, Gap and BIC from 1-15 number of clusters were used. For H, CH, KL and Gap we have used R package “clustersim” (version: 0.45.2, [Walesiak and Dudek \(2017\)](#)). ASW is calculated through R package “cluster” ([Maechler et al. \(2017\)](#)). For PS and BI we have used R implementation through package “fpc” (version: 2.1.10, [Hennig \(2015a\)](#)). For the Jump method we have used the R code provided by the author see [Sugar and James \(2003a\)](#) for reference. We have used various transformation powers for the Jump method in simulation, particularly, we have estimated $\hat{\delta}_k$ (see Section [2.5.1.7](#) for details) using $Y = p/2$, $Y = p/3$, $Y = p/4$, $Y = p/5$, $Y = p/6$ and $Y = p/7$.

In a nutshell, for each DGP, we have first performed clustering for 10 clustering methods mentioned above for 2 to 15 numbers of clusters. From the DGP the true number of clusters are known which are referred as fixed or known number of clusters (k). For the fixed known k we have calculated ASW values and ARI for k -means, pam, single, complete, average, Wards, McQuitty, Spectral, and model-based clustering methods. For PAMSIL and HOSil we have calculated ASW values and ARI values. The ARI was calculated using the true clustering labels for each DGP. For these clustering methods we have only reported mean ASW and ARI values. Only for PAMSIL and HOSIL we have reported results for both fixed and estimated k . We have then estimated the number of clusters with H, CH, KL, Gap, Jump, PS, BI, ASW, BIC, PAMSIL, HOSil using the clustering results calculated from the clustering methods.

From these simulations we can infer many interesting results. For instance, HOSil clustering characteristics, analysis of HOSil for the estimation of k , and comparison of HOSil with existing methods for clustering. Also this systematic study has provided an insight about how existing clustering methods results differ from each other.

3.5.1 Discovering the true clustering

The major findings of the simulation are discussed in this section. Note that all discussions are based on the results on the $B = 50$ runs except for Model 13 which was computationally expensive because of greater number of observations and number of clusters. From Model 13, only $B = 25$ data sets were generated. Clustering results for each DGP and clustering method included in the study are displayed in figures in Section [B.1](#) of Appendix [B](#). These clustering results are only for one out of the 50 runs to give the readers an idea about the clustering solution found by various methods and to make discussions more understandable. Let $HOSil_k$ and $HOSil_{\hat{k}}$ denote the situation when we used HOSil for the true known fixed k and when k was estimated by HOSil, respectively. Similarly, $PAMSIL_k$ and $PAMSIL_{\hat{k}}$ notations hold for PAMSIL method. In the following discussions whenever the term “size of clusters” is used, we mean to refer the number of observations in clusters. The clustering represented in these figures are for the known k case for all clustering methods included in the study except for HOSil for which results are plotted for both known and estimated k . Moreover, cluster labels

are represented by colors and digits.

The aggregated values for $B = 50$ of ASW and ARI data sets for each DGP are reported in Tables 3.6-3.24. These results and results presented in Sections B.1, and B.2 (used in next section) of Appendix B are results of one simulation and are comparable.

Model 1 All the clustering methods performed well overall with a few misclassified points (see Figure B.1). The best ASW was obtained from k -means method and highest ARI was obtained from spectral clustering method (see Table 3.6). PAMSIL gave higher ASW value as compared to HOSil but with a smaller value of ARI as compared to HOSil.

Model 2 k -means, model-based clustering and HOSil performed well. PAM, spectral and PAMSIL clusterings have a few misclassified points whereas all the hierarchical methods were not able to identify the desired known clustering result for Model 2 (see Figure B.2). The highest ASW value was obtained from k -means clustering whereas the best ARI was obtained from model-based clustering (see Table 3.7). PAMSIL have achieved higher ASW value but lower ARI value as compared to HOSil.

Table 3.6 Results for Model 1.

Methods	fixed k		
	ASW	SE	ARI
true	0.6340		
k -means	0.6402	0.0017	0.8750
PAM	0.6400	0.0017	0.8902
single	0.5224	0.0297	0.7761
complete	0.6291	0.0043	0.9122
average	0.6323	0.0021	0.9163
Ward's	0.6184	0.0058	0.9040
McQuitty	0.6184	0.0058	0.9040
Spectral	0.6259	0.0018	0.9946
model-based	0.6138	0.0046	0.9689
PAMSIL	0.6461	0.0031	0.8845
HOSil	0.6354	0.0026	0.9797

	estimated k		
	PAMSIL	HOSil	
PAMSIL	0.6461	0.0031	0.8845
HOSil	0.6354	0.0026	0.9797

Table 3.7 Results for Model 2.

Methods	fixed k		
	ASW	SE	ARI
true	0.5575		
k -means	0.5794	0.0029	0.7527
PAM	0.5820	0.0028	0.8074
single	0.3038	0.0179	0.0317
complete	0.4252	0.0143	0.2859
average	0.5400	0.0125	0.5561
Ward's	0.4061	0.0199	0.2872
McQuitty	0.4061	0.0199	0.2872
Spectral	0.5543	0.0075	0.8179
model-based	0.5618	0.0036	0.9887
PAMSIL	0.5854	0.0034	0.8565
HOSil	0.5697	0.0032	0.9438

	estimated k		
	PAMSIL	HOSil	
PAMSIL	0.5854	0.0034	0.8565
HOSil	0.5699	0.0031	0.9363

Model 3 All the clustering methods combined the two closely related Gaussian clusters together and divided the cluster with bigger spread among observations into smaller clusters. Figure B.3 depicts the clustering results obtained for this model. Single linkage gave a one point cluster for an observation far from the dense region of the data. Only PAM and HOSil were able to retain the desired clustering for the fixed k . PAMSIL was not able to recover the correct clustering even for the fixed known k . The highest ARI was obtained for PAM and then for HOSil clustering for fixed k (see Table 3.8).

Model 4 Complete, single, and k -means methods didn't give the correct clusterings (see Figure B.4). All the hierarchical clustering methods have gave very low ARI values (Table 3.9).

Table 3.8 Results for Model 3.

Methods	fixed k		
	ASW	SE	ARI
true	0.5821		
k-means	0.7884	0.0017	0.4651
PAM	0.5845	0.0034	0.9062
single	0.7873	0.0061	0.5103
complete	0.7869	0.0029	0.4753
average	0.8028	0.0026	0.5068
Ward's	0.7916	0.0036	0.4839
McQuitty	0.7916	0.0036	0.4839
Spectral	0.7441	0.0213	0.4822
model-based	0.7350	0.0225	0.5409
PAMSIL	0.8070	0.0031	0.4960
HOSil	0.6354	0.0139	0.8358

	estimated k		
	ASW	SE	ARI
PAMSIL	0.8456	0.0018	0.5249
HOSil	0.8463	0.0012	0.5270

Table 3.9 Results for Model 4.

Methods	fixed k		
	ASW	SE	ARI
true	0.6750		
k-means	0.6837	0.0027	0.9177
PAM	0.6841	0.0028	0.9250
single	0.5342	0.0276	0.6784
complete	0.5707	0.0177	0.5905
average	0.6792	0.0029	0.9554
Ward's	0.5359	0.0187	0.6563
McQuitty	0.5359	0.0187	0.6563
Spectral	0.5239	0.0357	0.9476
model-based	0.6758	0.0041	0.9984
PAMSIL	0.6898	0.0033	0.9447
HOSil	0.6799	0.0026	0.9927

	estimated k		
	ASW	SE	ARI
PAMSIL	0.6898	0.0033	0.9447
HOSil	0.6799	0.0026	0.9927

Model 5 Single, complete, Ward, McQuitty have not identified different covariances and sizes of clusters correctly (Figure B.5). PAMSIL has also not identified this clustering structure correctly. The remaining methods performed good for this model. The ASW values obtained were smaller from both PAMSIL and HOSil than the maximum ASW value. HOSil has produced smaller ASW but greater ARI value as compared to PAMSIL (see Table 3.10).

Model 6 Only Model-based clustering, PAMSIL and HOSil performed well (see Figure B.6). Although PAMSIL \hat{k} and HOSil \hat{k} estimated the correct number of clusters, they gave a few misclassified points. Table 3.11 represents the ASW and ARI values obtained for this model. PAMSIL got higher ASW values but with smaller ARI values for both fixed and estimated k as compared to HOSil.

Model 6.A is a variation of Model 6 where cluster 3 was moved away from the top of cluster 2. All the hierarchical clustering method and k -means performed poorly here. Only model-based clustering and HOSil gave an exact classification (Figure B.7). The performance of k -means, PAM and spectral was also close to the methods just mentioned, but they have combined a few points from cluster 1 with cluster 2. PAMSIL and HOSil was not able to estimate the numbers of clusters here as three. They have combined the two clusters that were far away from a compact cluster. The maximum ARI was obtained from model-based clustering method (Table 3.12). HOSil produced smaller ARI as compared to PAMSIL. Similar results were observed for **Model**

Table 3.10 Results for Model 5.

Methods	fixed k		
	ASW	SE	ARI
true	0.5878		
k-means	0.5917	0.0039	0.9448
PAM	0.5921	0.0039	0.9471
single	0.5164	0.0198	0.7096
complete	0.4963	0.0146	0.7091
average	0.5838	0.0066	0.9155
Ward's	0.5265	0.0150	0.7641
McQuitty	0.5265	0.0150	0.7641
Spectral	0.5352	0.012	0.9308
model-based	0.5147	0.0172	0.9903
PAMSIL	0.5831	0.0054	0.9649
HOSil	0.5814	0.0037	0.9722

	estimated k		
	PAMSIL	SE	ARI
	0.5992	0.0088	0.8610
	0.5959	0.0058	0.8929

Table 3.11 Results for Model 6.

Methods	fixed k		
	ASW	SE	ARI
true	0.6021		
k-means	0.6357	0.0033	0.8194
PAM	0.6354	0.0032	0.8462
single	0.3163	0.0359	0.1377
complete	0.5217	0.0124	0.5606
average	0.6076	0.0092	0.865
Ward's	0.5082	0.0128	0.5977
McQuitty	0.5082	0.0128	0.5977
Spectral	0.5473	0.0174	0.8829
model-based	0.6080	0.0039	0.9834
PAMSIL	0.6390	0.0035	0.8677
HOSil	0.6716	0.0427	0.9415

	estimated k		
	PAMSIL	SE	ARI
	0.6402	0.0034	0.8476
	0.6716	0.0428	0.9192

6.B (Figure B.8), where HOSil again combined the previously mentioned two clusters and added a new compact cluster to the data. For this model HOSil produced smaller ARI and ASW values as compared to PAMSIL (Table 3.13).

Table 3.12 Results for Model 6.A.

Methods	fixed k		
	ASW	SE	ARI
true	0.6345		
k-means	0.6505	0.0025	0.9114
PAM	0.6503	0.0025	0.9183
single	0.5007	0.0129	0.5972
complete	0.5615	0.0124	0.7071
average	0.6247	0.0073	0.8321
Ward's	0.5573	0.0117	0.7098
McQuitty	0.5573	0.0117	0.7098
Spectral	0.5901	0.0204	0.8904
model-based	0.6375	0.0029	0.9873
PAMSIL	0.6486	0.0036	0.9266
HOSil	0.5912	0.0030	0.7853

	estimated k		
	PAMSIL	SE	ARI
	0.6823	0.0027	0.5764
	0.6867	0.0022	0.5890

Table 3.13 Results for Model 6.B.

Methods	fixed k		
	ASW	SE	ARI
true	0.7055		
k-means	0.7137	0.0021	0.6834
PAM	0.7135	0.0021	0.6839
single	0.5953	0.0118	0.7029
complete	0.6210	0.0088	0.6605
average	0.6895	0.0058	0.7083
Ward's	0.6171	0.0084	0.6812
McQuitty	0.6171	0.0084	0.6812
Spectral	0.6290	0.0292	0.6100
model-based	0.7026	0.0024	0.9907
PAMSIL	0.7174	0.0022	0.9522
HOSil	0.6432		0.7098

	estimated k		
	PAMSIL	SE	ARI
	0.7278	0.0021	0.8049
	0.7216	0.0428	0.7838

Model 7 k-means, PAM, Model-based clustering, PAMSIL and HOSil gave the clustering with a few misclassified points (see Figure B.9). Single linkage performed very

bad for this data resulting the lowest ARI (0.3042) among all methods. HOSil has produced smaller ARI values (0.9293) as compared to PAMSIL (0.9686) and many other clustering methods (see Table 3.14).

Model 8 The cluster labels 1, 2, 3 and 4 in panel (b) of the Figure B.10 represent the clusters generated from Student's t (t), Gaussian (N), Uniform (\mathbb{U}) and N distributions respectively. Single, complete, Ward's, McQuitty methods have combined many points from the t distributed cluster to the \mathbb{U} distributed cluster. k -means and PAM were not much different in this respect. Average linkage has combined the N distributed cluster with the \mathbb{U} distributed cluster by putting a few points from the \mathbb{U} distributed cluster in a separate cluster. Spectral clustering has divided the N cluster into two clusters and combined the t and \mathbb{U} clusters. Only Model-based clustering, PAMSIL and HOSil were able to identify the t and \mathbb{U} clusters (fixed k). However, many methods including HOSil, PAMSIL failed to estimate the numbers of clusters as four here. They put all the three clusters from N , \mathbb{U} and t distributions together in one cluster. PAMSIL produces higher ASW value as compared to HOSil (see Table 3.15). The maximum ASW was obtained from average linkage method and this maximum was higher than ASW value achieved from PAMSIL. However, the ARI value for HOSil was higher than PAMSIL and from all other clustering methods except from model-based clustering for fixed k . For estimated k the ARI for both PAMSIL and HOSil are very low.

Table 3.14 Results for Model 7.

Methods	fixed k		
	ASW	SE	ARI
true	0.5963		
k-means	0.6029	0.0026	0.9633
PAM	0.6027	0.0026	0.9633
single	0.0743	0.0335	0.3042
complete	0.5778	0.0056	0.8861
average	0.5955	0.0028	0.9350
Ward's	0.5388	0.0086	0.8697
McQuitty	0.5388	0.0086	0.8697
Spectral	0.4932	0.0216	0.8988
model-based	0.6029	0.0028	0.9623
PAMSIL	0.6076	0.0030	0.9686
HOSil	0.5965	0.0031	0.9293

Table 3.15 Results for Model 8.

	fixed k			estimated k		
	ASW	SE	ARI	ASW	SE	ARI
true	0.7388					
k-means	0.7634					
PAM	0.7673					
single	0.5941					
complete	0.6558					
average	0.6753					
Ward's	0.6630					
McQuitty	0.6630					
Spectral	0.5484					
model-based	0.7279					
PAMSIL	0.7650					
HOSil	0.7560					
PAMSIL	0.6076	0.0030	0.9686	0.7793	0.0025	0.5811
HOSil	0.5965	0.0031	0.9293	0.7758	0.0020	0.5211

Model 9 has four correlated Gaussian clusters. Model-based, PAMSIL, k -means, PAM, average linkage and HOSil were able to identify the clustering most closely related (in order names are mentioned) to the true clustering (see Figure B.11) based on

ARI values (see Table 3.16). Spectral clustering and single linkage methods have combined two clusters and formed a fourth cluster by just isolating 2 and 1 observations, respectively. All the other hierarchical clustering methods performed poorly as well. HOSil has produced higher ASW values as compared to PAMSIL.

Model 10 has five clusters each coming from χ^2 , F , t , N and SN distributions. Single, Ward's, McQuitty and spectral clustering were not able to return the desired clustering results (Figure B.12). These methods have produced smaller ARI values as compared to other methods (see Table 3.17). PAMSIL has produced higher ASW and ARI values as compared to HOSil for both fixed and estimated number of clusters.

Table 3.16 Results for Model 9.

Methods	fixed k		
	ASW	SE	ARI
true	0.6295		
k-means	0.6417	0.0018	0.9679
PAM	0.6415	0.0018	0.9674
single	0.2994	0.0396	0.5652
complete	0.5500	0.0123	0.7760
average	0.6367	0.0022	0.9548
Ward's	0.5421	0.0132	0.7798
McQuitty	0.5421	0.0132	0.7798
Spectral	0.5369	0.0274	0.8875
model-based	0.6400	0.0019	0.9769
PAMSIL	0.6329	0.0035	0.9733
HOSil	0.6374	0.0023	0.9546

Methods	estimated k		
	ASW	SE	ARI
PAMSIL	0.6341	0.0032	0.9323
HOSil	0.6395	0.0020	0.8611

Table 3.17 Results for Model 10.

Methods	fixed k		
	ASW	SE	ARI
true	0.8227		
k-means	0.8251	0.0014	0.9857
PAM	0.8250	0.0014	0.9834
single	0.6887	0.0109	0.8159
complete	0.7704	0.0141	0.9158
average	0.8205	0.0030	0.9766
Ward's	0.8061	0.0066	0.9573
McQuitty	0.8061	0.0066	0.9573
Spectral	0.5877	0.0318	0.8210
model-based	0.7645	0.0078	0.9405
PAMSIL	0.8259	0.0016	0.9863
HOSil	0.8240	0.0016	0.9845

Methods	estimated k		
	ASW	SE	ARI
PAMSIL	0.8259	0.0016	0.9863
HOSil	0.8240	0.0016	0.9845

Model 11 has six clusters formed from Weibull, Exponential, skew-Gaussian, Gamma, non-central Beta and Uniform distributions represented by the labels 1 to 6 respectively, displayed in Figure B.13 panel (b). Single linkage has combined clusters number 3 and 5 and has divided cluster number 4 into two clusters. Complete, average, Ward's and Mcquitty have combined clusters 1, 2, 3 and 5 by splitting cluster 4 into four clusters. k -means, PAM and spectral methods have combined clusters 1 and 2 and have divided cluster 4 into two clusters. Model-based clustering returns a model with 5 components by considering cluster 1 and 2 as a single component. Only HOSil and PAMSIL have estimated the correct number of clusters and correct solution for clustering (refer to Figure B.13). In terms of ASW and ARI both HOSil and PAMSIL gave very close values with HOSil doing slightly better in terms of ASW value for estimated k (Table 3.18). Both HOSil and PAMSIL performed much better than all other methods in terms of ARI values.

Model 12 has 6 correlated Gaussian clusters. Only k -means, model-based clustering and HOSil $_k$ have discovered the true clustering as depicted in Figure B.14. Many clustering methods have combined the two closely located Gaussian clusters into one cluster and have divided one of the other four clusters into two clusters. Model-based clustering method performed best in terms of ARI values among all methods (Table 3.19). After this average linkage, HOSil and PAMSIL has performed good in terms of ARI values. HOSil has produced higher ASW value but lower ARI values as compared to PAMSIL.

Table 3.18 Results for Model 11.

Methods	fixed k		
	ASW	SE	ARI
true	0.7479		
k-means	0.7213	0.0013	0.7710
PAM	0.7445	0.0018	0.9598
single	0.5654	0.0098	0.5967
complete	0.5738	0.0026	0.2982
average	0.5798	0.0050	0.3755
Ward's	0.5730	0.0041	0.2938
McQuitty	0.5730	0.0041	0.2938
Spectral	0.6425	0.0164	0.7932
model-based	0.7330	0.0018	0.8176
PAMSIL	0.7485	0.0019	0.9966
HOSil	0.7478	0.0022	0.9888

	estimated k		
	PAMSIL	SE	ARI
PAMSIL	0.7488	0.0018	0.9942
HOSil	0.7493	0.0016	0.9924

Table 3.19 Results for Model 12.

Methods	fixed k		
	ASW	SE	ARI
true	0.6392		
k-means	0.6406	0.0021	0.9655
PAM	0.6412	0.0021	0.9539
single	0.5400	0.0124	0.7008
complete	0.5355	0.0096	0.7182
average	0.6299	0.0048	0.9385
Ward's	0.5605	0.0099	0.7916
McQuitty	0.5605	0.0099	0.7916
Spectral	0.5305	0.0203	0.8819
model-based	0.6099	0.0077	0.9853
PAMSIL	0.6502	0.0027	0.8962
HOSil	0.6634	0.0023	0.8914

	estimated k		
	PAMSIL	SE	ARI
PAMSIL	0.6735	0.0025	0.9110
HOSil	0.6814	0.0024	0.8944

Model 13 has 14 Gaussian clusters each having different covariance matrices. HOSil has shown best performance for fixed k and estimation of k for ASW and ARI values among all methods considered. After HOSil, single, average, and PAM methods have also performed close to it. The clustering results are displayed in Figure B.15. Model-based, PAMSIL, complete, and spectral clustering performed poor here for both fixed and estimated k . Table 3.20 shows the numerical results for this model.

Model 14 has 3 clusters. All the methods included in the study were able to retrieve desired clustering solution except model-based clustering for fixed k . Figure B.16a represents the clustering result obtained by all clustering methods except model-based clustering. Model-based clustering was not able to determine any clustering results for this model for fixed k therefore ASW and ARI can not be calculated. It has always estimated 1 number of clusters. All clustering methods gave same ASW with ARI=1 except spectral clustering method which performed bad (see Table 3.21).

Model 15 has 9 clusters. Figure B.16b depicts the clustering results found by k -means, PAM, single, average, model-based, and HOSil clustering methods. These meth-

Table 3.20 Results for Model 13.

Methods	fixed k		
	ASW	SE	ARI
true	0.4331		
k-means	0.7369	0.0042	0.9549
PAM	0.7541	0.0016	0.9969
single	0.7508	0.0037	0.9958
complete	0.5930	0.0104	0.7710
average	0.7498	0.0029	0.9899
Ward's	0.7187	0.0078	0.9295
McQuitty	0.7187	0.0078	0.9295
Spectral	0.5371	0.0199	0.7855
model-based	0.4354	0.0228	0.6396
PAMSIL	0.6280	0.0030	0.7316
HOSil	0.7568	0.0017	0.9985

	estimated k		
	PAMSIL	SE	ARI
PAMSIL	0.6460	0.0025	0.7207
HOSil	0.7579	0.0016	0.9944

Table 3.21 Results for Model 14.

Methods	fixed k		
	ASW	SE	ARI
true	0.8111		
k-means	0.8111	0.0023	1
PAM	0.8111	0.0023	1
single	0.8111	0.0023	1
complete	0.8111	0.0023	1
average	0.8111	0.0023	1
Ward's	0.8111	0.0023	1
McQuitty	0.8111	0.0023	1
Spectral	0.6971	0.0364	0.9105
model-based	NA	NA	NA
PAMSIL	0.8111	0.0023	1
HOSil	0.8144	0.0023	0.9992

	estimated k		
	PAMSIL	SE	ARI
PAMSIL	0.8111	0.0023	1
HOSil	0.8144	0.0023	0.9992

ods were able to retrieve the correct clusterings with ARI=1 (Table 3.22). Complete, Ward, McQuitty, and PAMSIL also performed well.

Model 16 The true clustering for this model in two dimensions are depicted in Figure B.17a and numerical results are presented in Table 3.23. All the methods gave correct clustering results as well as ARI=1 except k-means (ARI=0.9775), model-based (ARI=0.8945) and spectral methods (ARI=0.8045).

Model 17 The graphical and numerical clustering results are presented in Figure B.17b and Table 3.24, respectively. All the clustering methods performed well and produced ARI=1 except PAMSIL and HOSil for the estimation of number of clusters. These two methods have always estimated 2 number of clusters resulting in a smaller ARI than 1. However, they were able to produced the desired clustering results for the fixed k always, resulting in the ARI=1.

Table 3.22 Results for Model 15.

Methods	fixed k		
	ASW	SE	ARI
true	0.8036		
k-means	0.8036	0.0011	1
PAM	0.8036	0.0011	1
single	0.8036	0.0011	1
complete	0.8014	0.0017	0.9964
average	0.8036	0.0011	1
Ward's	0.8033	0.0011	0.9993
McQuitty	0.8033	0.0011	0.9993
Spectral	0.6337	0.0202	0.8810
model-based	0.8036	0.0011	1
PAMSIL	0.8037	0.0011	0.9983
HOSil	0.8062	0.0011	1

	estimated k		
	PAMSIL	SE	ARI
PAMSIL	0.8037	0.0011	0.9983
HOSil	0.8062	0.0011	1

Table 3.23 Results for Model 16.

Methods	fixed k		
	ASW	SE	ARI
true	0.9230		
k-means	0.9101	0.0042	0.9775
PAM	0.9230	0.0027	1
single	0.9230	0.0027	1
complete	0.9230	0.0027	0.9998
average	0.9230	0.0027	0.9998
Ward's	0.9230	0.0027	0.9998
McQuitty	0.9230	0.0027	0.9998
Spectral	0.5846	0.0315	0.8045
model-based	0.8737	0.0028	0.8945
PAMSIL	0.9230	0.0029	1
HOSil	0.9083	0.0027	0.9995

	estimated k		
	PAMSIL	SE	ARI
PAMSIL	0.9230	0.0029	1
HOSil	0.9083	0.0027	0.9995

Table 3.24 Results for Model 17.

Methods	fixed k		
	ASW	SE	ARI
true	0.3299		
k-means	0.3299	2e-04	1
PAM	0.3299	2e-04	1
single	0.3299	2e-04	1
complete	0.3299	2e-04	1
average	0.3299	2e-04	1
Ward's	0.3299	2e-04	1
McQuitty	0.3299	2e-04	1
Spectral	0.3299	2e-04	1
model-based	0.3299	2e-04	1
PAMSIL	0.3299	2e-04	1
HOSil	0.3299	2e-04	1

	estimated k		
	PAMSIL	SE	ARI
PAMSIL	0.3834	2e-04	0.5673
HOSil	0.3855	2e-04	0.5673

3.5.2 Performance for the estimation of k

In this section comparisons of different clustering method and validation indices for the estimation of number of clusters is made. The results are presented in Appendix B.2. The table represents the counts obtained for estimated k from 1-15 from each

combination of clustering method and validation indices considered in the study. The percentage performance rate (PPR) is reported throughout in this section for comparison which is calculated by dividing the count for desired value which is the true known k by total data sets generated which is $B=50$ or 25 (for Model 13 only).

Model 1 Table B.1 represents the number of clusters estimated by various clustering methods and various indices used. ASW (complete, average, k -means, PAM, spectral, model-based), Jump (with $p/3$), PS (with k -means), BI (with k -means), PAMSIL, and HOSil were consistent in the estimation of the number of clusters in all runs. The performance of Hartigan's method was not good.

Model 2 has two clusters. The results for the estimation of the number of clusters are given in Table B.2. H and KL have lowest performance rate. The Jump method is also not good, as it has estimated the numbers of clusters as 2, just 16 times out of 50 runs. CH and BI also have low overall performance rate. PS (98%) and BI (94%) have also performed well just with k -means. The maximum percentage of correct estimation of the number of clusters for Gap (with PAM) was 84%. ASW (with k -means, PAM, spectral, model-based), model-based clustering (with BIC), PAMSIL and HOSil have PPR at 98%, 100%, 100% and 98% respectively.

Model 3 has three clusters. The counts received for all the methods are displayed in Table B.3. None of the indices included suggested 3-clusters solution very often. In fact, the maximum number of choices for 3-clusters was obtained from PS (single linkage:64%, PAM: 62%), and H (with k -means:48%). All the methods were in favour of the 2-cluster solution here.

Model 4 (Table B.4) has three Gaussian clusters, where the bigger cluster (in size and spread) is located between two smaller and compact clusters. H, CH and KL perform poorly. Gap, PS and BI performed very bad with all clustering methods except k -means. ASW (average, k -means, PAM, model-based), Jump ($p/3$, $p/4$), PS (k -means), BI (k -means), model-based, PAMSIL, and HOSil showed 100% PPR. Gap with average linkage have a PPR of 88%. All the other choices had much lower PPR than 50%.

Model 5 (Table B.5) has 3 Gaussian clusters of unequal sizes and variations. Only HOSil has a better PPR(96%) here. Gap (using PAM), model-based (with BIC), ASW (using PAM), and PAMSIL gave 80%, 65%, 64%, and 63% PPR, respectively. H, CH, KL and Jump performed poorly here.

Model 6 (Table B.6) has three clusters of different variations among observations and of equal sizes. ASW (with k -means and PAM), PS (with k -means and PAM), BI (with PAM), model-based (with BIC), PAMSIL and HOSil performed well.

Model 6.A (Table B.7) only Gap (k -means), PS (Ward, k -means and PAM), and model-based clustering suggest the three cluster solution with higher percentages. ASW with all the clustering methods, PAMSIL, and HOSil suggest two clustering solutions by combining cluster 1 and 2 together.

Model 6.B (Table B.8) ASW, PAMSIL, and HOSil suggest number of clusters three

instead of four majority of times. Adding a new cluster did not help in identification of cluster one and two in Model 6.B. However, PS (Ward, k -means, PAM), BI (PAM) and model-based (with BIC) clustering suggest a 4-clusters solution with 90%, 84%, 100%, 98% and 100% of the time, respectively.

Model 7 has four very close clusters. H, KL and Gap methods did not perform well. Overall, the performance of estimation methods with clustering methods was good here. Many combinations achieved 100% PPR for this model.

Model 8 has four clusters. Most of the methods were not able to suggest 4-cluster solution. Only model-based clustering found a 4-cluster solution for 66 times out of 100 for this model. The inclinations of all the other methods were towards two and three cluster solutions (results reported in Table B.10). PAMSIL (28 counts) and HOSIL (34 counts) is in favour of 2-cluster solution. ASW also suggest 2-cluster solutions with all the methods with majority counts.

Model 9 (Table B.11) has 4 correlated Gaussian clusters. Many methods failed to estimate 4-cluster solution here. Only CH (complete: 92%), Jump ($p/3/p/3$: 100%), PS (k -means: 70%), model-based clustering (BIC: 100%), ASW (Complete:80%, PAM:86%, model-based:94%), PAMISL(90%), and HOSil (80%) were able to estimate 4-cluster solution here.

Model 10 (Table B.12) has five clusters. H, CH, KL, and model-based (BIC) performed poorly. ASW (single, complete, spectral), Gap (single, complete, Ward, k -means, PAM), PS (single, k -means), BI (single, k -means) performed poorly. Only Jump, PS (Ward, PAM), PAMSIL, and HOSIL gave 100% PPR.

Model 11 (Table B.13) has 6 clusters. H, CH, KL, ASW (except PAM), Gap, BS, BI, PS (except PAM) have no high preference for any single number of clusters. They have estimated a range of different numbers of clusters. Jump($p/3$) suggests 6 numbers of clusters 40 times out of 50. Model-based clustering with BIC suggests the 5-cluster solution 45 times. ASW with model-based clustering have always estimated 5-cluster solution (50 times) and with PAM clustering it has favoured 6-cluster solution (40 times). HOSil suggests the 6-cluster solution 48 times and PAMSIL 47 times.

Model 12 (Table B.14) has six Gaussian clusters where the variables used to generate 2 dimensions within clusters are correlated. H, CH and KL have no high preference for any number of clusters. ASW, PAMSIL, and HOSil suggest a five clusters solution. Only model-based (BIC) clustering and Gap (average, PAM) suggest a six clustering solution 36, 32 and 36 times out of 50, respectively.

Model 13 (Table B.15) has 14 clusters. Many methods failed entirely to estimate the number of clusters as 14 here including H, KL, Gap, PS, BI, model-based (BIC) and PAMSIL. Among other methods, only few combinations performed well. CH (single, average), ASW (single, average) and HOSil have estimated the correct number of clusters, and the PPR for the correct estimation were 96, 92, 92, 88, and 84, respectively. Gap, PS and BI suggested the numbers of clusters to be two for a majority of times.

Model-based clustering has estimated 8 and 9 clusters 5 and 20 times, respectively.

Model 14 has three elongated clusters shown in Figure B.16a. The estimation of the number of clusters is shown in Table B.16. H, CH and KL were not able to estimate three clusters except a very few times. Jump($p/7$) has estimated three clusters 47 out of 50 times. ASW, PS, BI, PAMSL, and HOSil estimated the correct number of clusters with all the clustering methods included for a majority of the times. Model-based clustering with BIC always suggested a 1-cluster solution for this model.

Model 15 has 9 clusters shown in Figure B.16b. CH (only with single linkage), Gap (single), Jump, PS (only with Ward and PAM), BI (only with Ward and PAM), model based (BIC) clustering, ASW (except with k -means and spectral), PAMSIL, and HOSil have 100% PPR. H, KL, CH and Gap with other clustering methods were not able to estimate nine clusters a majority of times. The complete results for the estimation of the number of clusters are given in Table B.17.

Model 16 has ten clusters in 100 dimensions. Many methods failed to estimate correct clusters here. The Jump method could only estimate 10 clusters 16 times out of 50 with transformation power $p/2$. The maximum number of times CH estimated 10 clusters was only 20 with single linkage. KL could not estimate the numbers of clusters to be 10 even once with any of the clustering methods included. Gap estimated 14 clusters mostly. BI always estimated 2 clusters. Only PS (except with k -means), model-based with BIC, ASW (except with k -means and spectral), PAMSIL, and HOSIL have always estimated 10-cluster solution.

Model 17 has three clusters in 1000 dimensions. Gap (single, average, McQuitty), and model-based clustering with BIC estimated the correct number of clusters always. BI (complete linkage) and PS (Ward) have also performed well with 96% and 82% PPR, respectively. H, CH, KL, ASW, PAMSIL and HOSil suggested two clusters for this model. Jump always estimated 15 clusters. See Table B.19 for complete results.

Summary The performance of the H and KL indices was not good for the majority of the models included. The CH method performed well only for very few models and that was only for one or two clustering methods. The Jump method also estimated correct number of clusters for a very few models (never for Model 3, 13, 17) and very low PPR for (Model 2, 5, 6.A/6.B, 8, 12, 16). The results for the Gap method were below average for the majority of the models included in the analysis. The BI and PS also performed badly for majority of the models except for a few of them with one or two clustering methods only. Model-based clustering has estimated \hat{k} other than expected for Models 3, 5, 8, 10, 11, 12, 13 and 14. HOSil clustering also estimated k other than the true k for Models 3, 6.A/B, 8, 12 and 17. PAMSIL was also not able to estimate the correct number of clusters for all the models mentioned for HOSil and in addition Model 13 and had much smaller PPR for Model 5.

3.6 Further exploration

As found in the previous section, HOSil has shown potential for retrieving the correct clustering aligned with the DGPs for the known k case and has shown good performance for the estimation of the true k for the majority of the models. We have also explored other models than the ones considered in the previous section in order to dig in more into what other kind of clustering structures HOSil can identify. We don't intend to give all of these models for brevity sake and only a few more datasets are given below followed by the discussions on them. The selection of these models has been made keeping in mind to provide insight not only about what other types of clustering challenges HOSil can achieve but also to make reader aware of the models where HOSil will fail.

Model 18:

Five clusters in two dimensions: Four clusters with I_2 covariance matrix each having 50 observations were generated from a Gaussian distribution centred at $(0, 8)$, $(8, 0)$, $(0, -8)$, $(-8, 0)$. 50 realizations from a uniform distribution were generated in the interval $(-2, 2)$ along both dimensions independently. The resulting clusters look like one square-shaped cluster surrounded by 4 spherical clusters.

Model 19:

7 Gaussian clusters were generated through independent variables in 2 dimensions. The data set contains a Gaussian cluster of size 100 with mean $(0, 0.1)$ and covariance $\begin{bmatrix} 0.3 & 0 \\ 0 & 0.1 \end{bmatrix}$. The remaining 6 clusters contain 50 observations each with covariance matrix as $\Sigma = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.3 \end{bmatrix}$. The clusters are centred at $(-1, 2.5)$, $(-1, 7.5)$, $(0, 2.5)$, $(0, 7.5)$, $(1, 2.5)$ and $(1, 7.5)$.

Model 20:

9 equally sized differently shaped clusters in 2 dimensions. Four elongated clusters are added to the corners of Model 18. For this four different Gaussian bi-variates were added to a uniformly generated variable. Let x and y denote the first and second dimensions of clusters and $v \sim \mathbb{U}(-1, 1)$. Generate cluster 1 as: $x = y \sim v + N(-7, 0.1)$, cluster 2 as: $x = y \sim v + N(7, 0.1)$, cluster 3: as $x = v + N(-7, 0.1)$, $y = v + N(7, 0.1)$, and cluster 4 as: $x = v + N(7, 0.1)$, $y = v + N(-7, 0.1)$. Each cluster contains 50 observations. This model was designed by combining the situations in Model 14 and Model 18 to see will HOSil also be able to handle these situations combined.

Four Shapes:

200 observations were generated from the “mlbench.shapes” data set from the R package “mlbench” (version: 2.1.1, [Leisch and Dimitriadou \(2010\)](#)). The data set contains four different shapes lying very close to each other. The four shapes are a Gaussian, square, triangle and a wave in two dimensions. The purpose is to check whether HOSil can identify each shape or not.

The following data sets are also freely available and taken from the Fundamental Clustering Problems Suite ([Ultsch \(2005\)](#), FCPS)

Diamonds:

The data set contains two clusters of diamond shape in 2 dimensions and has 800 observations. Note that we have just included 400 observations to reduce the runtime of the algorithm. Each of the clusters has 200 observations. The clusters are defined by the densities. The purpose is to check whether HOSil can identify two similar clusters separately if such a situation arises in real life.

Tetra:

The data set contains 4 clusters in three dimensions and has 400 observations. The clusters are almost touching and have no clear separation between them.

These data sets are shown in Figures 3.3 and 3.4.

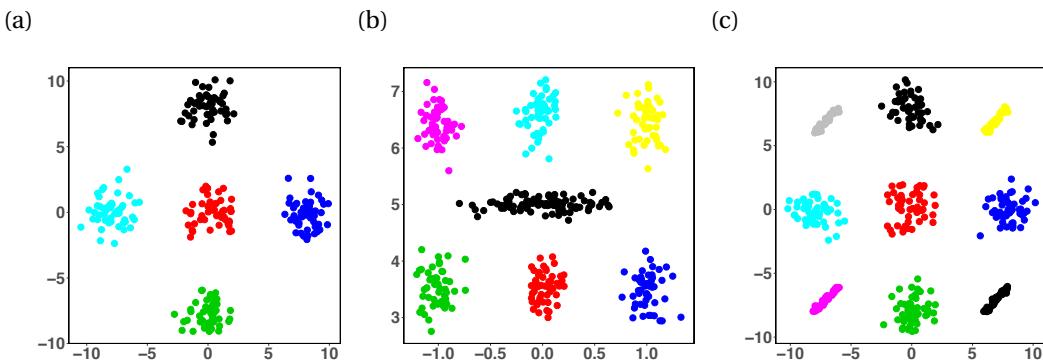


Figure 3.3 Data plots for (a) Model 18, (b) Model 19, and (c) Model 20.

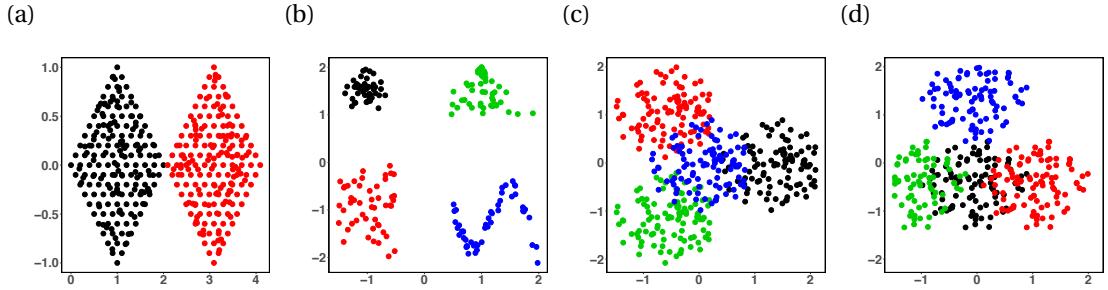


Figure 3.4 Data plots for (a) Dimonds data, (b) Four Shapes data, and (c) Tetra data dimensions 1 and 2, and (d) Tetra data dimensions 1 and 3.

For the dataset defined above, HOSil returns the expected clustering results with an advantage of estimating the numbers of clusters correctly itself. Figure B.18 depicts the clustering results obtained by HOSil against estimated k for Model 18, 19 and 20. For all of these models HOSil was successful in capturing the clustering results as well as the estimation of number of clusters as defined by DGPs. All the other existing clustering methods also returned the correct clustering with the correct estimated k except model-based clustering for the Four Shapes data set(see Figure B.19c), which does not agree with the desired clustering identified by the other methods.

We now define some of the data sets to give an idea what kind of clusterings HOSil cannot identify. HOSil was not always successful in giving the expected clustering result that comply with DGPs for the fixed k case, and this is also true for the existing clustering methods. The data sets are first defined below followed by the discussion on them.

Model 21:

13 clusters in 2 dimensions. 100 observations are generated from the non-central t distributions independently parameterized as $t_{25}(5)$ and $t_{25}(10)$. 12 clusters are generated from Gaussian distribution each having 25 observations and common covariance matrix as $0.1^2 I_2$. The clusters are centred at $(0, 5)$, $(0, 8)$, $(0, 20)$, $(0.5, 0.3)$, $(2, 13)$, $(2, 17)$, $(4, 5)$, $(7, 18)$, $(8, 5)$, $(8, 15)$, $(10, 5)$ and $(10, 20)$.

Smiley:

200 observations were generated from the “mlbench.smiley” data set from R "mlbench" package. The data set contains two Gaussian eyes, one trapezoid nose and a parabolic mouth. The purpose was to check whether HOSil can identify different parts of the face included each as a cluster.

Lsun:

The data set contains 3 clusters and 400 observations in two dimensions. The clusters differ in their variance and have unequal inter cluster distances. This data set is from FCPS ([Ultsch \(2005\)](#)).

Aggregation

The data set has 7 clusters in 2 dimensions and contains 788 observations. The dataset is taken from [Gionis et al. \(2007\)](#). All the seven clusters have different shapes, have different cluster diameters, sizes and cluster distances from each other. The four data sets are displayed in Figure 3.5.

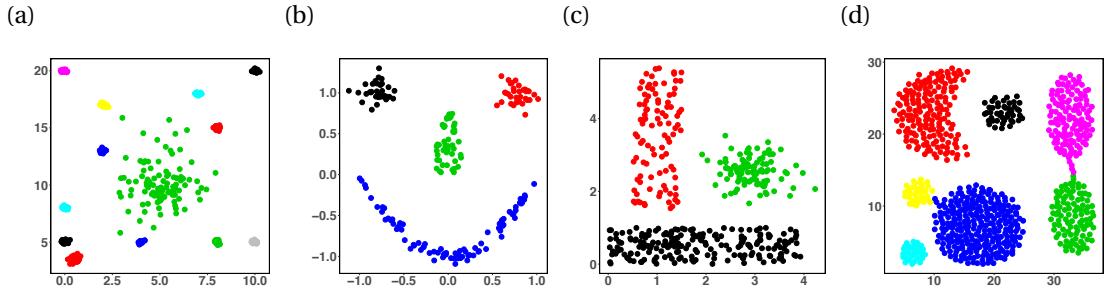


Figure 3.5 Data plots for Model 21, Smiley, Lsun, and Aggregation data sets.

For Model 21 (Figure B.20), many clustering methods have split the bigger central cluster into 2 or 3 smaller clusters. On the other hand they have also combined smaller clusters into one bigger cluster. For the Lsun data (Figure B.21), only single linkage was able to capture desired clustering. For Aggregation data, HOSil estimated 5 as number of clusters by putting the three spherical balls into one cluster. Only average linkage was able to retain the desired clustering here (see Figure B.22). Figure B.23 shows the clustering results for the Smiley data set. Only single linkage, spectral clustering and HOSil_k were able to return the Smiley data classification. HOSil did not estimate the correct number of clusters here.

A common characteristic in Model 21, Smiley, and Aggregation data is they don't have same within cluster distances. In order to get a good ASW value, the within-cluster dissimilarities should be small as compared to between clusters dissimilarities. In case of wide spread of observations within clusters i.e., large distances between clusters the small within-cluster dissimilarities requirement of ASW index dominate and it splits this bigger cluster into smaller clusters such that rather than having one cluster with bigger within cluster distances it prefers to split this cluster into nicely compacted smaller clusters.

One of the major results of the simulations was that for some data models the ASW based clustering optimiser function can produce the desirable clusters but might not estimate that number of clusters, for instance Model 3, 6.A, 6.B, 8, 12 and smiley data sets. Which is surprising result because ASW is in extensive use for the estimation of number of clusters rather than for finding clustering.

3.7 Complexity Analysis

The time complexity of calculating the Euclidean distance between a pair of points having p dimensions has a linear time complexity, i.e., $O(p)$ for two unique observations in data say $X_1 = (x_{11}, \dots, x_{1p})$ and $X_2 = (x_{21}, \dots, x_{2p})$. There are total $n(n - 1)/2$ unique pair of points in a data sets of n points such that the total complexity is $O(pn(n - 2)/2)$. HOSil takes as an input the pairwise dissimilarities between the objects to cluster. Therefore, this complexity doesn't add to HOSil's complexity. We now calculate the complexity of HOSil as below.

For a data set of size n to build the full hierarchy there will be n levels in the hierarchy. Let the level of hierarchy is denoted by l . Starting from the bottom at level $l = 1$ there will be n clusters. At level $l = 2$ there will be $n - 1$ clusters, so on such that when we reach near the top, at level $l = n - 2$ there will be three clusters, at level $l = n - 1$ there will be two clusters and finally, at level $l = n$ there will be just one cluster. For ASW optimization there should be at least two points in one cluster therefore we can not start at $l = 1$ and there should be at least two clusters in a clustering therefore, we must stop at $l = n - 1$ when there are two clusters. The total hierarchy where operations will actually take place are $l = 2, \dots, l = n - 2$.

At each hierarchy level there are certain combinations to be checked. These are, $\binom{n-1}{2}$, $\binom{n-2}{2}$, $\binom{n-3}{2}$, \dots , $\binom{3}{2}$, $\binom{2}{2}$ for hierarchy levels $l = 2, l = 3, \dots, l = n - 3, l = n - 2$. This can be summarized as $(n-3) \binom{n-i}{2}$, $i = 1, 2, \dots, (n-3)$. The time complexity is equal to the maximum number of combinations there are to calculate. This gives a time complexity of $O(n^n C_2)$ (ignore the constant terms gives the maximum complexity). The highest complexity that comes from here is $(n-3) \binom{n-1}{2}$. As data size n will grow the combination will grow, and hence the complexity will grow.

At each hierarchy level for each of the combinations the algorithm has to perform a set of different operations which involves nested loops. The algorithm has main function that calls other functions which involves a sorting algorithm (quadratic time $O(n^2)$), and four set of nested loops used to implement the ASW formula. Three sets out of these four set of loops has 2 nested loop hence Quadratic time $O(n^2)$ complexity. The only remaining set has 3 nested loops hence the cubic complexity. Note that these four set of loops have "if" and "if-else" statements at different levels which are only assignment statements thus time complexity is constant $O(1)$ and they don't contribute towards the reduction in the nested loops' complexity. These are ignored because the

smaller or constant complexities are ignored in presence of higher complexities. Aggregating these becomes $4O(n^2) + O(n^3)$. Multiplying this with the earlier complexity we get the total complexity of the algorithm as $O(n^n C_2)[4O(n^2) + O(n^3)]$. Which simplifies to $O(n^n C_2)O(n^3) \Rightarrow O(n^{4^n} C_2)$.

3.7.1 Runtime complexity

The proposed method takes more time than the other clustering algorithms considered in this work. Since there are lots of nested loops involved, we have implemented the algorithm in C++ and have provided an interface for its call in the R language ([R Core Team \(2015\)](#)). We have compared the runtime of the proposed method with the existing algorithms for all the simulated data sets of various sizes, dimensions and numbers of clusters used in the simulation. The computations were done using the UCL super computing facility Legion, and MacBook Pro 2.8 GHz i7 processor with a 16 GB of RAM memory. Table 3.25 contains the runtime of HOSil. Note that only time taken by the proposed algorithm is mentioned, however, the time for each method was recorded. The time reported here does not include the time for distance calculations. Although the proposed algorithm's performance for retrieving an accurate clustering is good with the advantage that it can also estimate the number of clusters provided the time it takes. It is not good for the data sets with more than (500-800) observations. Thus, the algorithm is only useful for the smaller data sets. However larger p is not a problem for the algorithm (for instance see data Model 16 and 17) because it's not represented in distances. The algorithm is implemented in such a way that if k is known, then the algorithm can be stopped at the desired hierarchy level. It is worthwhile to look for some fast approximation to improve the algorithm's performance potential for the data sets of bigger sizes. In the next section we make a proposal with the potential of reducing the computational time complexity of the HOSil algorithm.

3.8 A faster approximation

One immediate suggestion could be to combine the hierarchical clustering with partitioning methods to reduce the computational cost. The idea is to first partition the data for very large k from a partitioning clustering algorithm and then from there start building the hierarchy to take advantage of this method. Since partitioning methods take less time, they should be computationally more efficient than HOSil for relatively bigger data sets. The choice of k depends upon how large we can go such that its affordable for HOSil to build hierarchy. We have noticed from the simulation that building hierarchies based on HOSil for $n = 200$ takes reasonable time. Thus for a data set of larger n say $n = 500$ if we first cluster the data from any partitioning clustering method for instance, k -means clustering for very large number of clusters say $v = 200$ and then

Table 3.25 Time taken by various models for HOSil clustering algorithm for simulated data sets.

DGP	<i>k</i>	<i>n</i>	<i>p</i>	runtime
Model 1‡	2	200	2	11m*
Model 2‡	2	150	2	2.5m*
Model 3‡	3	200	2	10.6m*
Model 4	3	200	2	10m*
Model 5	3	200	2	2.6m*
Model 6	3	250	2	35m*
Model 6.A	3	250	2	4.4m*
Model 6.B	4	300	2	19m *
Model 7‡	4	200	2	11.6m*
Model 8‡	4	200	2	10m*
Model 9	4	200	2	19m*
Model 10	5	250	2	32.5m*
Model 11	6	300	2	2.7h*
Model 12	6	250	2	1h*
Model 13	14	350	2	6h**
Model 14	3	300	3	4m*
Model 15‡	9	233	3	43m*
Model 16	10	250	500	1h*
Model 17‡	3	120	1000	1.4m*
Model 18	5	250	2	36m
Model 19	7	400	2	13h
Model 20	9	450	2	24h
Model 21	13	400	2	12h
Four Shapes	4	200	2	1.6h
Diamonds	2	400	2	11h
Tetra	4	400	3	12h
Smiley	4	200	2	11m
Lsun	3	400	2	1.6h
Aggregation	7	788	2	5h†

m = minutes, h = hours.

* represents run time averaged over $B = 50$ runs.

** represents run time averaged over $B = 25$ runs.

The values without * in last column represent time only for single data set.

The Models with ‡ in first column represent the simulations that were done on MacBook 2.8 GHz i7 processor. Simulations for all other models were run on Legion.

† represents time taken by PAM with $v = 400$. See the next section for more details. Calculations only for this data set was done using fast version of HOSil.

Note that R returns a runtime in seconds. The reported run time here is subject to approximation to minutes and hours.

start building hierarchy from 200 clusters, can result in reasonable time. Note that we want to keep v as close to n as possible to take maximum benefit of HOSil, but we need to find the cutoff where HOSil is computationally not too painful.

One benefit of HOSil's implementation code is that it can start building hierarchies from any level. It just needs the corresponding clustering label vector to begin. One can do the initial clustering through some partitioning clustering algorithm to reduce the computational cost in terms of time and then build a hierarchy on top of it to see what observations go in which cluster based on hierarchical clustering. In principle any partitioning clustering algorithm can be used. We have used PAM because of its flexibility to use with any distance measure.

The model used for an experiment here was the same as that of Model 7 with a difference that 100 observations were generated from each cluster such that $n = 400$, $k = 4$, $p = 2$. Let the new model be called Model 7.B. We have checked the reduction in time for various values of v . Table 3.26 gives the experimental results for Model 7.B. First the time taken by PAM for v number of clusters is reported. After getting a clustering with v clusters, the HOSil algorithm was applied to construct hierarchy. The time taken by HOSil is reported in the next column. Note that the first row in the table represents the time taken by direct calculation from HOSil. These experiments were performed on Legion.

Table 3.26 Time taken by Model 7.B for different values of v .

v	Runtime	
	PAM	HOSil
-	-	13.6h
150	3.563s	40m
200	3.281s	1.62h
250	2.371s	3.4h
300	1.141s	6.3h

Note that the experiments for four values of v were done independently. More research is needed to explore how combining PAM and HOSil will effect the clustering results and optimum value of ASW. An important point to note here is, some of the nodes of Legion are very old (for a detailed specification of nodes please see UCL supercomputing website) and take much longer than needed. Therefore, there can be significant decrease in the time reported here. For instance, Aggregation data only took 5 hours (see Table 3.25) to cluster with a much greater model complexity i.e., $v = 400$, $n = 788$ and $k = 7$, whereas Model 7.B took 6.3h to cluster with the relatively less model complexity as $v = 300$, $n = 4$ and $k = 4$. Clearly there will be significant reduction to the time taken which we have noticed here. Since a user can't control his submitted job will be

allocated to which Legion's node and is this node old or new. Due to these reasons it should not be surprising that in the Table 3.26 PAM took much less time to produce a clustering for 300 clusters than for 150 clusters. Indeed the latter one should take lesser time. Thus the Table 3.26 just gives a rough idea of the reduction in time by using PAM first to skip the hierarchy from the bottom level and to raise the starting hierarchy level to v .

3.9 Applications

In this section we will apply the proposed clustering methodology to real life applications.

3.9.1 Tetragonula bee data clustering

We here considered the data set by [Franck et al. \(2004\)](#) to find the number of clusters. The data set is available for free download from the international federation for the classification society (IFCS) Cluster Benchmark Data Repository¹. The data set is the taxonomy of 236 species among the Tetragonula bees. The dataset gives the genetic information of these bees at 13 microsatellite loci from eastern Australia and between Indian and Pacific Ocean. The 13 variables are categorical. Each entry of these 13 variables is a string of a 6 digit code representing pairs of alleles for microsatellite loci. The purpose of clustering for this data is to find how many bee species are present. The authors have provided 9 true species of the bees based on morphological information in addition to genetic information, which can be used as the true clustering.

We have applied the HOSil algorithm for the newly proposed ASW based linkage to perform agglomerative hierarchical clustering on the bees' data set for species delimitation. The distance measure used was the "shared allele dissimilarity" particularly designed for calculation of genetic dissimilarities between species by [Bowcock et al. \(1994\)](#). The distance measure is implemented through the package 'prabclus' ([Hennig and Hausdorf \(2015\)](#)) available through the statistical programming language R.

[Hennig \(2014\)](#) lists four possibilities for the required characteristic of clustering for species delimitation as: (1) Small within cluster cluster gaps, (2) Well-separated clusters (depends upon geographical locations), (3) Within cluster's homogeneity, and (4) Cluster stability (see [Hennig \(2014\)](#) for detailed definitions). Since the ASW linkage is based on the concept of cluster separation and compactness, it makes sense to apply this method to the bees dataset for clustering. According to the ASW linkage criterion the best number of species present in the data is 10. The 4 best ASW values obtained with adjusted rand index in bold together with the number of clusters were 0.48406(**0.914795**, $k = 10$), 0.47999(**0.91082**, $k = 9$), 0.47673(**0.834181**, $k = 11$),

¹http://ifcs.boku.ac.at/repository/data/tetragonula_bee/index.html

0.47058(**0.907194**, $k = 8$). The value of the ASW monotonically decreases for the k before and after $k=10$ (see Figure 3.6 for $k = 2$ to 16).

The manual species delimaition by [Franck et al. \(2004\)](#) can't be taken as 100% ground truth as discussed in [Hennig \(2014\)](#). This is because even the experts will not agree on the number of clusters as there is no formal definition of a "species" (see [Hausdorf \(2011\)](#) for subject matter knowledge on the species concepts). [Hennig \(2014\)](#) has also applied average linkage hierarchical clustering using the earlier mentioned distance measure and concluded that the $k=9$ is the best cluster number regarding the second characteristic mentioned above and $k=10$ for the third characteristic.

The heatplot of the data set (shown in Figure 3.7) favours 9, 10 or 11 clusters. From the map it is evident that there are 9, 10, or 11 bee clusters present in the data. A dimensionality reduction method was applied to the dataset to visualize the data in 2-dimensions. Multidimensional scaling (MDS) is a method to visualize the relative positions of the data points using the pairwise distances between them. This is a way to observe the distance matrix directly and interpret the dissimilarity between data points on a low (often 2 or 3) dimensional scatter plot. The classical MDS is available from R base package "stats". The classical MDS plot for the Tetragonula bees data is shown in Figure 3.8 using the true classification provided by [Franck et al. \(2004\)](#) and using the HOSil clustering results.

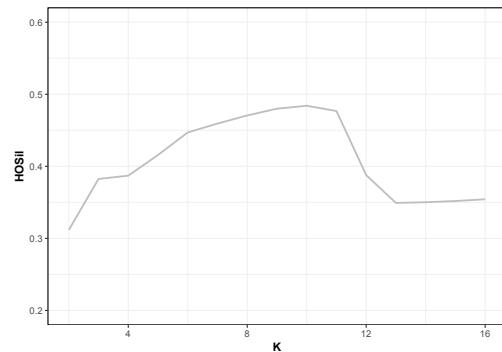


Figure 3.6 ASW value obtained from HOSil for the range of number of clusters from 2 to 16. The maximum value of ASW is obtained at $k=10$.

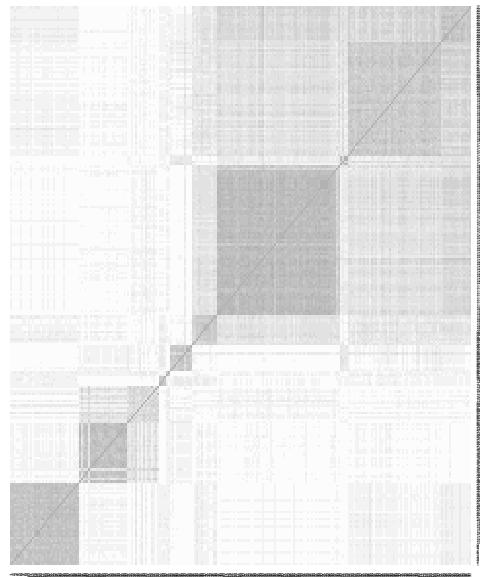


Figure 3.7 Heatplot for the Teteragonula dataset.

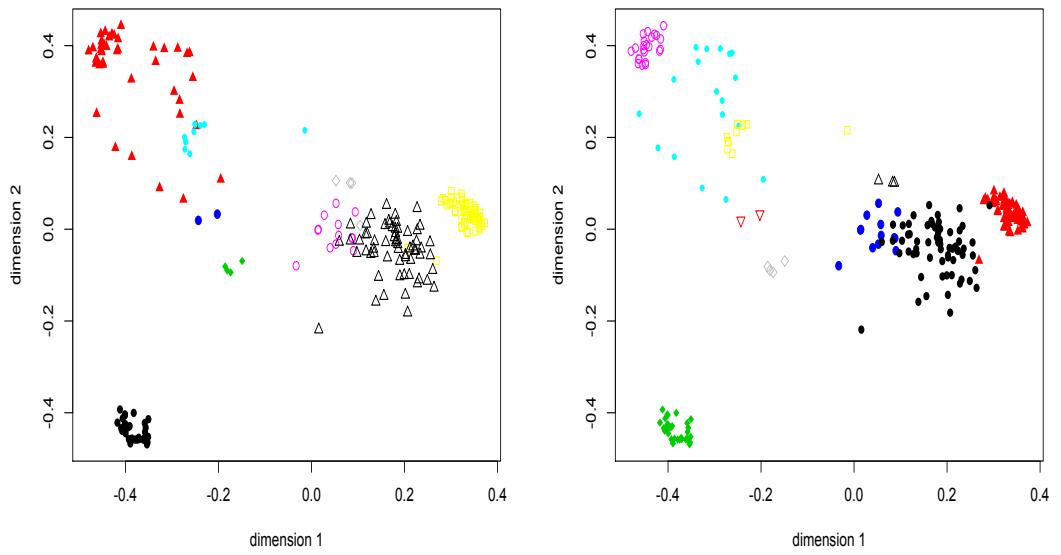


Figure 3.8 The classical multidimensional scaling of the Tetragonula dataset. Left panel represents the species delimitation provided by [Franck et al. \(2004\)](#), and right panel represent the HOSil clustering results.

3.9.2 French rainfall data clustering

Finding spatial or temporal patterns in climate data sets based on statistical techniques is of crucial importance for climatologists. For instance, clustering of earth regions based on similar climate attributes can provide insight about the physical environmental changes, or clustering maxima can provide insight for understanding the causes of the occurrence of extreme rainfall events in weather. Heavy rainfall is a well known extreme weather event. Not all the clustering algorithms can be applied straight away to climate applications. For instance, since k-means makes use of clusters' means for minimizing sum of squares of within cluster distances, it's not suitable for the applications where arithmetic means are not applicable. [Bernard et al. \(2013\)](#) proposed a clustering algorithm based on a combination of PAM algorithm and a distance measure for geostatistics data called the F-madogram for the clustering French weather stations based on maxima of rainfall data. F-madogram is a distance measure for calculation of the pairwise distance among time series of maxima proposed in [Cooley et al. \(2006\)](#). We have considered the data used in [Bernard et al. \(2013\)](#) available through the software they wrote to apply the newly proposed algorithm here for the clustering of french weather into climate regions based on rainfall precipitation maxima observed at the stations. The data is for 92 French weather stations for the three months of fall, from September to November for 19 years. The weekly maxima of hourly precipitation from 1993 to 2011 were considered. The length of each time series used was 288. The purpose of clustering is to find the pattern among stations i.e., spatial clustering.

The application of the HOSil algorithm to this data has more advantages as compared to the algorithm proposed in [Bernard et al. \(2013\)](#). The definition of the distance measure for the maxima of time series is dependent on the generalised extreme value (GEV) family of distributions (see [Cooley et al. \(2006\)](#)). As discussed in [Bernard et al. \(2013\)](#) an important point to note here is that the averages of GEV distributed variables are not GEV distributed and the averages of maxima are not maxima such that F-madogram distance becomes uninterpretable in this situation. Therefore, all the statistical clustering methods based on averages can't be applied here. HOSil does not make use of any kind of cluster representative like centroids for clustering. It works with the individual data points. Therefore, it will deal directly with the time series of maxima rather than any kind of averages of these maxima.

[Bernard et al. \(2013\)](#) have listed three preprocessing steps to perform before applying their algorithm in Section 2 of their paper. The first is the calculation of the pairwise F-madogram distances between time series, the second is to specify the number of clusters and the third is to initialize the set of medoids to run the algorithm. While the R software implementation of the PAM algorithm has a default way of choosing the set of medoids, one still needs to specify the number of medoids. As discussed already the HOSil algorithm just needs the pairwise distances and it can give an estimate of the number of clusters based on the maximum ASW value itself. Thus out of the two

remaining preprocessing steps only one is required for HOSil.

The algorithm is applied to the rainfall data, and spatial locations were taken into account. The underlying philosophy behind the clustering here is that if the local conditions at two weather stations are similar, then the two maxima precipitation series at these stations are not independent and the two weather stations should be in one cluster. For two identical locations the F-madogram distance for these two time series is close to zero, and for locations far away from each other the F-madogram distance is close to 1/6.

The resulting clustering from HOSil algorithm is plotted in Figure 3.9. The results are displayed for 2 to 7 numbers of clusters. For numbers of clusters two the HOSil has classified french weather stations in (clock-wise) the east, south and south-west regions of France together in a cluster (red cluster in Figure 3.9a, say Cluster 1) and from south-west, north up till east in the other cluster (say cluster 2, blue cluster). This is due to the presence of the highest mountain peaks of the Alps in the east and Pyrenees (second highest mountain peaks) in the south of France. The two weather stations in Corsica, which has the third highest mountain peaks, are also classified in this cluster. This is a slightly different clustering result from [Bernard et al. \(2013\)](#) (see Figure 2(a) of their paper). Since PAM looks for equally sized clusters, [Bernard et al. \(2013\)](#) got an almost equal number of weather stations in the north and south clusters dividing France along the Loire valley line. HOSil for number of clusters three further separates, the cluster 1 in south-east and south-west fashion i.e., isolating the regions with the Alps and Pyrenees. For the numbers of clusters four, the HOSil has further isolated central France from the northern region. For the numbers of clusters five the upper northern region is further divided into west to north-west and from north-west to the north regions. As the number of clusters increases, the clusters are located consistently with the geographical regions. This finding is consistent with results of [Bernard et al. \(2013\)](#). For seven numbers of clusters Nice, Bastia, and Ajaccio were put together. This is an indication that the climate patterns of Nice, Bastia, and Ajaccio are more similar to each other as compared to other stations in this region. Geographically, Nice is closer to Corsica(Bastia and Ajaccio) as compared to Toulon from Corsica. The weather in Toulon is more similar to Marseille and the Mediterranean coastal line south of it e.g., Montpellier down till Perpignan and hence it is put together with these.

In terms of the number of clusters, the highest ASW value was obtained for $k=2$ (0.1390), indicating the strongest weather pattern, meaning that the climates of these two regions differs most significantly in the country. The second best ASW value was achieved for $k=3$ (0.1251). After that, the strongest climate pattern was observed for $k=7$ (0.1166). The fourth best ASW was obtained for $k=6$ (0.1135).

The best numbers of cluster suggested by the implementaton of the PAM algorithm by [Bernard et al. \(2013\)](#) was $k=2$ based on the optimal value of ASW. The first few best choices for the number of clusters proposed by this algorithm together with their

ASW values are: $k=2$ (ASW=0.1225), $k=5$ (0.1116), $k=4$ (0.1101) and $k=7$ (0.1058). For $k = 2$ the far regions from the north-west of France are classified together with the southern cluster because of the presence of the Armorican mountains in that region, which makes the climate appear more similar to the other mountainous regions in south of the country.

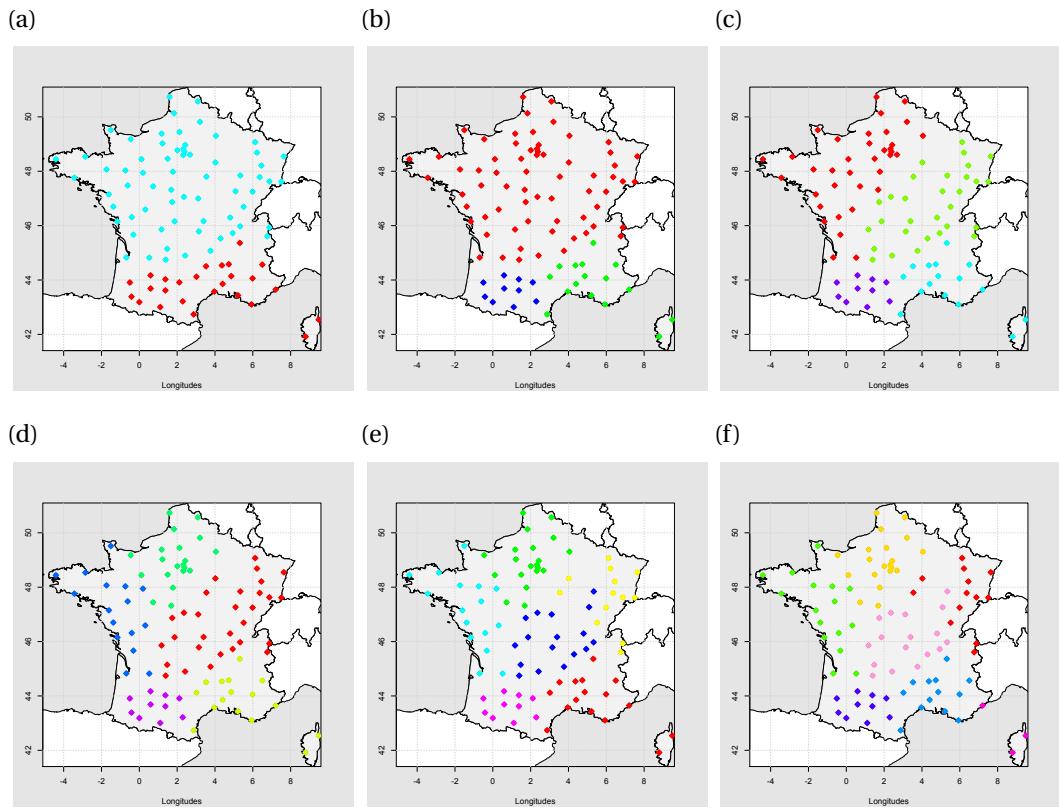


Figure 3.9 Clustering results from HOSil algorithm. Panels from (a) - (f) denote clustering against $k=2$ to $k=7$.

3.10 Closing remarks

HOSil suffers from a high computational cost which makes it unfit to be used for data values larger than 800. This method is good for only small data sets. More discussion and data application about HOSil algorithm will be presented at the end of next chapter. We have applied HOSil on single cell RNA sequencing data clustering problems and the results are reported in Section 4.17.7, together with the methods purposed in next chapter for the comparisons.

One way to improve HOSil is to make it computationally faster. We have proposed a fast approximation of HOSil to further improve its computational cost in last chapter. However, for now we didn't work further to improve computational cost of HOSil and focus to move towards other clustering domains that can be naturally faster for larger data sets. We decided not to compromise on the performance of HOSil with a caveat that it is only suitable to small datasets. Therefore, focus from now on is on the development of another algorithm based on non-hierarchical clustering methods since they are in general faster than hierarchical methods.

Chapter 4

The Optimum ASW Partitioning Clustering Method

4.1 Background and preliminary notations

In this chapter we will introduce a partitional algorithm for the optimization of ASW for clustering. The new method needs an initial clustering solution to begin like the k -means or PAM algorithm. This is because ASW is originally an index which is computed for a given clustering. We will call the initial clustering solution as the initialization of the method. Based on some clustering method, for instance, random initialization, k -means, PAM, agglomerative hierarchical clustering (AHC), or even HOSil itself as an initialization, an initial clustering solution can be found. Then these solutions can be improved further by maximizing the silhouette width for every point in the data.

Suppose that the aim is to cluster a data set \mathcal{X} of size n where $n \geq 2$ into $k \geq 2$ clusters, i.e., each object to cluster is a p dimensional vector as set in Definition 2.2.1. Let $d(x_i, x_h)$ be some distance measure such that $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$. The pairwise dissimilarities between n objects can be represented as a symmetric square matrix of size n . The main diagonal of such a matrix will be zero due to Definition 2.2.2 - (ii), and its lower and upper triangular matrix are the same due to Definition 2.2.2 - (iii). Therefore, we only need the lower triangular matrix whose entries are determined by the distance function. For all $(x_i, x_h) \in \mathcal{X}$ and $i \neq h \in \mathbb{N}_n$ a function $d(x_i, x_h)$, determines the entries of the following matrix as:

$$D = \begin{bmatrix} d(x_2, x_1) & & & \\ d(x_3, x_1) & d(x_3, x_2) & & \\ \vdots & \vdots & \ddots & \\ d(x_n, x_1) & d(x_n, x_2) & \dots & d(x_n, x_{n-1}) \end{bmatrix} \quad (4.1)$$

Any distance measure, for instance Minkowski distance as introduced in Definition 2.2.3 or mentioned in Section 2.3.1, or any other distance measure can be used.

Note that although we used \mathbb{R}^p for the simulation of the data \mathcal{X} in experiments, however the proposed algorithm in this and the previous chapter works with the data from other spaces i.e., we don't need to assume that data is from \mathbb{R}^p . The proposed algorithms also work with general distances, thus, specifying \mathcal{X} belongs to some space S characterised by distances $d : S \times S \rightarrow \mathbb{R}^+$, such that \mathcal{X} , the data is a subset of S is enough for the formalism.

There are two trivial clustering cases which are not of interest in this work. They are defined as: all the data points belong to one cluster only i.e., $k = 1$, and each data point forms its own cluster i.e., $k = n$. Let $\mathcal{P}(\mathcal{X})$ be the set of all non-trivial partitions on \mathcal{X} . Let $\mathcal{C}_k \in \mathcal{P}(\mathcal{X})$ such that $\mathcal{C}_k = \{C_1, C_2, \dots, C_k\}$ be a clustering with any size k characterised by any clustering method.

4.2 OASW clustering

The Optimum Average Silhouette Width (OASW) clustering of \mathcal{X} is defined by maximizing the function given in (2.12), over all $\mathcal{C}_k \in \mathcal{P}(\mathcal{X})$, where $\mathcal{P}(\mathcal{X})$ represents the set of all possible non-trivial clusterings \mathcal{C}_k on \mathcal{X} . We recall (2.12) as follows:

$$\bar{S}(\mathcal{C}_k, d) = \frac{1}{n} \sum_{i=1}^n S_i(\mathcal{C}_k, d),$$

where $S_i(\mathcal{C}_k, d)$ is defined in (2.11). The objective function can be defined in different ways all of which are useful, which we give now. Let $l(\mathcal{X}, k)$ be a brief notation for $(l(1), \dots, l(n))$, where $l(i) = r; r \in \{1, \dots, k\}, i \in \{1, \dots, n\}$ be the vector of labels for clustering \mathcal{C}_k . Since a clustering \mathcal{C}_k is determined by its label set i.e., an identification of cluster membership for each object in the data, replacing \mathcal{C}_k by $l(\mathcal{X}, k)$ in above equation will make the objective function more clearly understandable in terms of what exactly is needed to maximize the objective function. Therefore, replacing \mathcal{C}_k by the clustering label set gives the following representation:

$$\bar{S}_l(l(\mathcal{X}, k), d) = \frac{1}{n} \sum_{i=1}^n S_{l_i}(l(\mathcal{X}, k), d). \quad (4.2)$$

The OASW clustering objective function is defined as follows:

$$f(l(\mathcal{X}, k), d) = \arg \max_{l^*(\mathcal{X}, k) \in \mathcal{L}} \bar{S}_l(l^*(\mathcal{X}, k), d), \quad (4.3)$$

where \mathcal{L} represents a set of all possible label vectors $l^*(\mathcal{X}, k)$ for all possible non-trivial clusterings $\mathcal{C}_k \in \mathcal{P}(\mathcal{X})$. Back substitution in (4.3) will give us a full definition of the ob-

jective function through one equation. First substitute (2.11) in (4.2) keeping in mind that \mathcal{C}_k is now replaced by $l(\mathcal{X}, k)$ and then substituting this result in (4.3) gives the following expression,

$$f(l(\mathcal{X}, k), d) = \arg \max_{l^*(\mathcal{X}, k) \in \mathcal{L}} \frac{1}{n} \sum_{i=1}^n \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad (4.4)$$

where $l^*(\mathcal{X}, k)$ is required for $b(i)$ and $a(i)$ which are defined in (2.9) and (2.10) respectively.

We define the following implementation named as OSil of the **Optimum average Silhouette width clustering**.

OSil algorithm

Initialize

- (i) For all $(x_i, x_h) \in \mathcal{X}$, where $(i, h) \in \mathbb{N}_n$ and $i \neq h$, calculate $d(x_i, x_j)$.
- (ii) Calculate a clustering using any crisp clustering criterion and initialize the clustering label vector with k clusters as $l(\mathcal{X}, k) = (l(1), \dots, l(n))$.
- (iii) Calculate $f^{(0)} = f(l(\mathcal{X}, k), d)$.
- (iv) Set $q = 1$. Let $l^{(1)}(\mathcal{X}, k) = l(\mathcal{X}, k)$.

Swap

- (i) For all pairs (i, r) such that $l^{(q)}(i) \neq r$, for $i \in \mathbb{N}_n$ and $r \in \mathbb{N}_k$, assign $l(i) = r$ and denote the new label set as $l_{(i,r)}^*(\mathcal{X}, k) = (l^*(1), \dots, l^*(n))$.
- (ii) Compute $f_{(i,r)} = f(l_{(i,r)}^*(\mathcal{X}, k), d)$.
- (iii) $(h, s) = \arg \max_{(i,r)} f_{(i,r)}$, $f^{(q)} = f_{(h,s)}$, $l^{(q)}(\mathcal{X}, k) = l_{(h,s)}^*(\mathcal{X}, k)$.

Stop

If $f^{(q)} \leq f^{(q-1)}$. Else $q = q + 1$. Repeat *Swapping*: Step (i)-(iii).

Return

$f^{(q)}$ and $l^{(q)}(\mathcal{X}, k)$.

The objective of OASW clustering is to find a clustering for which $\tilde{S}(l(\mathcal{X}, k), d)$ is maximum from all the possible clusterings \mathcal{C}_k of \mathcal{X} . The set \mathcal{L} is determined by all the possible combinations of an object with the membership of a cluster for an initial clustering label vector $l(\mathcal{X}, k)$. This is achieved by changing the membership of each observation i in cluster C_r to every other cluster of which it was not previously member of, i.e., by performing all possible swaps between observation and cluster memberships and calculating objective function in (4.4). These are in total $n \times (k - 1)$ swaps per iteration. If a swapping increases the objective function we get a new label set to restart the swapping for all n objects again and this continues till convergence. Thus

the OASW clustering objective function is in search of a non-trivial clustering \mathcal{C}_k such that no other contender can further optimize this objective function.

OSil is a combinatorial algorithm that directly assigns each observation to a cluster ([Friedman et al. \(2001\)](#)). OSil is one way of trying to solve the problem in [\(4.4\)](#). The algorithm does not exactly solve but only tries to solve the optimization problem given in [\(4.4\)](#) and will only get a local optimum. The algorithm continues its search iteratively that will maximize the value of function in [\(4.4\)](#). Since the algorithm takes one step at a time therefore, it only guarantees local optimum.

4.3 Implementation of the OASW method

The algorithm first finds an initial clustering solution to start. This is done in the “initialize” step. The initial label set obtained from an initial clustering is denoted by $l(\mathcal{X}, k)$ or $l^{(1)}(\mathcal{X}, k)$, and $f^{(0)}$ represents the value of the objective function for these initial labels. Then we shift every object i , $i = 1, \dots, n$ into a cluster C_r to other than its present cluster. This will change the values of SW of the observations and the ASW. This is numbered as $q = 1$. Note that for each observation i there will be $(k - 1)$ possible clusters to consider for swapping. Thus the total number of possible swaps at each iteration (q value) will be $n \times (k - 1)$. In $f_{(h,s)}$ the index h holds the observation number i and s holds the cluster number r against which the maximum of objective function was achieved. This best swapping indicator (h, s) then gives us the label set to start the next $(q + 1)^{th}$ iteration, and $f_{(h,s)}$ gives the best value of the objective function from this iteration.

Note that it is possible that while swapping observations between clusters we find more than one such case (i.e., combination of (i, r)) that improves that value of the objective function from where we started this iteration. For instance at $q = 1$ we have $f^{(0)}$ as a starting value of the objective function, and we could get more than one such pair (i, r) such that $f_{(i,r)} > f^{(0)}$. The above defined algorithm uses argmax meaning that at each iteration it chooses that swap from all possible swaps between observation i and cluster r to define the clustering label set for the next iteration that gives the maximum impact to the value of the objective function. Thus, out of $n \times (k - 1)$ possibilities we only swap one observation to one other cluster at each iteration. Then we update the objective function with the value obtained for the swap (this is $f_{(h,s)}$). Go to the next iteration, consider swapping every observation to every other cluster again and finally update the objective function and so on. Stop the process if the objective function value in $f^{(q)}$ at the current iteration is not improved as compared to its value at the previous iteration. This means that there is no such swap that can increase the value of the objective function further. The algorithm searches for a better value of the ASW by iteratively incrementing it, and will always converge. The increment is performed by moving one element of the solution and it keeps going until no further improve-

ment can be made. The main algorithm is written in “C++” language due to its known computational power. Since most of the clustering algorithms are available in R due to the popularity of the language and the flexibility of use, we have also integrated the algorithm in R through the package ‘Rcpp’ (version: 0.12.10, [Eddelbuettel et al. \(2011\)](#)). Our implementation is similar to [Van der Laan et al. \(2003\)](#)’s C implementation.

Note that it is possible that the algorithm terminates at the first iteration and it does not find any further improvement in the value of objective function that is received from the initial clustering. The number of iterations taken by the OASW depends upon the initialization method used. For instance, the algorithm always takes more iterations to optimize the ASW when average linkage was used as an initialization as compared to k -means.

4.4 Simulation setup

To find out the best initialization method(s) for OSil, and for the comparison of it’s performance with the existing methods, extensive simulations have been conducted. The first concern here was to find out what’s the best ways of initializing the algorithm to maximize ASW for the majority of the data conditions. Careful initialization of the algorithms can improve the performance of the algorithms greatly. For instance [Arthur and Vassilvitskii \(2007\)](#) demonstrate this for the k -means algorithm and shows how just changing initialization not only speeds up the computational time for the algorithm but also improves the accuracy of the results.

The other motivation for setting up these simulations was to illustrate the characteristics, and what kinds of clusters OASW clustering can capture. The simulations in this regard were conducted in two fundamentally different ways. First two simulations (namely, Simulation I and simulation II) were run for various data generating processes (DGPs), and a third simulation is run using an experimental design approach (namely, Simulation III) to explore the characteristics that can’t be learn through the DGPs.

Simulation I (Section 4.5) is for the known true number of clusters case, and Simulation II (Section 4.6) is for the estimation of number of clusters case. This case distinction is because, first we want to learn, the performance of the method for the known true number of clusters, i.e., OSil can produced the true clustering defined by DGPs or not. Through this we can learn two things, firstly, what kind of clusters OSil can produce, and secondly, to get an idea how different initialization methods are performing for different DGPs. The performance was then explored for the estimation of number of clusters.

Simulation III (Section 4.7) was run using the experimental design approach where factors were defined and varied systematically. This setup was designed to vary various factors of interest, like how the OSil clustering will be affected if the difference between the means of clusters and various co-variance structures vary. The experimental design

setup was run for the overlapping, not well-separated, very close and nested clustering structures. These types of clustering structures are hard for many methods to identify. This setup was run for the fixed/known k case and the focus was on the comparison of the ASW values obtained for various methods under different conditions.

We now define the DGPs used for Simulation I & II in the following subsection.

4.4.1 Definitions of data generating processes

To explore the characteristics of the proposed algorithm and what kind of clusters it is good in finding, much attention has been given to data sets in two and three dimensions so that the clustering results can be visualized. Several DGPs were defined based on the characteristics listed in Section 3.3. The DGPs were made more challenging in this chapter as compared to previous by increasing the observation spread within clusters, or by decreasing the difference between the mean of clusters. The DGPs are defined now as under.

Model 1:

2 clusters in 2 dimensions: 50 observations each were generated from two independent Gaussian random variables, to form two spherical clusters in two dimensions, of unequal variations. One cluster has mean $(0, 5)$ with covariance matrix as $0.1I_2$ and the other cluster has mean $(2, 5)$, where t represents the transpose, with covariance matrix as $0.7I_2$. The result is one bigger spherical cluster with wider spread lying next to a compact spherical cluster.

Model 2:

3 clusters in 2 dimensions: The observations in each of the three clusters were generated from independent Gaussian random variables centred at $(-2, 0)$ and covariance matrix $0.1I_2$ for cluster 1, mean $(0, 0)$ and covariance matrix $0.7I_2$ for cluster 2, and mean $(2, 0)$ and covariance matrix $0.1I_2$ for cluster 3. The cluster contains 50 observations each. The clusters are of such nature that the cluster with greater observational variation is located between the two clusters having less variations among observations.

Model 3:

4 clusters in 2 dimensions: Cluster one was generated from two independently distributed non-central t variables with parameters $t_7(10)$ and $t_7(30)$. Cluster two was generated from $\mathbb{U}(10, 15)$ along both dimensions independently. Cluster 3 was generated from independent Gaussian distribution having mean $(2, 2)$ with covariance

matrix $\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$. Cluster four was also generated from independent Gaussian distributions with mean $(20, 80)$ and covariance matrix $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$ respectively. Each cluster contains 50 observations.

Model 4:

5 clusters in 2 dimensions: the clusters are parametrized from \mathbb{F} , Chi-squared, Gaussian, skewed Gaussian and t distributions respectively as: $\mathbb{F}_{(2,6)}(4)$ along first dimension and $\mathbb{F}_{(5,5)}(4)$ along second dimension, $\chi^2_7(35)$ and $\chi^2_{10}(60)$, $N(100, 2)$ and $N(0, 2)$, $SN(20, 2, 2, 4)$ and $SN(200, 2, 3, 6)$, $t_{40}(100)$ and $t_{35}(150)$ respectively. This cluster ordering is also reflected in the data plot generated from this DGP shown in Figure 4.4b as well i.e., a label 1 represents an \mathbb{F} distributed clusters and so forth. The clusters contains 50 observations each and were generated independently along both dimensions.

Model 5:

6 clusters in 2 dimensions: the clusters 1 and 2 are generated from Uniform and exponential distributions as $\mathbb{U}(-6, -2)$, $Exp(10)$ in both dimensions. The cluster 3 is $NBeta(2, 3, 220)$ along one dimension and $NBeta(2, 3, 120)$ across the other dimension. Cluster 4 is from $SN(5, 0.6, 4, 5)$ and $SN(0, 0.6, 4, 5)$. Cluster 5 is $W(10, 4)$ across both dimensions. Cluster 6 is $Gam(15, 2)$ and $Gam(15, 0)$ along first and second dimension respectively. This ordering of clusters are reflected in the data plot generated from this DGP shown in the Figure 4.5a as well. The clusters contains 50 observations each and were generated independently along both dimensions.

Model 6:

5 correlated dimensions within 5 clusters are generated from multi-variate Gaussian distributions each containing 50 observations. The clusters are formed as follows:

Cluster 1 is centred at $(0, 0, 0, 0, 0)$ with $\Sigma = \begin{bmatrix} 9 & & & & & \\ 1 & 17 & & & & \\ 1 & -1.4 & 12 & & & \\ 0.4 & 0.6 & 0.5 & 2 & & \\ -1.2 & -1.6 & -1.4 & -0.6 & 16 & \end{bmatrix}$.

$$\begin{aligned}
\text{Cluster 2 is centred at } (40, 80, 15, 30, 22) \text{ with } \Sigma = & \begin{bmatrix} 25 \\ 0.2 & 9 \\ 0.2 & -0.2 & 16 \\ -0.2 & -0.2 & 0.2 & 1 \\ -0.2 & -0.2 & -0.2 & -0.2 & 49 \end{bmatrix}. \\
\text{Cluster 3 is centred at } (15, 40, 40, 55, 80) \text{ with } \Sigma = & \begin{bmatrix} 25 \\ 0.3 & 9 \\ 0.3 & -0.3 & 16 \\ -0.3 & 0.3 & 0.3 & 1 \\ -0.3 & -0.3 & -0.3 & -0.3 & 49 \end{bmatrix}. \\
\text{Cluster 4 is centred at } (70, 80, 70, 70, 70) \text{ with } \Sigma = & \begin{bmatrix} 5 \\ 0.1 & 0.9 \\ 0.1 & -0.2 & 1.6 \\ -0.7 & 0.2 & 0.2 & 1 \\ -0.2 & -0.9 & -0.2 & -0.2 & 4.9 \end{bmatrix}. \\
\text{Cluster 5 is centred at } (100, 100, 100, 100, 100) \text{ with } \Sigma = & \begin{bmatrix} 2 \\ 0.2 & 9 \\ 0.2 & -0.1 & 3 \\ -0.3 & 0.2 & 0.1 & 1 \\ -0.1 & -0.1 & -0.2 & -0.9 & 4 \end{bmatrix}.
\end{aligned}$$

Model 7:

7 clusters in 10 dimensions having 50 observations each. All the clusters are from the Gaussian distributions. The clusters are present in the first two dimensions only. Cluster 1 has mean (0, 5) with covariance matrix $\begin{bmatrix} 0.5 & 0 \\ 0 & 0.2 \end{bmatrix}$. Cluster 2 has mean (-0.5, 3.5) and covariance matrix $\begin{bmatrix} 0.2 & 0 \\ 0 & 0.1 \end{bmatrix}$. Cluster 3 has mean (0, 3.5) with covariance matrix $\begin{bmatrix} 0.4 & 0 \\ 0 & 0.3 \end{bmatrix}$. Cluster 4 has mean (0.5, 3.5) and covariance matrix $\begin{bmatrix} 0.2 & 0 \\ 0 & 0.1 \end{bmatrix}$. Cluster 5 has mean (-0.5, 6.5) and covariance matrix $\begin{bmatrix} 0.2 & 0 \\ 0 & 0.1 \end{bmatrix}$. Cluster 6 has mean (0, 6.5) and covariance matrix $\begin{bmatrix} 0.3 & 0 \\ 0 & 0.2 \end{bmatrix}$. Cluster 7 has mean (0.5, 6.5) and covariance matrix $\begin{bmatrix} 0.3 & 0 \\ 0 & 0.2 \end{bmatrix}$. Further dimensions 3 to 6 of cluster 1 were generated by subtracting the values 3, 6, 9, 12 from its second dimension. Dimensions 7 to 10 of cluster 4 were

generated by adding the values 3, 6, 9, 12 from its second dimension. Dimensions 3 to 10, of clusters 2 to 4 were generated by adding the values 3, 6, 9, 12, 15, 18, 21, 24 to the second dimensions of these clusters. For dimensions 3 to 10, of cluster 5 to 7, the values 3, 6, 9, 12, 15, 18, 21, 24 are subtracted from the second dimensions of the respected clusters.

Model 8:

10 cluster in 500 dimensions. This model is same as the Model 16 of Chapter 3. The motivation for including this dataset is specifically the estimation of k case. Since the clusters are of unequal sizes and variations the intuition is most of the existing clustering methods to estimate number of clusters will fail in estimating the correct number of clusters majority of times.

Model 9:

3 clusters in 1000 dimensions. Each cluster contains 40 realizations from standard Gaussian distributions with each of first 100 coordinates centred at -3, 0, and 3 respectively. The remaining coordinates of all clusters have mean 0. All the clusters have I_{1000} covariance matrices.

Model 10:

7 clusters in 60 dimensions with 500 observations: This is a data structure designed by [Van der Laan et al. \(2003\)](#) to simulate gene expression profiles like structure for three distinct types of cancer patients' populations. Suppose that in reality there are 3 distinct groups 20 patients each corresponding to a cancer type. Three multivariate normal distributions were used to generate 20 samples each having different mean vectors. For the first multivariate distribution (first cancer type) the first 25 dimensions(genes) are centred at $\log_{10}(3)$, dimensions 26-50 are centred at $(-\log_{10}(3))$ the remaining 450 dimensions are centred at 0. For the second multivariate distribution (second cancer type) the first 50 dimensions(genes) are centred at 0, the next 25 dimensions (51-75) are centred at $\log_{10}(3)$, dimensions 76-100 are centred at $(-\log_{10}(3))$ and the remaining 400 dimensions are also centred at 0. For the third multivariate distribution (third cancer type) the first 100 dimensions(genes) are centred at 0, dimensions 101-125 are centred at $\log_{10}(3)$, dimensions 126-150 are centred at $(-\log_{10}(3))$ and dimensions 151-500 are also centred at 0. The three multivariate distributions has diagonal covariance matrix with diagonal elements as $(\log(1.6))^2$. Note that the described data has 20 samples each of 3 types of cancer patients each containing 500 genes. The purpose here is to cluster genes not patients. Therefore, the transpose of

the data is required to transfer it to the standard format and the number of clusters to seek are 7 in 60 dimensions of 500 observations.

4.5 Simulation I: Fix k case

Each data generating process (DGPs) has certain kinds of clustering problem(s) to solve. We expect from the algorithms to retrieve the clustering as defined by the DGPs. Through comparisons we will learn which clustering algorithm is good in reproducing the certain kinds of clusters. To find the best initialization method for the algorithm several existing clustering methods were used as an initialization for *OSil*, namely, *k-means*, *PAM*, *model-based*, *spectral*, agglomerative hierarchical linkage methods using *average*, *complete*, *single*, *McQuitty* and *Ward's* method were used which are reviewed in Section 2.4. The simulation is done in the R language. For all the clustering methods, the known number of clusters from the corresponding DGPs were used. For all the clustering methods we have used their R functions as described in Section 3.5 with the default parametric choices except otherwise stated. For *k-means* we have set random centres to be chosen 100 times (this is argument ‘*nstart*’ of the function ‘*kmeans()*’). This is because the performance of *k-means* improves if one allows the algorithm to optimize the objective function by taking several set of cluster centres. For the ASW calculations of the clustering solutions obtained from the clustering methods other than *OSil* we used the ‘*silhouette()*’ function in the R package ‘*cluster*’. For the ARI calculation the function ‘*adjustedRandIndex()*’ from the package ‘*mclust*’ was used.

Let the number of data sets generated for each DGP is denoted by B . For each DGP, $B = 25$ data sets were generated. Clusterings were performed using the 9 clustering methods just mentioned. We then note the ASW and ARI values for these clusterings together with their standard errors (SE). We then pass these clustering solutions to the *OSil* algorithm as initializations, i.e., with each initialization method *OSil* was run separately. This will result in 9 different *OSil* clusterings. We then recorded the ASW and ARI values together with their SE for these *OSil* clusterings as well.

For comparison, the aggregated results, i.e., averages of 25 runs of the ASW value for each initialization method considered are reported together with their average SE. The average number of iterations (abbreviated as ‘*iter*’ in tables of Appendix C) and the average runtime in seconds taken by *OSil* are also reported. The time reported in the tables is in seconds and includes initialization time as well. The box-plots for the ASW values are also plotted. The empirical distributions for ASW obtained from *OSil* are plotted as histograms against each initialization method. For each generated data set we have also computed the PAMSIL clustering solution for the comparison. Note that ‘*init*’ represents the average ARI values against the clustering obtained from the existing clustering methods before passing these clusterings as initializations to *OSil*. Since the clustering methods used to initial *OSil* algorithm are well developed

clustering methods as their own, therefore, we used the clustering results (referred to as the initial clustering in the following discussions) obtained from these to compare OSil clustering.

Note that PAMSIL is not same as OSil initialized with PAM. The difference between the two is that PAMSIL each time makes a non-medoid object a medoid. It reassigns the cluster memberships based on the new set of medoids, and then calculates the ASW for this clustering. OSil doesn't work based on medoids. It just uses the cluster labelling vector. OSil changes the cluster membership of each object to each cluster i.e., it tries to optimize the silhouette width of each individual object in the data. OSil initialized with PAM optimizes the value of the ASW on top of the PAM clustering solution, whereas PAMSIL optimises ASW on top of PAM-like medoids.

4.5.1 Results discussion

For the results discussion below the first evaluation principle is the value of ASW achieved. The higher the value of ASW obtained from OSil the better the results is in terms of optimization of ASW. We have however, also compared the ARI values obtained in order to learn how good the best OASW clustering will perform in delivering the know true clustering structures defined by DGPs.

(Model 1) Figure 4.1a represents a data plot generated from Model 1. Table 4.1 represents above mentioned statistics for Model 1. The box and histogram plots of ASW against each initialization method and PAMSIL are plotted in Figure 4.1b and C.1, respectively. The best values for ASW and their ARI together with the best runtime are made bold in the corresponding columns. PAM clustering gave the highest ASW value, however the highest ARI was obtained for model-based clustering. The highest ASW was obtained for a number of initialization methods, namely k -means, PAM, Ward's, spectral clustering, model-based clustering and PAMSIL. The highest ARI value (0.8603) was achieved by OSil clustering with Ward's, spectral clustering and model-based clustering initializations. An important point to note here is that although k -means and PAM have given maximum ASW they have not given the best ARI. Another interesting finding is, OSil reached at the best ASW value for many ASW values obtained from these initialization methods. See Table 4.1 where ASW obtained for k -means, pam, Ward's, model-based clustering, spectral clustering and PAMSIL are different but ASW values obtained from these methods are same. This is true for many other DGPs as well. Another important thing to note here is that the best ARI value achieved for OSil clustering is actually lesser than the best ARI achieved by the existing clustering method. Ward's method, spectral clustering and model-based clustering initializations performed best for Model 1 based on the best ARI and ASW values obtained by OSil. The minimum time to optimize ASW was obtained by the PAM initialization.

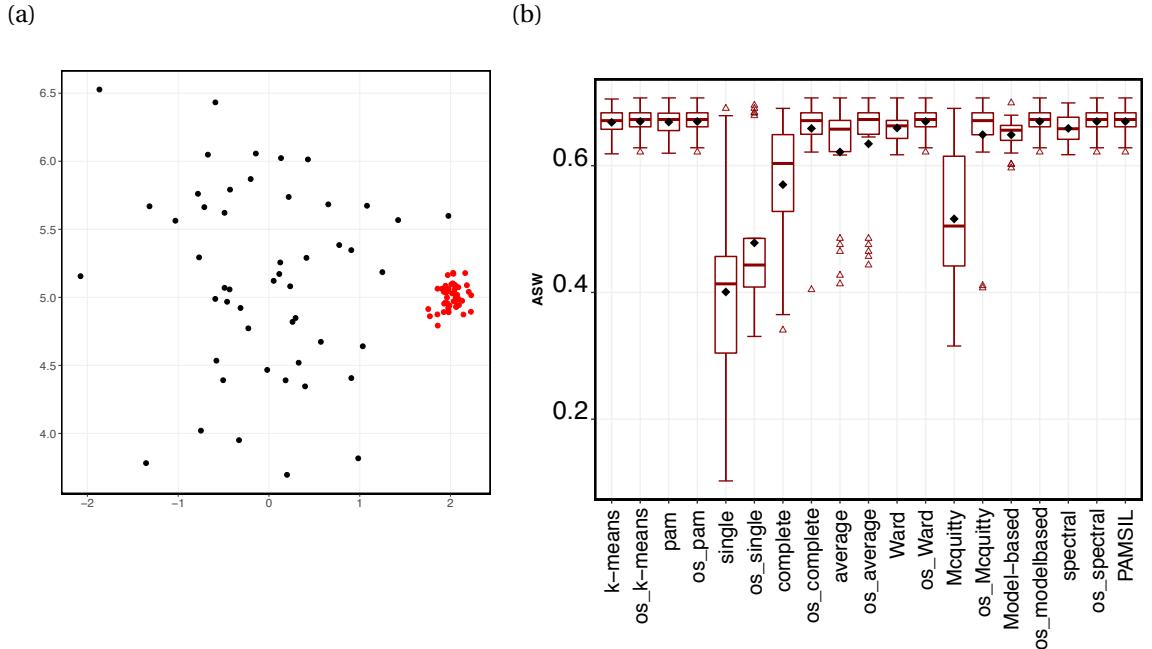


Figure 4.1 (a) represents a synthetic data plot generated from Model 1. (b) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles.

(Model 2) Figure 4.2a represents a data plot generated from Model 2. Table 4.2 shows the statistics of interest for Model 2. The box and histogram plots of ASW against each of initializations and PAMSIL are plotted in Figure 4.2b and C.2, respectively. The maximum ASW was achieved by k -means clustering, whereas the maximum ARI was achieved by model-based clustering. The best ASW was obtained from 4 initialization methods (see table). The best ARI for OSil was attained only for model-based initialization method. The highest ARI value achieved for OSil was smaller than the highest value of ARI obtained against the maximum ASW clustering. The minimum time for OSil is observed with k -means initialization.

(Model 3) Figure 4.3a represents a data plot generated from Model 3. Table 4.3 represents the statistics of interest for Model 3. The box and histogram plots of ASW against each initialization method and PAMSIL are plotted in Figure 4.3b and C.3 respectively. Among all clustering methods, the PAM clustering gave the best ASW, but model-based clustering gave the best ARI value. However, note that the model-based clustering did not give the highest ASW value. In comparison, the best ASW was obtained by OSil initialized with PAM, model-based clustering and PAMSIL method. However the best ARI

Table 4.1 Results for Model 1 against fixed k .

Methods	ASW (init)	SE	ASW (OSil)	SE	iter	runtime	ARI (init)	ARI (OSil)
k-means	0.6684	0.0044	0.6697	0.0043	2	0.0189	0.8197	0.8573
PAM	0.6689	0.0044	0.6697	0.0043	2	0.0140	0.8351	0.8573
single	0.4008	0.0288	0.4782	0.0225	6	0.0304	0.1172	0.1742
complete	0.5701	0.0208	0.6588	0.0114	14	0.0616	0.4831	0.8239
average	0.6217	0.0178	0.6345	0.0176	3	0.0188	0.6652	0.6836
Ward's	0.6596	0.0047	0.6697	0.0043	3	0.0265	0.9387	0.8603
McQuitty	0.5161	0.0229	0.6489	0.0151	18	0.0800	0.3569	0.7823
model-based	0.6488	0.005	0.6697	0.0043	4	0.0856	0.9920	0.8603
spectral	0.6589	0.0046	0.6697	0.0043	4	0.0957	0.9575	0.8603
PAMSIL	-	-	0.6697	0.0043	2	0.0232	-	0.8448

The second column represents the average ASW obtained against the existing clustering methods. The third column represents the average standard error (SE) for ASW. The fourth column represents the ASW when the solutions from the existing clustering methods are passed to OSil. The fifth column is average SE for ASW. The sixth column is the average number of iterations taken by OSil for convergence. The seventh column is the average time taken by OSil including initialization time. The last two columns represents the ARI for the clustering found by the existing methods and OSil.

Table 4.2 Results for Model 2 against fixed k .

Methods	ASW (init)	SE	ASW (OSil)	SE	iter	runtime	ARI (init)	ARI (OSil)
k-means	0.7114	0.0030	0.7118	0.0030	2	0.0647	0.8463	0.8556
PAM	0.7103	0.0031	0.7118	0.0030	2	0.0808	0.8491	0.8556
single	0.2976	0.0677	0.4161	0.0515	7	0.2255	0.3844	0.3535
complete	0.6128	0.0102	0.6859	0.0113	17	0.5026	0.6039	0.7631
average	0.6851	0.0105	0.6968	0.0084	5	0.1521	0.8161	0.7985
Ward's	0.6975	0.0044	0.7118	0.0030	5	0.1541	0.9140	0.8570
McQuitty	0.5889	0.0123	0.6726	0.0119	18	0.5172	0.5912	0.7313
model-based	0.6780	0.0040	0.7118	0.0030	8	0.2841	0.9880	0.8577
spectral	0.6164	0.0451	0.6536	0.0335	6	0.3469	0.8704	0.7993
PAMSIL	-	-	0.7117	0.0030	3	0.0992	-	0.7967

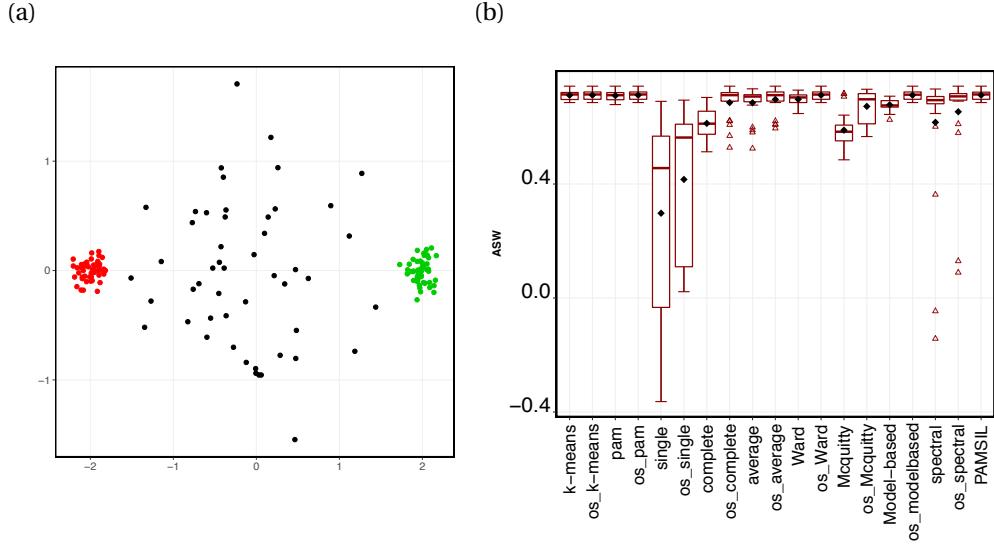


Figure 4.2 (a) represents a synthetic data plot generated from Model 2. (b) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles.

for ASW was achieved by PAMSIL only. The maximum ARI obtained from OSil clustering is smaller than the ARI obtained from the maximum ASW clustering. The PAM initialization took the minimum time for OSil.

(Model 4) Figure 4.4a represents a data plot generated from Model 4. Table 4.4 represents the statistics of interest for Model 4. The box and histogram plots of ASW against each initialization method and PAMSIL are plotted in Figure 4.4b and C.4 respectively. The best ASW value was achieved by k -means but the best ARI was achieved by Ward's method. The best ASW was achieved by average linkage initialization but the best ARI was by PAMSIL. The ASW clustering has increased the ARI as compared to the clustering obtained with the maximum ASW value. The k -means initialization takes the minimum time.

(Model 5) Figure 4.5a represents a data plot generated from Model 5. Table 4.5 represents the required statistics for Model 5. The box and histogram plots of ASW against each initialization method and PAMSIL are plotted in Figure 4.5b and C.5 respectively. PAM gave the best ASW value and the best ARI value as well. The best ASW was achieved for PAM initialization and PAMSIL clustering but the maximum ARI was obtained by PAMSIL. However, OSil with PAM initialization performed very close to PAMSIL in

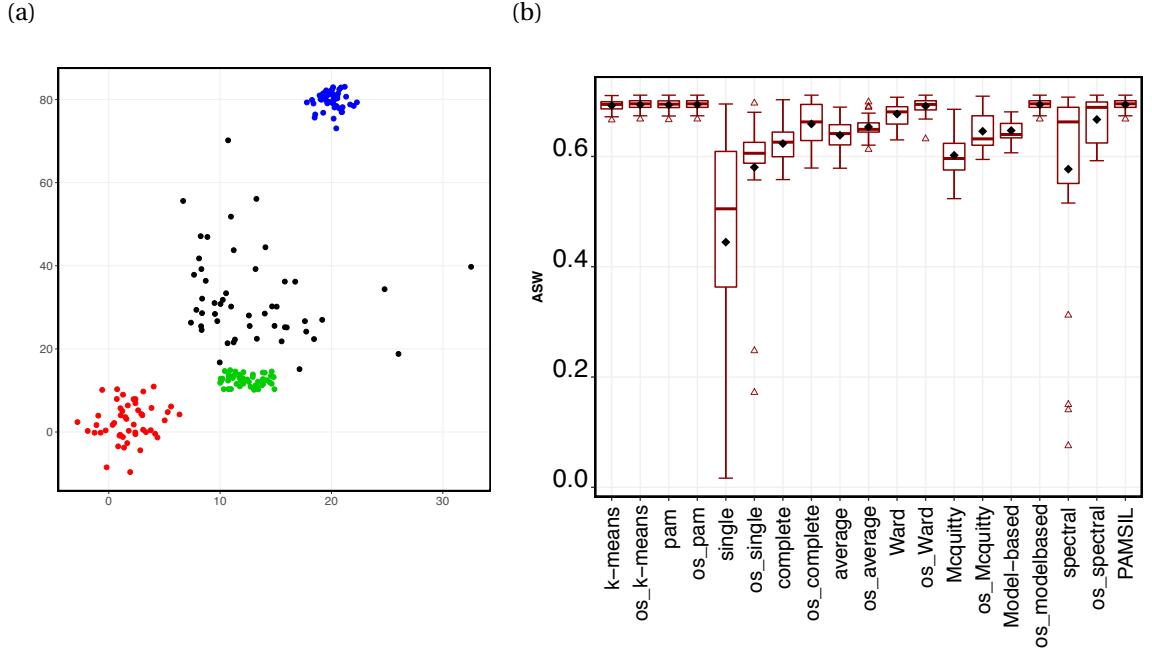


Figure 4.3 (a) represents a synthetic data plot generated from Model 3. (b) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles.

Table 4.3 Results for Model 3 against fixed k .

Methods	ASW (init)	SE	ASW (OSil)	SE	iter	runtime	ARI (init)	ARI (OSil)
k-means	0.6923	0.0023	0.6939	0.0023	3	0.3340	0.8711	0.8881
PAM	0.6936	0.0023	0.6940	0.0023	2	0.1924	0.8857	0.8871
single	0.4446	0.0407	0.5804	0.0234	11	1.0654	0.4219	0.3999
complete	0.6234	0.0075	0.6590	0.0078	13	1.2271	0.7054	0.7643
average	0.6385	0.0057	0.6535	0.0041	6	0.5426	0.6681	0.6661
Ward's	0.6770	0.0040	0.6914	0.0034	7	0.6937	0.9125	0.8866
McQuitty	0.6021	0.008	0.6457	0.0071	16	1.4817	0.6704	0.7122
model-based	0.6470	0.0038	0.6940	0.0023	10	0.9892	0.9920	0.8910
spectral	0.5770	0.0384	0.6670	0.0081	15	1.7418	0.8321	0.7941
PAMSIL	-	-	0.6940	0.0023	3	0.2283	-	0.9352

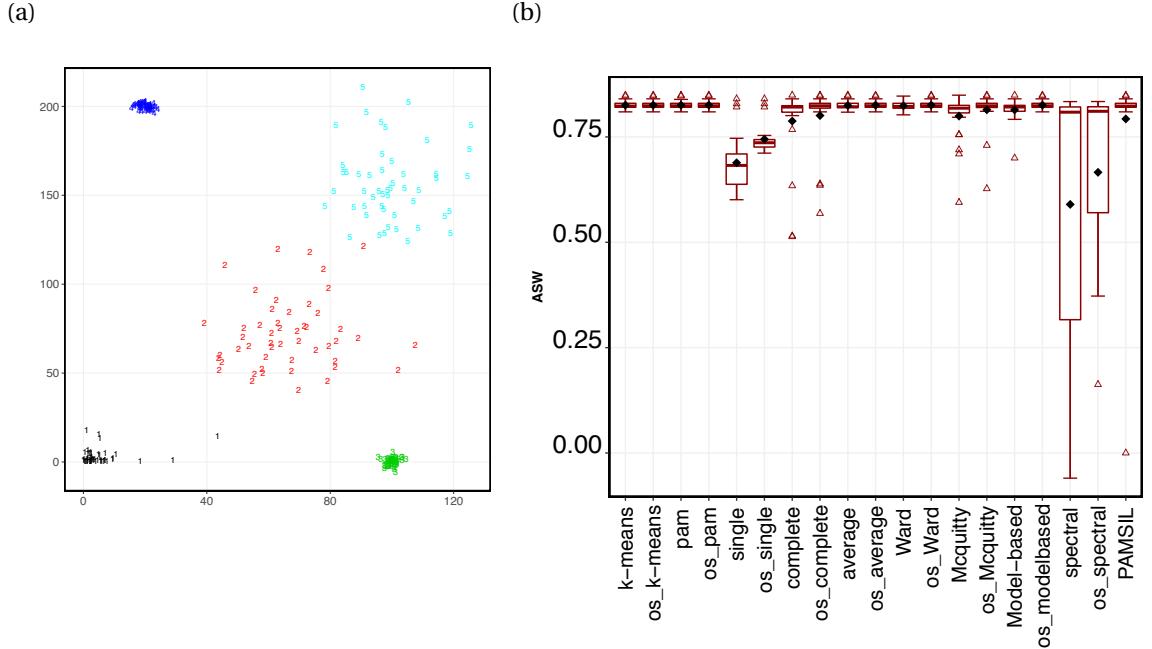


Figure 4.4 (a) represents a synthetic data plot generated from Model 4. (b) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles.

Table 4.4 Results for Model 4 against fixed k .

Methods	ASW (init)	SE	ASW (OSil)	SE	iter	runtime	ARI (init)	ARI (OSil)
k-means	0.8254	0.0021	0.8255	0.0021	2	0.4671	0.9845	0.9845
PAM	0.8254	0.0020	0.8255	0.0020	2	0.6651	0.9837	0.9853
single	0.6887	0.0164	0.7445	0.0064	5	1.2910	0.8029	0.7990
complete	0.7876	0.0083	0.8008	0.0047	15	4.0142	0.9374	0.9568
average	0.8247	0.0074	0.8255	0.0028	5	1.3531	0.9830	0.9845
Ward's	0.8239	0.0023	0.8255	0.0021	4	1.0016	0.9846	0.9845
McQuitty	0.7996	0.0134	0.8145	0.0056	17	4.4438	0.9444	0.9686
model-based	0.8143	0.0042	0.8255	0.0021	7	1.8341	0.9744	0.9853
spectral	0.5899	0.0388	0.6660	0.0079	17	5.0266	0.8457	0.8478
PAMSIL	-	-	0.8255	0.0020	3	0.7344	-	0.9853

terms of the ARI for this model. These two best ARI values for ASW clusterings are higher than the ARI obtained from the maximum ASW clustering. The minimum time taken was by k -means initialization.

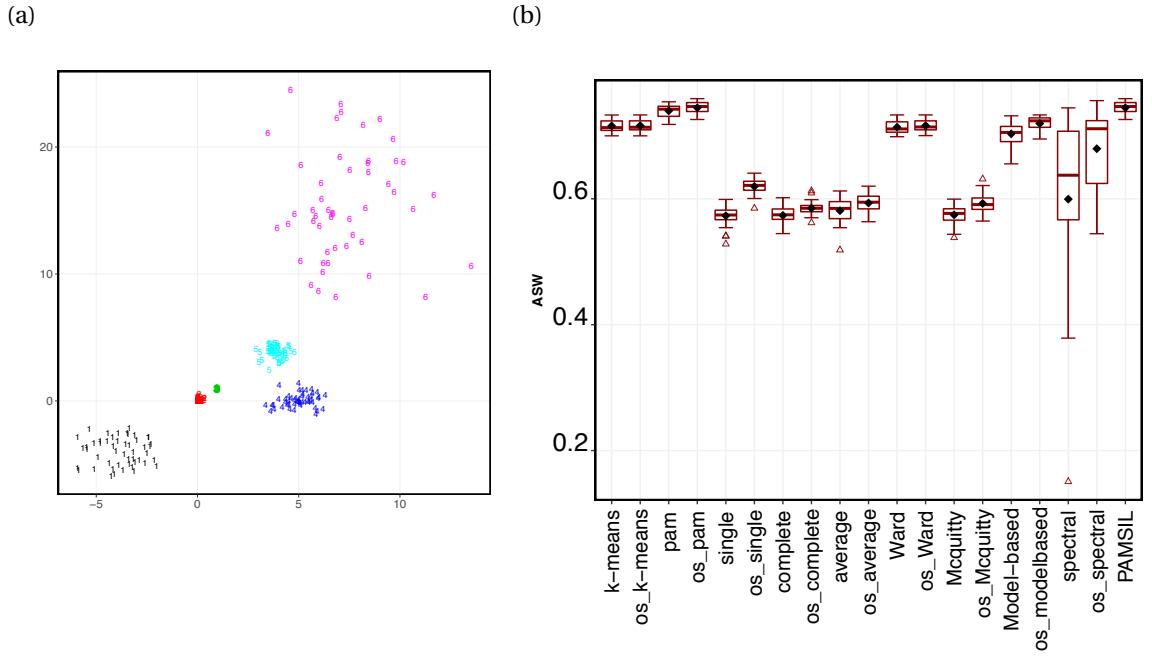


Figure 4.5 (a) represents a synthetic data plot generated from Model 5. (b) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles.

(Model 6) Figure 4.6a represents a data plot generated from Model 6. Table 4.6 represents the statistics of interest for Model 6. The box and histogram plots of ASW against each initialization method and PAMSIL are plotted in Figure 4.6b and C.6, respectively. This data model is particularly tough due to complex covariance matrices designs for clusters across five dimensions. The covariance matrices for each of the five clusters in Model 6 vary. The correlation among the dimensions are positive as well as negative. The clusters are also overlapping in some dimensions. McQuitty similarity gave the best ASW value and single linkage gave the best ASW. Ward's method gave the maximum ARI value among the existing clustering methods as well as for OSil. The ARI from the OSil clustering is smaller than the ARI for maximum ASW clustering. The minimum time was taken by the k -means initialization for OSil.

Table 4.5 Results for Model 5 against fixed k .

Methods	ASW (init)	SE	ASW (OSil)	SE	iter	runtime	ARI (init)	ARI (OSil)
k-means	0.7159	0.0020	0.7163	0.0020	2	0.9600	0.7716	0.7733
PAM	0.7398	0.0021	0.7448	0.0019	3	1.8634	0.9806	0.9984
single	0.5731	0.0036	0.6197	0.0025	5	2.5304	0.5218	0.5170
complete	0.5737	0.0029	0.5851	0.0024	7	4.1459	0.2845	0.2856
average	0.5812	0.0041	0.5934	0.003	6	3.0568	0.3012	0.3006
Ward's	0.7138	0.0021	0.7163	0.0020	3	1.7466	0.7750	0.7743
McQuitty	0.5745	0.0033	0.5927	0.0032	8	4.4237	0.2810	0.2830
model-based	0.7033	0.0036	0.7197	0.0022	8	4.7016	0.7762	0.7822
spectral	0.5995	0.0288	0.6796	0.0122	15	9.6341	0.7535	0.7841
PAMSIL	-	-	0.7448	0.0019	3	1.3833	-	1

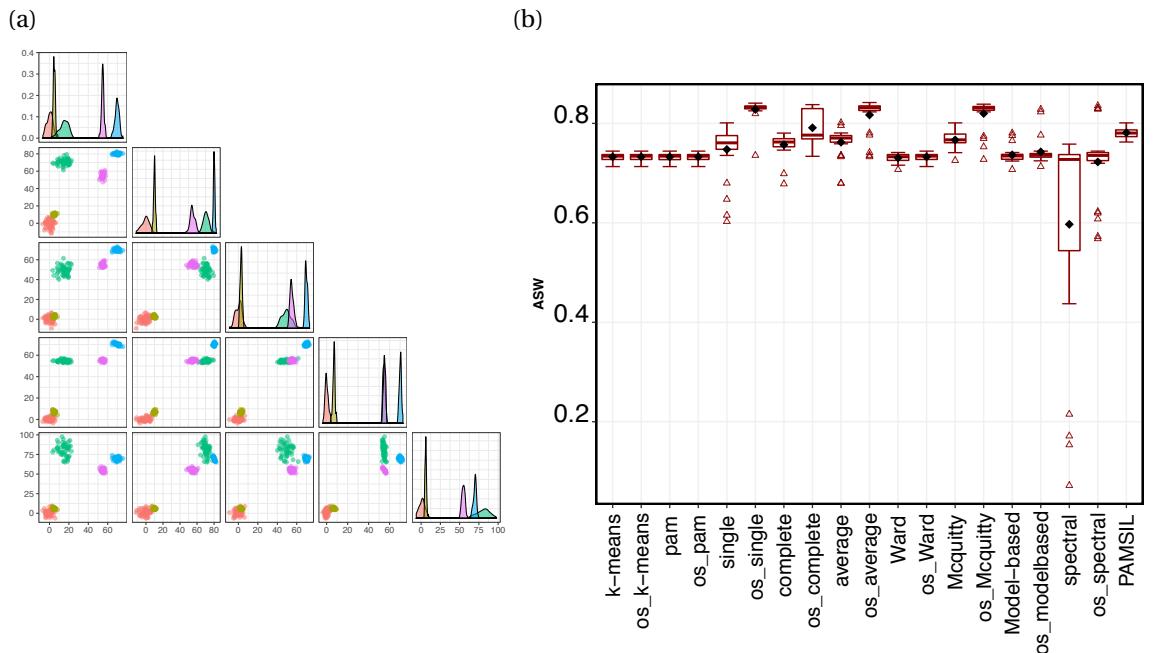


Figure 4.6 (a) represents a synthetic data plot generated from Model 6. (b) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles.

Table 4.6 Results for Model 6 against fixed k .

Methods	ASW (init)	SE	ASW (OSil)	SE	iter	runtime	ARI (init)	ARI (OSil)
k-means	0.7327	0.0015	0.7327	0.0015	1	0.3739	0.9795	0.9822
PAM	0.7325	0.0015	0.7327	0.0015	1	0.4296	0.9784	0.9822
single	0.7476	0.0107	0.8282	0.0040	3	0.6821	0.7842	0.7786
complete	0.7570	0.0045	0.7912	0.0070	9	2.4306	0.7386	0.7586
average	0.7623	0.0059	0.8170	0.0068	4	1.0423	0.7909	0.7926
Ward's	0.7307	0.0016	0.7328	0.0015	3	0.7090	0.9992	0.9834
McQuitty	0.7668	0.0034	0.8200	0.0059	10	2.547	0.7511	0.7710
model-based	0.7366	0.0032	0.7427	0.0055	3	0.9263	0.9697	0.9565
spectral	0.5970	0.0439	0.7225	0.0147	13	3.8811	0.8967	0.8930
PAMSIL	-	-	0.7813	0.002	4	1.0125	-	0.7657

(Model 7) Figure 4.7a represents a data plot generated from Model 7. Table 4.7 represents the statistics of interest for Model 7. The box and histogram plots of ASW against each initialization method and PAMSIL are plotted in Figure 4.7b and C.7. Model-based clustering gave the best value of ASW whereas single linkage initialization gave the best value for the ASW. However average linkage gave the best value of ARI for both the initial clustering and the OSil clustering. The ARI from the OSil clustering is greater than the ARI for maximum ASW clustering. Single linkage took the minimum time.

(Model 8) Figure 4.8a represents a data plot generated from Model 8. Table 4.8 represents the statistics of interest for Model 8. The box and histogram plots of ASW against each initialization method and PAMSIL are plotted in Figure 4.8b and C.8 respectively. Model-based clustering and PAMSIL clustering gave the maximum ASW, respectively. Model-based clustering gave the maximum ARI values for initial clustering, whereas, the maximum value for OSil clustering was achieved by PAM initialization. The minimum runtime is taken by PAM initialization for OSil. The ARI from the OSil clustering is greater than the ARI for maximum ASW clustering. One thing to note here is that the PAMSIL gave the highest ASW value but too low ARI value.

(Model 9) Table 4.9 represents the statistics of interest for Model 9. The box and histogram plots of ASW against each initialization method and PAMSIL are plotted in Figure 4.9 and C.9, respectively. All the clustering methods reached at the same ASW values except the spectral clustering method. All the methods also gave the ARI value equal to 1 except the spectral clustering method. The minimum time was taken by complete and average linkage initializations.

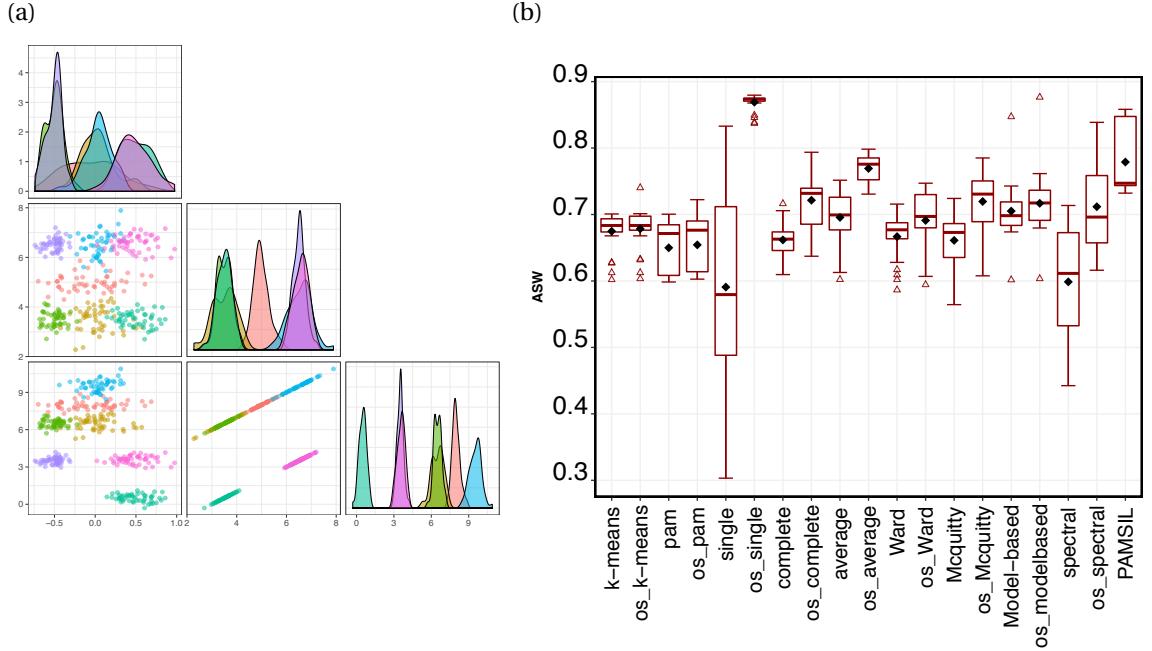


Figure 4.7 (a) represents a synthetic data plot generated from Model 7. (b) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles.

Table 4.7 Results for Model 7 against fixed k .

Methods	ASW (init)	SE	ASW (OSil)	SE	iter	runtime	ARI (init)	ARI (OSil)
k-means	0.6748	0.0055	0.6785	0.006	5	6.0188	0.6434	0.6469
PAM	0.6500	0.0079	0.6543	0.008	8	8.9586	0.6535	0.6509
single	0.5908	0.0275	0.8691	0.0024	5	5.8560	0.5351	0.5315
complete	0.6618	0.0050	0.7215	0.0076	21	22.73	0.6444	0.6701
average	0.6957	0.0075	0.7693	0.0042	10	10.63	0.6839	0.6939
Ward's	0.6668	0.0066	0.6911	0.0083	13	14.65	0.6477	0.6558
McQuitty	0.6611	0.0085	0.7196	0.0098	19	22.19	0.6421	0.6702
model-based	0.7052	0.0083	0.7167	0.0095	7	8.4592	0.6473	0.6537
spectral	0.5987	0.0162	0.7117	0.0129	17	22.65	0.6790	0.6907
PAMSIL	-	-	0.7790	0.0103	6	6.3595	-	0.4923

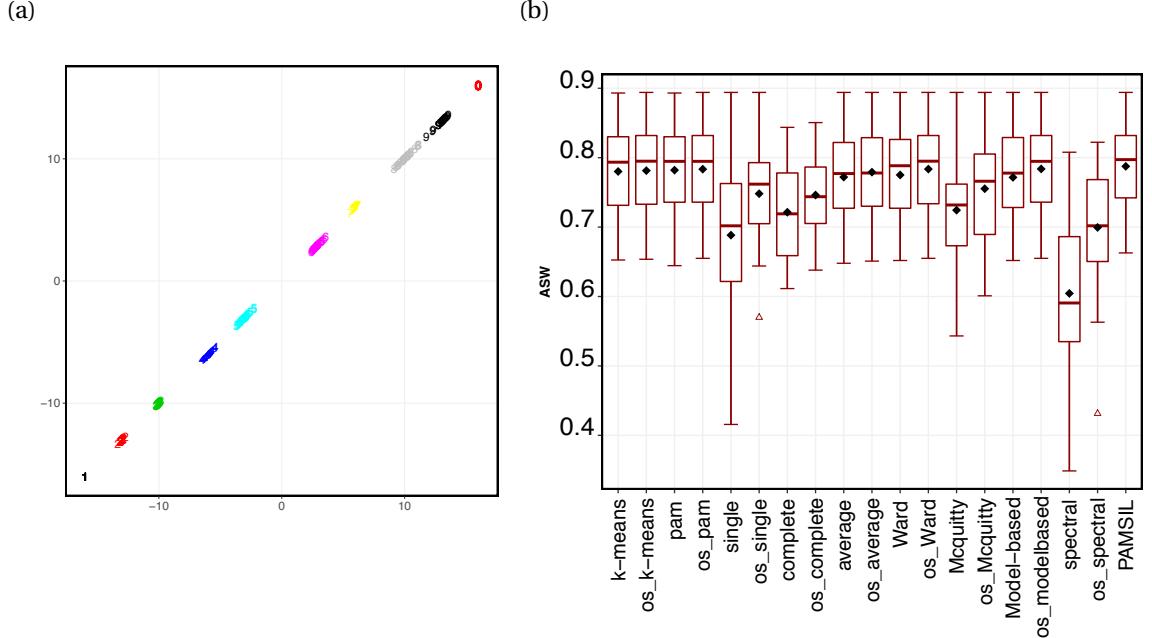


Figure 4.8 (a) represents a synthetic data plot generated from Model 8. (b) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles.

Table 4.8 Results for Model 8 against fixed k .

Methods	ASW (init)	SE	ASW (OSil)	SE	iter	runtime	ARI (init)	ARI (OSil)
k-means	0.7801	0.0123	0.7814	0.0122	3	2.2126	0.9019	0.9101
PAM	0.7820	0.0122	0.7834	0.0119	3	1.7200	0.9145	0.9185
single	0.6884	0.0242	0.7481	0.015	4	2.8365	0.7855	0.7787
complete	0.7214	0.0126	0.7462	0.0115	10	6.5182	0.7963	0.8335
average	0.7720	0.0129	0.7792	0.0122	4	2.6371	0.9028	0.9034
Ward's	0.7751	0.0128	0.7836	0.0120	6	3.5233	0.9139	0.9119
McQuitty	0.7244	0.0159	0.7553	0.0144	11	7.1804	0.8279	0.8495
model-based	0.7716	0.0126	0.7838	0.0119	5	4.0705	0.9418	0.9173
spectral	0.6045	0.0244	0.6995	0.0188	16	10.57	0.7614	0.7608
PAMSIL	-	-	0.7875	0.0114	3	1.8779	-	0.7885

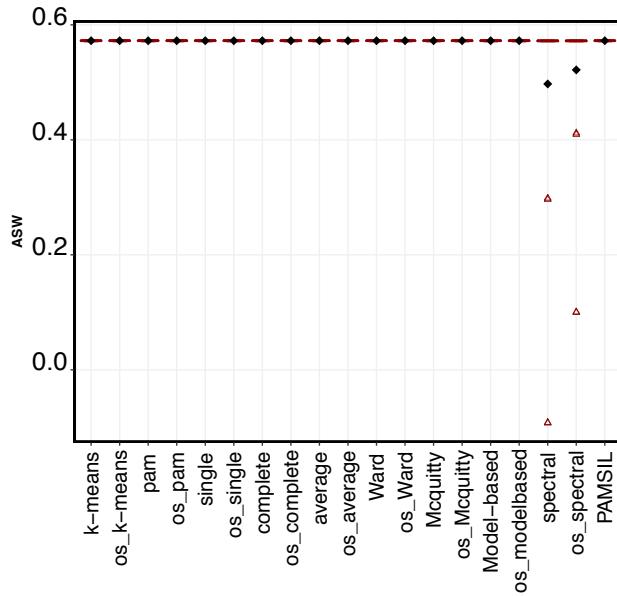


Figure 4.9 Model 9: Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles.

Table 4.9 Results for Model 9 against fixed k .

Methods	ASW (init)	SE	ASW (OSil)	SE	iter	runtime	ARI (init)	ARI (OSil)
k-means	0.5725	1e-04	0.5725	1e-04	1	0.3771	1	1
PAM	0.5725	1e-04	0.5725	1e-04	1	0.0200	1	1
single	0.5725	1e-04	0.5725	1e-04	1	0.0328	1	1
complete	0.5725	1e-04	0.5725	1e-04	1	0.0196	1	1
average	0.5725	1e-04	0.5725	1e-04	1	0.0196	1	1
Ward's	0.5725	1e-04	0.5725	1e-04	1	0.0204	1	1
McQuitty	0.5725	1e-04	0.5725	1e-04	1	0.0204	1	1
model-based	0.5725	1e-04	0.5725	1e-04	1	0.6461	1	1
spectral	0.4972	0.0386	0.5217	0.0269	4	0.2824	0.9113	0.9283
PAMSIL	-	-	0.5725	1e-04	2	0.0450	-	1

(Model 10) Table 4.10 represents the statistics for Model 10 whereas Figure 4.10 and

C.10 display the box and histogram plots. All the clustering methods reached at the same ASW and ARI values, except smaller values for model-based and spectral clustering methods.

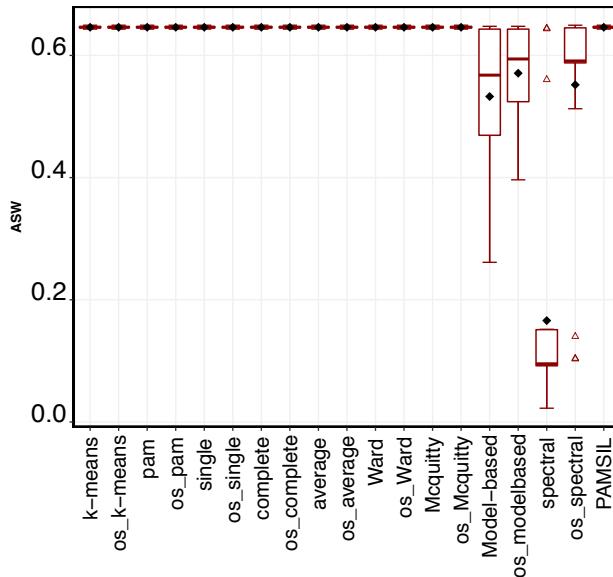


Figure 4.10 Model 10: Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles.

The empirical distribution of ASW values obtained from OSil for various initializations are not same. These are shown as histogram figures which displays the hit counts of the different ASW values obtained at each run, for each initialization method. Note the presence of more than one local optima in each histogram. The distributions are quiet varied ranging form positive skewed, to symmetrical to negatively skewed.

All the methods used for the initialization of the ASW clustering are well developed standalone clustering methods on their own. The ASW method can also be seen as the improvement in clustering quality relative to these clustering methods used as initialization.

4.5.2 Summary

In Simulation I, we have investigated OSil clustering for clustering quality using ASW values for a range of clustering methods and have validated them against an external index using ARI values. The results of simulation I are further summarized in this section in tables and the most important findings are discussed. Comments based on the

Table 4.10 Results for Model 10 against fixed k .

Methods	ASW (init)	SE	ASW (OSil)	SE	iter	runtime	ARI (init)	ARI (OSil)
k-means	0.6461	3e-04	0.6461	3e-04	1	1.51	1	1
PAM	0.6461	3e-04	0.6461	3e-04	1	1.37	1	1
single	0.6461	3e-04	0.6461	3e-04	1	1.35	1	1
complete	0.6461	3e-04	0.6461	3e-04	1	1.34	1	1
average	0.6461	3e-04	0.6461	3e-04	1	1.34	1	1
Ward's	0.6461	3e-04	0.6461	3e-04	1	1.34	1	1
McQuitty	0.6461	3e-04	0.6461	3e-04	1	1.33	1	1
spectral	0.1660	0.0349	0.5520	0.0336	221	284.7	0.4788	0.9123
model-based	0.5327	0.0233	0.5709	0.0152	15	20.09	0.7491	0.805
PAMSIL	-	-	0.6461	3e-04	2	1.73	-	1

clustering methods used as the initialization are made based on their overall performances across all DGPs.

Table 4.11 summarises ARI values and the best ASW value obtained for the clustering methods used as an initialization methods to OSil across all the DGPs. The table indicates which clustering method gave the best ASW and ARI values against the true labels of the data generating process.

The best ASW values were mostly obtained from k -means, PAM or model-based clustering methods whereas the best ARI values were mostly obtained for model-based or Ward's clustering methods. Table 4.12 summarizes which among the initialization methods performed best to optimize ASW¹.

The **single linkage** hierarchical clustering is an attractive clustering method due to its mathematical properties rooted in topology (Carlsson and Mémoli (2010), Carlsson and Mémoli (2013)). For model 6, 7 and 9 single linkage gave the maximum ASW with very poor ARI values. The resulting clusterings were not meaningful as often Single linkage has combined two points together and every thing else in another cluster. We have observed for the models included in this study that they often combined very few points, mostly one or two, together in one cluster that were a bit far from densely populated area of data. This is a verification of well known behaviour of single linkage, which is its tendency of making undesirable long thread-like clusters due to its chaining phenomenon. The chaining occurs because single linkage tries to merge two clusters based on the minimum distance between the two closest elements in the dif-

¹Note that Tables 4.11 and 4.12 are directly comparable and the best of these two are also of interest. These two tables with Table 4.13 can be used to draw the comprehensive conclusions about all DGPs in Simulation I.

Table 4.11 Summary table for Simulation I for ASW and ARI values.

DGMs	<i>k</i> -means	PAM	single	complete	average	Ward	McQuitty	spectral	model-based
Model 1		✓							†
Model 2	✓								†
Model 3		✓							†
Model 4	✓					†			
Model 5		✓ †							
Model 6						†		✓	
Model 7					†				✓
Model 8		✓							†
Model 9	†		✓						
Model 10	✓ †	✓ †	✓ †	✓ †	✓ †	✓ †	✓ †	✓ †	

A ✓ represents a clustering method that gave best ASW for the clustering results obtained from the existing clustering methods whereas a † represents a clustering method which has given best ARI value obtained for these clustering methods.

Table 4.12 Summary table for Simulation I for ASW and ARI values.

DGMs	<i>k</i> -means	PAM	single	complete	average	Ward	McQuitty	spectral	model-based	PAMSIL
Model 1	✓	✓				✓ †		✓ †	✓ †	✓
Model 2	✓	✓				✓			✓ †	
Model 3		✓ †						✓	✓ †	
Model 4					✓					†
Model 5		✓ †								
Model 6			✓			†				
Model 7			✓			†				
Model 8		†						✓		
Model 9			✓		✓				†	
Model 10	✓ †	✓ †	✓ †	✓ †	✓ †	✓ †	✓ †	✓ †		✓ †

A ✓ represents a clustering initialization method that gave best ASW from OSil whereas a † represents an initialization method which has given best ARI for OSil.

ferent clusters. This means that several clusters can be joined together if any two data points, one belonging to each cluster are within close proximity to each other, even though many of other data points in each cluster may be at a far distance. Due to this property single linkage fails in separating the spherical clusters despite the fact that it is a theoretically well developed clustering method. This behaviour is confirmed by the very poor value of the ARI for the models mentioned earlier for single linkage OSil clustering.

The **complete linkage** hierarchical clustering method has given the best ASW and ARI value for one data structure (Model 10) included in this study but this is also achieved by all the other models. When it comes to the consideration about keeping this method as an initialization we have decided to drop this method. McQuitty similarity method

has only given the best ASW once for Model 6 with a very low ARI as compared to the maximum achieved by Ward's method for this model. Other than this it has never given maximum ASW and ARI.

The **spectral clustering** has only improved ASW once (as compared to other clustering methods) namely Model 1 with the best ARI value as well. But many other methods also achieve this best value for the ARI. For all the other models spectral clustering has never achieved the best ASW, and the ARI values were also very low. This is also a computationally expensive method as compared to others. It might be the case that the potential of the spectral clustering method does not come up in the data generating models included here.

We have reported while discussing the results for each model individually in Section 4.5.1 that, the **OSil clustering** has decreased the best ARI values for some DGPs as compared to the best ARI values obtained for the maximum ASW clustering. In particular, the best ARI values from OSil clustering for the Models 1-3, 6 and 8 have decreased. This is an indication that either optimizing ASW is not a good idea for clustering these models if the purpose of clustering is to retrieve the known true clusters. On the contrary, the OSil has increased the ARI for Model 4, 5, and 7. For Model 4 both PAMSIL and OSil performed same. For Model 5, PAMSIL has performed best in terms of ARI, keeping in mind OSil with PAM initialization has also performed very close to this. For Model 7, OSil performed best for ARI. For Models 9 and 10 many methods gave ARI=1. This information is summarized in the Table 4.13.

Table 4.13 Summary table for ARI comparison for ASW and PAMSIL methods in Simulation I.

DGMs	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
ASW	*	*	*			*		*	*	*
OSil				*	*		*		*	*
PAMSIL/OSil					*				*	*

A * represents that a higher value of ARI was obtained from a method as compared to its competitors. If a * appears more than once in a column, this means that all corresponding methods gave the same ARI values.

However, it is interesting to note that, although for Models 1-3, 6 and 8 the best ARI values obtained from OSil has decreased but OSil has improved the individual ARI values for several clustering methods for these and other models. We now summarize these results in the Table 4.14. A ✓, ✗, and = represents that OSil has increased, decreased, and not changed the ARI values when a clustering method was used as an initialization for OSil as compared to the ARI values obtained from the same clustering methods for their standalone use.

Table 4.14 Summary table for the comparison of the ARI values obtained from OSil clustering and the maximum ASW clustering for Simulation I.

DGMs	<i>k</i> -means	PAM	single	complete	average	Ward	McQuitty	spectral	model-based
Model 1	✓	✓	✓	✓	✓	✗	✓	✗	✗
Model 2	✓	✓	✗	✓	✗	✗	✓	✗	✗
Model 3	✓	✓	✗	✓	✗	✗	✓	✗	✗
Model 4	=	✓	✗	✗	✓	=	✓	✓	✓
Model 5	✓	✓	✗	✓	✗	✗	✓	✓	✓
Model 6	✓	✓	✓	✓	✓	✗	✓	✗	✗
Model 7	✓	✗	✗	✓	✓	✓	✓	✓	✓
Model 8	✓	✓	✗	✓	✓	✗	✓	✗	✗
Model 9	=	=	=	=	=	=	=	✓	=
Model 10	=	=	=	=	=	=	=	✓	✓

The **PAMSIL** algorithm has given the best ASW only for Model 8, but with very low ARI (with a difference of 0.13) as compared to the ARI value for OSil with a slightly low (with a difference of 0.0041) ASW value. However, it has achieved the same ASW value as those of OSil for some other models. PAMSIL has overall given best ARI values for Models 3, 4 (higher than OSil as well but lower than ASW), Model 5 (higher than OSil as well as other clustering methods). PAMSIL performed good partially, for Model 3 and 5. For Model 3 it gave the same ASW value as OSil but a higher ARI value but this ARI is smaller than the ARI achieved from existing clustering methods with maximum ASW. For Model 5, OSil and PAMSIL achieved the same ASW values again, but PAMSIL gave the highest ARI value as compared to OSil and existing clustering methods.

Although in about half of the models ASW (both OSil and PAMSIL) approach has reduced the ARI as compared to the ARI got from the existing clustering methods for the maximum ASW, the superiority of OSil as compared to PAMSIL is evident from the following points.

- (i) The ASW value obtained from OSil are much higher than those of PAMSIL for Models 4, 6, 7 and 9. Out of these, for model 4, the ARI for PAMSIL is higher than OSil, whereas the ARI for Models 6 and 9 for PAMSIL was very low as compared to OSil.
- (ii) The models 1 and 2 for which PAMSIL gave similar values of ASW as that from OSil, its resulting ARI values are very low as compared to OSil.
- (iii) For Model 8 PAMSIL gave a higher ASW value but with very low ARI as compared to OSil.

Another important finding regarding ASW is that it is not necessary that a clustering method which will give the maximum ASW will also give the maximum ASW. Many initialization methods for some of the DGPs included in the study have reached the maximum ASW value with several initial ASW they have started with. PAM, Ward's and

model-based clustering methods have given the best ASW values consistently across DGPs, whereas k -means and average linkage have also shown potentially positive performances to be used as initialization methods for optimizing the ASW for some DGPs included in the study.

The empirical distributions of the ASW vary greatly across the initialization methods. The distribution is mostly not symmetric but left or right skewed. For a few times the distribution was also observed to be left or right J shaped for instance see spectral clustering initialization for Model 2.

Lastly, we don't know yet that how each of these initialization methods will perform in terms of the estimation of the number of clusters, which we will study in next Section 4.6.

Before we close this section, we discuss a final observation related to model-based clustering. Among the existing clustering methods considered here, the best ARI values were observed from model-based clustering method for 4 DGPs (Model 1, 2, 3 & 8) despite the fact that it did not give the best ASW for any of these models. We check out whether ASW or ASW brings any improvement in the estimation of numbers of clusters as compared to BIC with model-based clustering method? Thus, we will further investigate how this performance is compared to the number of clusters estimated by the BIC in combination with model-based clustering, and finally what are the ARI values for these clusterings? It would be useful to learn how the performances of model-based clustering with BIC varies for ASW and what the resulting ARI values for these clusterings are? We will investigate this further in Section 4.6.3 once the performance of clustering methods is being explored with respect to estimation of k .

4.6 Simulation II: Estimation of k case

In this section we will conduct simulation for the estimation of the number of clusters. Let the maximum number allowed to estimate the number of clusters is $K \in \mathbb{N}_n$. The number of clusters were estimated for 2 to K clusters. The OSil algorithm was then run for each value of k using the corresponding initial clustering results. The best number of clusters was decided based on the best ASW value obtained for the number of clusters in each case. Note that this was done for all the initialization methods listed in the previous section. For model-based clustering, the number of clusters was estimated using the BIC criterion, and using the maximum ASW criteria. For the data sets having correct number of clusters in the range 2 – 6 we have fixed K at 12, whereas for those having the number of clusters in the range 7 – 10, the clusters were estimated in the window ranging from 2 – 20. We have estimated number of clusters for PAMSIL as well for comparisons.

For the estimation of the number of clusters, we considered the proposed method and a broad spectrum of existing methods for comparisons. We have considered again

the 10 DGPs as defined for simulation I. For each data generating model we have again used all the 9 clustering methods, namely k -means, PAM, five hierarchical methods, namely single, complete, average, Ward's, McQuitty, spectral clustering and model-based clustering methods. The number of clusters were estimated from the 9 clustering methods as standalone methods and then 9 OSil clusterings, one initialized from each of the 9 clustering methods. Thus we have 18 clustering solutions in total, for each simulated data set of a DGP. For each of these clusterings, we have noted clustering results and statistics of interest for known fixed k as well as estimated k , which was estimated based on maximum ASW values. In addition we have calculated clustering and estimated number of clusters from PAMSIL.

Among the already existing estimation methods for the number of clusters, we have used H, Gamma, C, KL, CH, Gap, Jump, PS, BI, CVNN, model-based, ASW along with OSil with 11 clustering methods. For all the values of k ranging from 2 to K the clusterings were first calculated from all the 9 clustering methods in each run of the simulation and then these were used to pass to OSil to optimized ASW. These calculated clusterings for each value of k were also used to estimate the number of clusters from all the estimation indices for the estimation of the number of clusters. For H, KL, CH, Gap index, the functions `index.H()`, `index.KL()`, `index.G1()`, `index.Gap()` of “clusterSim”, for Gamma “clusterCrit”, for PS, BI, CVNN functions `prediction.strength()`, `ns-electboot()`, `cvnn()` of “fpc” packages were used, respectively, and for the Jump method [Sugar and James \(2003b\)](#) implementation (code available from their website) was used. For PS, BI, and Gap the parameters ‘M’(numbers of time the data set is divided into two halves), ‘B’(number of times to resample from data) and ‘B’(the number of reference data sets to compute the gap statistic, see [2.5.1.6](#)) of their R functions were fixed at 15 each. For Jump method six transformation powers were used, namely $p/2$, $p/3$, \dots , $p/7$ (see [2.5.1.7](#) for detail). The model-based and spectral clustering methods are not available to estimate the number of clusters with the Gap statistics with it’s current R implementation. Also note that the use of model-based clustering for the estimation of k is two-fold here. Firstly, we have estimated the number of clusters using all the estimation indices included in the study with model-based clustering. Secondly, the number of clusters was also estimated using the BIC criterion with model-based clustering as implemented in [Scrucca et al. \(2017\)](#) for comparison. Therefore, in a single run of a simulation we have estimated the number of clusters, from range 2 to K, with 105 methods (10 indices \times 9 clustering methods + Jump method with 6 transformation powers + PAMSIL + model-based clustering with BIC –2 (exclude two clustering methods for Gap method) + 9 (OSil initialized with 9 clustering methods)) for a single data set. In total 105×25 (runs) $\times 10$ (data models) = 26,250 times the numbers of clusters were estimated. All the simulations were done on a 2.8 GHz Intel core i7 processor.

The discussion hereafter is divided into two themes. First we will report the values of ASW and other statistics of interest to make a comparison of ASW values for the

estimation of k , and then we will investigate how each of these methods performed with estimation indices for the estimation of the number of clusters.

4.6.1 ASW Results

The results for ASW and ASW values are overall consistent with what we got earlier. For Model 6-8 many methods were able to give the best ASW and ARIs and there was no further increment observed for the value of ASW from OSil. In the following subsection we discuss the performance of each of cluster estimation methods. Throughout the explanation, the percentage performances of indices are reported. If an index estimates the known number of clusters correctly for 15 out of 25 runs, the index performance rate is 60% and so on. The bars in the charts in Appendix C.2 also represent this percentage. In the discussions below whenever we refer to the standard use of model-based clustering as proposed in [Fraley and Raftery \(1998\)](#), we will refer to it together with BIC criterion. If model-based occurs with a particular index, say CVNN, we mean that the number of clusters there are estimated by CVNN not by BIC.

Figure C.11a - C.20b represent the box-plot and density plot for the estimation of k for Model 1 to Model 9. Tables C.1, C.2, C.3, C.4, C.5, C.6, C.7, C.8, C.9 and C.10 represent the mean ASW with their SEs and ARIs for Model 1 to Model 10, respectively. The summary Tables 4.15 and 4.16 are prepared based on the results mentioned in these tables.

Note that Tables 4.15 and 4.16 are also comparable and best of these is also of interest. These two tables can be integrated with Table 4.17 to draw the conclusions about all DGPs. For instance, for Model 1, for the estimation of k , k -means clustering method gave the best ASW value (first row of Table 4.15), whereas Wards method gave the best ASW value (first row of 4.16), keeping in mind that the ASW value obtained was greater than ASW value. Next the best ARI value achieved was from McQuitty clustering method against maximum ASW value and this value was higher than the best ARI achieved from maximum ASW value (first row of Table 4.17). Similarly, the conclusions about the other DGPs can also be drawn from these three tables.

Table 4.15 Summary table for Simulation II for ASW and ARI.

DGMs	<i>k</i> -means	PAM	single	complete	average	Ward	McQuitty	spectral	model-based
Model 1	✓								†
Model 2	✓			†					
Model 3		✓							†
Model 4						✓ †			
Model 5		✓ †							
Model 6		✓ †	✓ †	✓ †	✓ †	✓ †	✓ †		✓ †
Model 7	†	✓	✓	✓	✓	✓	✓		
Model 8		✓ †	✓ †	✓ †	✓ †	✓ †	✓ †		✓ †
Model 9	†				✓				
Model 10		✓ †	✓ †	✓ †	✓ †	✓ †	✓ †		

A ✓ represents that an initialization method gave the best ASW from clustering methods on average for 25 runs and estimated *k* from 2 to K whereas a † represent if that initialization gave the best ARI value for clustering.

Table 4.16 Summary table for Simulation II for ASW and ARI.

DGMs	<i>k</i> -means	PAM	single	complete	average	Ward	McQuitty	spectral	model-based	PAMSIL
Model 1						✓			†	
Model 2					✓					✓
Model 3						✓			†	✓ †
Model 4	✓					†			✓	✓
Model 5	✓ †									✓ †
Model 6		✓ †	✓ †	✓ †	✓ †	✓ †	✓ †		✓ †	✓ †
Model 7	†	✓	✓	✓	✓	✓	✓		†	✓
Model 8		✓ †	✓ †	✓ †	✓ †	✓ †	✓ †		✓ †	✓ †
Model 9	✓								†	
Model 10		✓ †	✓ †	✓ †	✓ †	✓ †	✓ †			✓ †

A ✓ represents that an initialization method gave best ASW from OSil on average for 25 runs and estimated *k* from 2 to K whereas a † represent that an initialization gave the best ARI for OSil.

Table 4.17 Summary table for ARI comparison for ASW and PAMSIL methods in Simulation II.

DGMs	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
ASW	*	*		*		*		*	*	*
OSil			*		*	*	*	*	*	
PAMSIL/OSil				*	*			*	*	*

A * represents that a higher value of ARI was obtained from a method as compared to its competitors. If a * appears more than once in a column, this means that all corresponding methods gave the same ARI values.

4.6.2 Comparison with other indices

All the discussions in this section are based on the results presented in the figures and tables in Appendix C.2. Figures 4.11, 4.12, 4.14, 4.16, 4.18, 4.20, 4.22, 4.24, 4.25 and 4.26

represent the bar charts for the number of clusters estimated at the desired value by all the combinations of clustering methods and estimation indices for all the DGPs using the percentage counts. Since many of these combinations fail to estimate the number of clusters at the desired level for these DGPs, detailed tables are prepared to represent the exact counts for the estimated numbers of clusters for each of these combinations. These Tables are [C.11](#), [C.12](#), [C.13](#), [C.14](#), [C.15](#), [C.16](#), [C.17](#), [C.18](#), [C.19](#) and [C.20](#), which represent the frequency count for each combination for the estimated number of clusters for Models 1-10. We now discuss the overall performance of the individual indices with respect to various clustering methods.

CH has not performed very well. Ward's, McQuitty and spectral clustering have performed well with CH only for Model 1. For the rest of the models, CH has either failed to estimate k with many clustering methods or has performed below 40%.

H index has performed very poorly for all models except for Model 8. PAM has never estimated the correct number of clusters with the H index except for Model 8. The **Gamma** and the **C** indices have consistently performed poorly except for Model 4 & 8. These two indices have failed with many clustering methods in the estimation of the number of clusters. **KL** has also performed poorly with all clustering methods.

Gap method has performed above 60% except with single linkage for Model 1. Gap in combinations with all clustering methods has performed lowly for Model 2, poorly for Models 3, 5, 6, 7 and well for Model 4 (except with single linkage).

Jump has estimated the correct number of clusters with p/3 (87.5%) for Model 1, p/3 (97%) for Model 2, p/5 (58%) for Model 3, p/2 (100%) for Model 4 and 6 and never for Model 5, 7, 8, or 9.

PS has performed poorly with complete linkage clustering. It has performed 100% with model-based clustering for Model 1, with k -means and model-based clustering for Model 2, with PAM and model-based clustering for Model 4, with PAM for Model 5, with PAM, Ward and model-based clustering for Model 6, with PAM, single, complete, average, Ward, McQuitty and model-based clustering for Model 8. It has estimated the desired number of clusters for Model 3 with k -mean, PAM and Ward about 3%, single linkage 37%, average linkage about 12%, McQuitty about 10%, model-based clustering about 82% and for Model 7 only with single linkage (about 30%). PS has never been able to estimate the numbers of clusters at the desired value for Model 9.

BI has never been able to estimate the correct number of clusters for Model 6. Only a few clustering methods performed well in combination with this index for Model 1 and

2. Model-based clustering with BI has never been able to estimate the correct number of clusters. BI has performed well only for Model 4 with all clustering methods except *k*-means and single linkage clustering.

CVNN has performed well only for Models 1, 2, 3, 4 (except single linkage). It has performed poorly for Models 5, 7, 8, 9 in combinations with all clustering methods. It has performed well with Model 6 only with Ward's and model-based clustering.

BIC in combination with model-based clustering method has estimated the correct number of clusters 100% of the times for Models 1, 2, 6, 35% for Model 3, less than 30% for Model 4, very poorly for Model 5 and never for Models 7, 8 and 9.

ASW shows an overall good performance with Models 1 and 2, a very good performance for Models 4, 8, and 10, and a poor performance for model 3. It also performed well for Model 5, but only with a few clustering methods, and it was never able to estimate the correct number of clusters for Models 6, 7, and 9. ASW mostly showed better performance than PAMSIL in combination with *k*-means and spectral clustering.

PAMSIL has estimated the correct number of clusters for 100% of the simulations for Models 4, 5, 8 and 10. The performance rate is 80% for Model 1, 28% for Model 2, 12% for Model 3 and never for Models 6, 7, and 9.

OSil has 88% performance rate for Model 1 for the estimation of number of clusters. It has shown good performance (100%) for the estimation of number of clusters for Models 4, 5, 8, 9, 10 with various initialization methods. It performed poorly for Models 2, Model 3(estimated number of clusters as 2 instead of 3 majority of the times), 6 (always estimated 4 as a number of clusters instead of 5), 7 (always estimated number of clusters as 3 instead of 7).

Single linkage has consistently performed poorly for all the estimation procedures for the estimation of the number of clusters for all DGPs except very few.

We now expand the discuss about the performance of indices and clustering methods in more detail with respect to each DGP. Other possible artificial clustering solutions for the DGPs are also considered.

(Model 1) OSil performed better for Model 1 as compared to all clustering methods with ASW, except a slightly smaller value of ASW for spectral clustering method only. PS and BI in combination with *k*-means and model-based clustering methods and BIC also performed very well for this model. Many other methods failed to estimate the number of clusters at desired level or performed very poorly. Table C.11 represents the

number of clusters estimated by each of these methods.

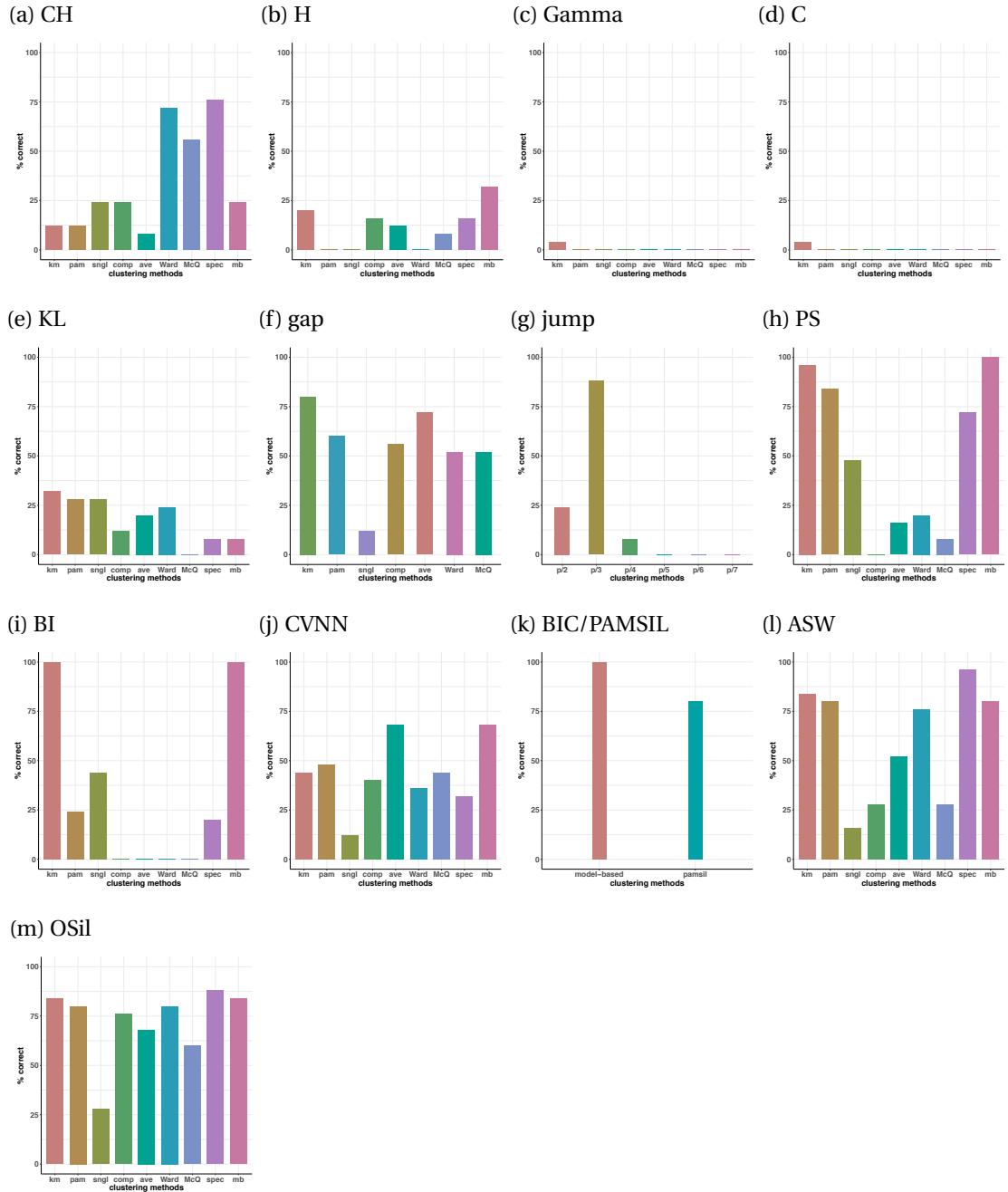


Figure 4.11 Bar plots for the estimation of k for Model 1. Each bar represents the percentage count of correct estimate of k .

(Model 2) The Jump method, PS, and BI in combination with k -means and model-based clustering, CVNN with model-based clustering and BIC with model-based clustering have performed best for Model 2. OSil increased the performance rates of the single, complete, Ward's and McQuitty methods. Table C.12 represents details of the numbers of clusters estimated by these methods.

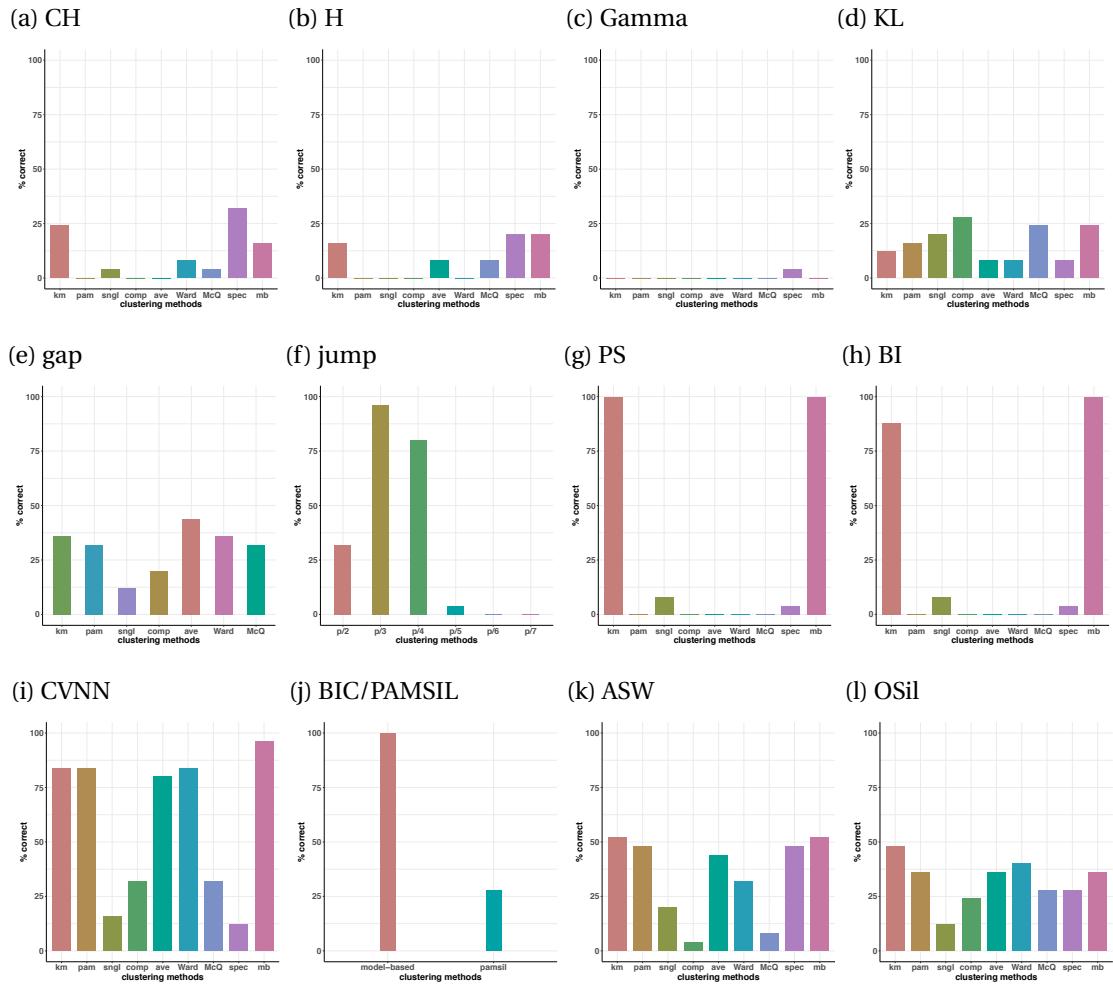


Figure 4.12 Bar plots for the estimation of k for Model 2. The C index was never able to estimate correct number of clusters for Model 2.

(Model 3) Only CVNN with model-based clustering has been able to estimate the number of the clusters at desired level 100% of the times for Model 3. All the other methods performed poorly. However, OSil increased the value of ASW in combination with k -means, average, McQuitty, spectral and model-based clustering methods. Overall the

ASW family for this model didn't work well like many other estimation methods. Table C.13 represents details of the numbers of clusters estimated by these methods. For Model 3 the distances between the clusters are very small and a clustering solution with 2 clusters or 3 clusters also makes sense, despite the fact that the data was generated originally as 4 clusters. The 2-clusters and 3-cluster are shown in the Figure 4.13a and 4.13b below.

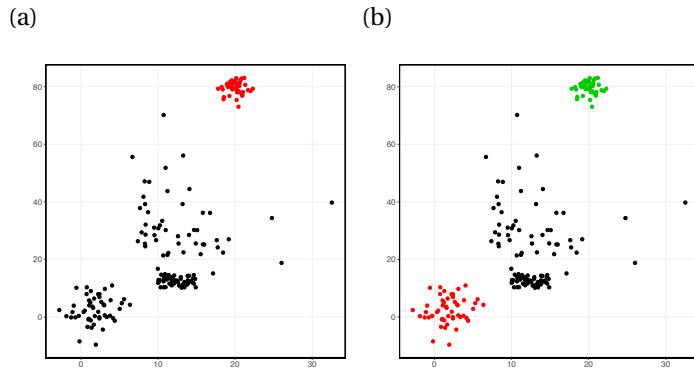


Figure 4.13 Artificially generated clustering solutions for Model 3. Figure (a) a 2-cluster solution, and (b) a 3-cluster solution.

The Gap method has estimated the 3-clusters solution most of the time for this model. KL has shown a greater frequency for the numbers of clusters 2 and 3 as compared to the other methods. PS and BI have shown the same trend. CVNN has a majority trend for the numbers of clusters 3 or 4. ASW based clustering approach have proposed 2 as an estimate with higher frequency with all the clustering methods than any other method indicating the distance between the clusters as mean factor for this index. Some methods have also break the cluster with bigger observation spread, which is t distributed here, into two smaller clusters as shown by the Jump method. Some other methods like CH, C and Gamma have further split the clusters into more clusters and propose an even higher numbers than these as the estimate. Methods that are trying to form clusters with bigger spread can prefer the cluster solution shown in Figure 4.13a over the solution shown in Figure 4.13b. For a combination of clustering method and the estimation index, that heavily depend upon the distance between clusters and within cluster distances the solution in Figure 4.13b will be preferable.

(Model 4) For Model 4, ASW has estimated the correct number of clusters 100% of the times in combination with PAM and Ward, whereas OSil has estimated k 100% of the times with PAM, Ward and model-based clustering. OSil has also increased the percentage of all clustering methods, for the estimation of the correct k, as compared to, estimation of the number of clusters from these methods using ASW. The model-based

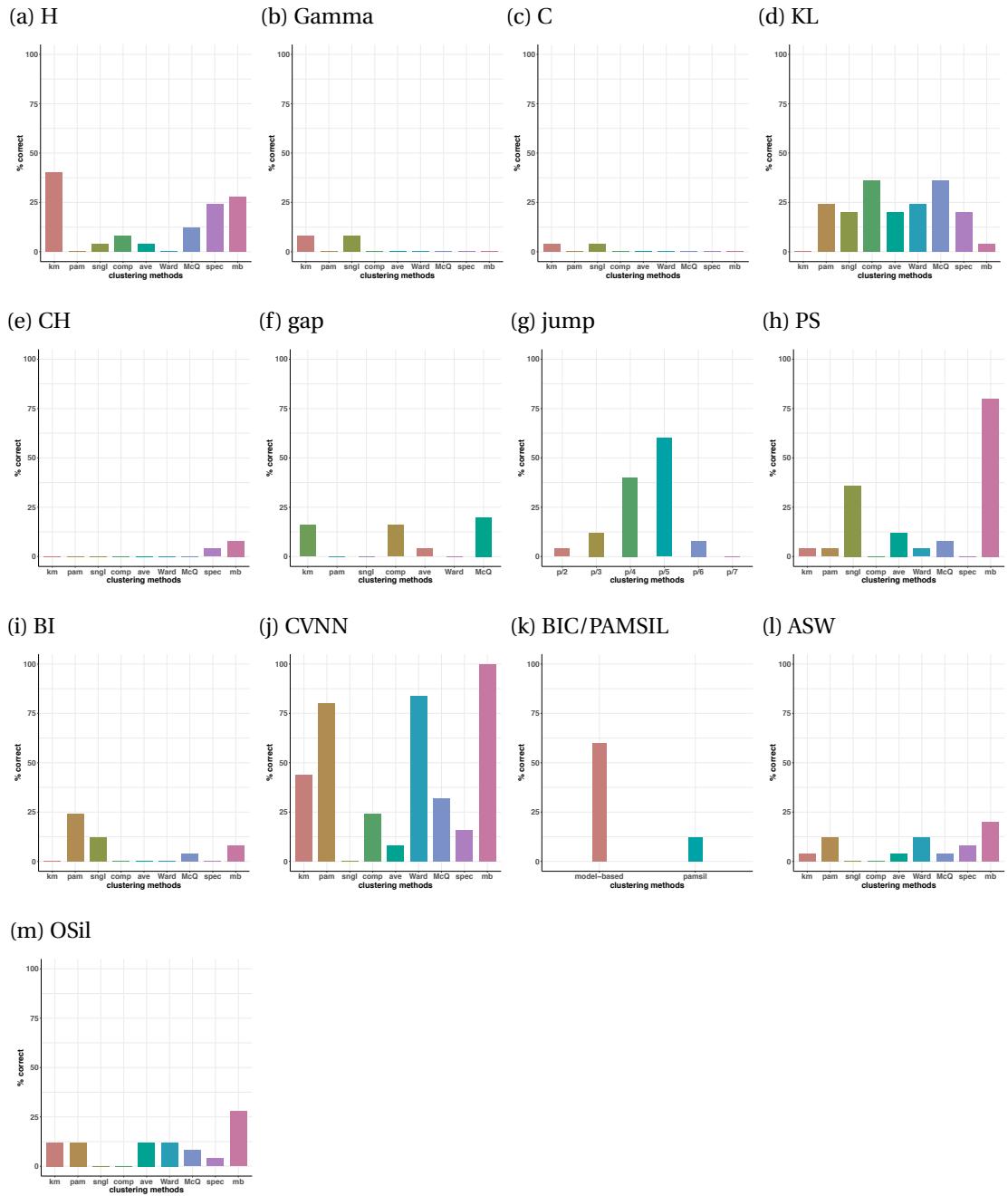


Figure 4.14 Bar plots for the estimation of k for Model 3.

clustering method in combination with BIC has estimated the correct k for about 45% of the times, whereas model-based clustering in combination with ASW and OSil has estimated the correct k about 95% and 100% of the times respectively. Table C.14 represents details of numbers of clusters estimated by these methods. Figure 4.15 represents the 4, 3, and 2 clusters' clustering solutions for Model 4.

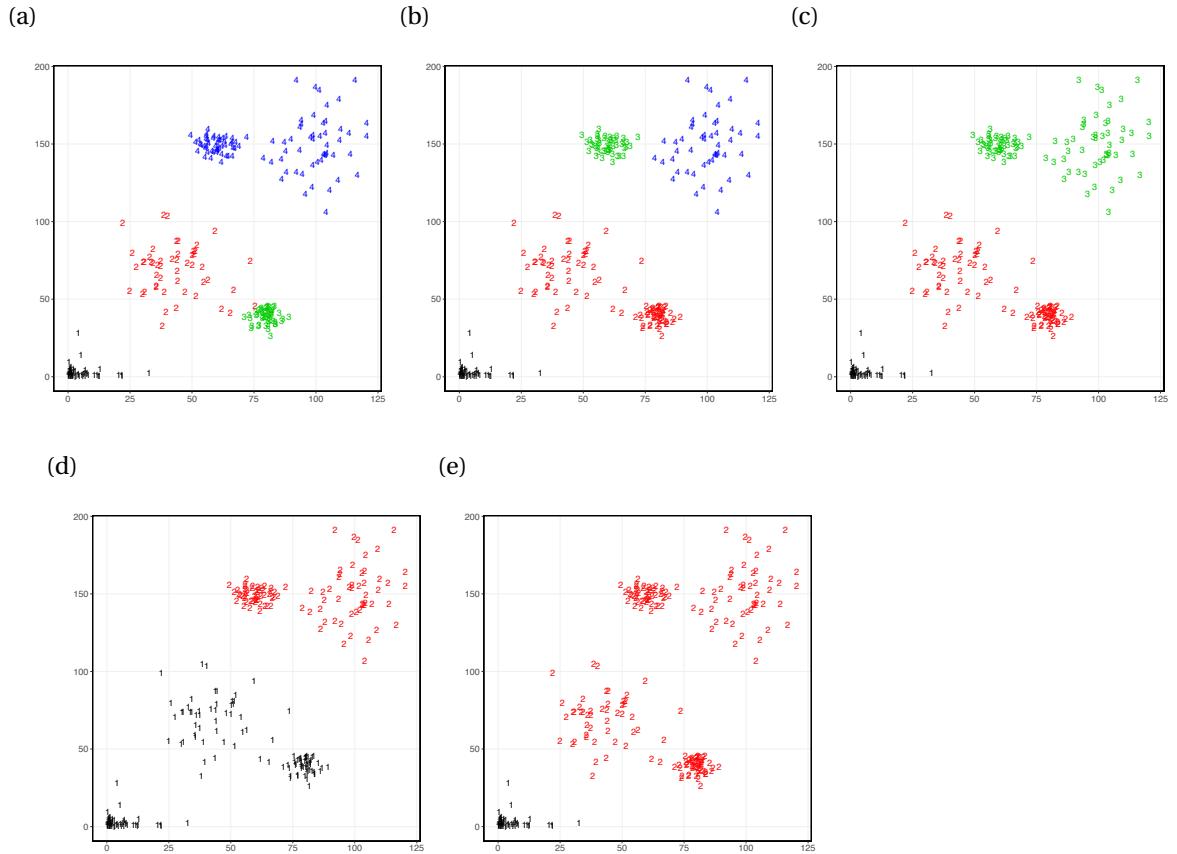


Figure 4.15 Artificially generated clusterings for Model 4 other than true number of clusters. Figure (a) a 4-cluster solution, (b) another 4-cluster solution, (c) a 3-cluster solution, (d) and (e) two different 2-cluster solutions.

(Model 5) This model has 6 clusters, where each cluster has the same number of observations. The structure of clusters, distance between clusters, and spread of observations within clusters vary greatly as clusters are generated from different distributions like Uniform, Gamma, Beta, Exponential and Weibull distributions. Only PAMSIL and OSil with PAM were able to estimate the correct number of clusters 100% of the times for Model 5. ASW in combination with PAM performed 95%. *k*-means, Ward's, spectral methods in combination with both ASW and OSil performed below 50%. The remain-

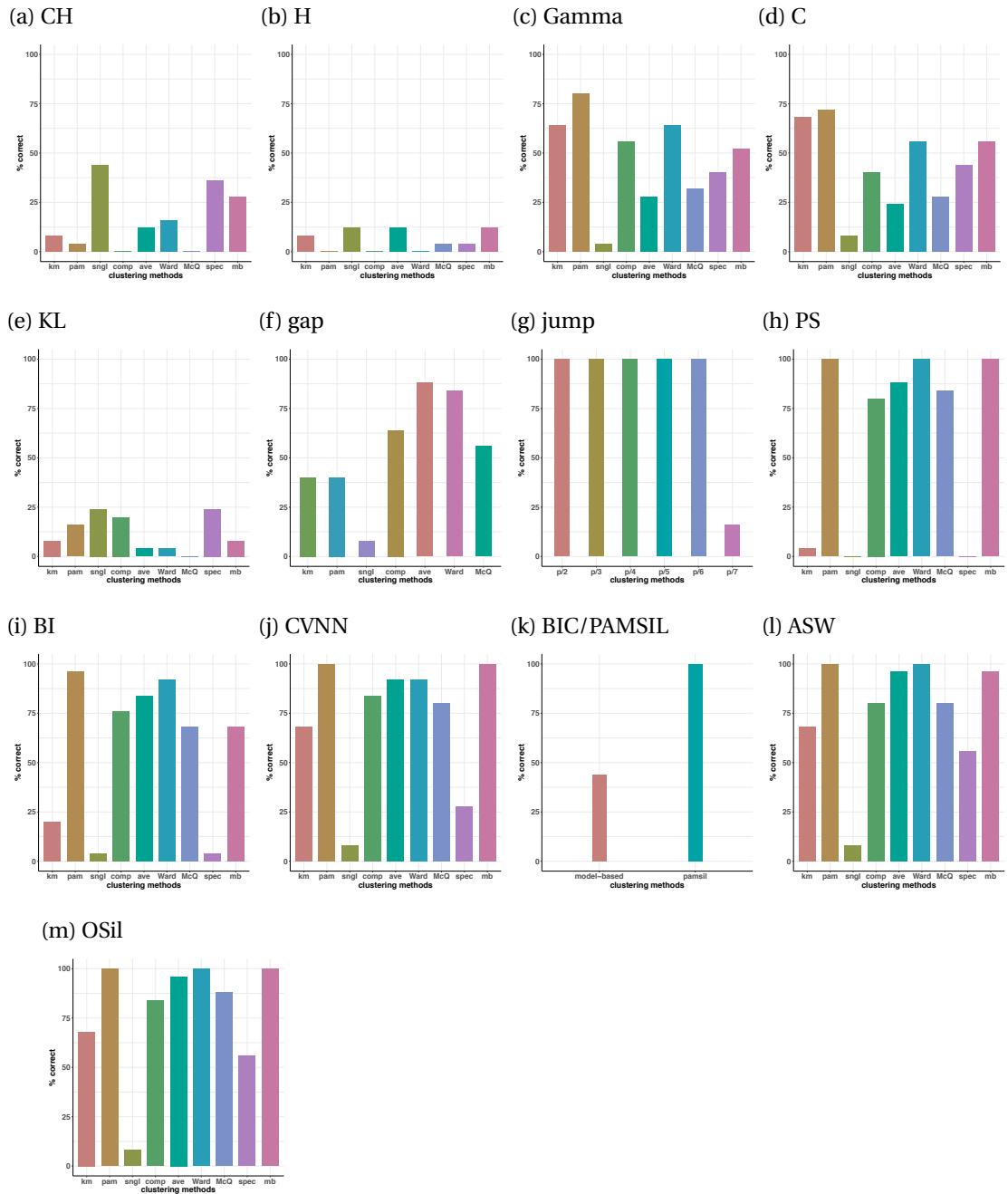


Figure 4.16 Bar plots for the estimation of k for Model 4.

ing clustering methods in combination with ASW or OSil were never able to estimate the correct number of clusters. Model-based clustering in combination with BIC performed below 3%. Table C.15 represents the number of clusters estimated by each method. Since the distances between the clusters are not the same, some estimation methods or clustering methods can merge the two nearest clusters as one cluster resulting in fewer number of clusters, any value from 5 to 2 instead of 6. Figure 4.17 represents these other possible clustering solutions for Model 5.

Figure 4.17b shows just the four clusters which lie in the middle of data sets generated from Model 5. The clusters are generated from Unifrom (label 1 in figure), Exponential (label 2), Beta (label 3) and skewed normal (label 4) distributions to highlight the fact, that for a five cluster solution the exponential cluster is closer to the beta cluster to the Weibull and skewed normal clusters, and it is more intuitive for a method to combine Uniform and exponential cluster first in a cluster. Figures 4.17d and 4.17e both represent two different 3 cluster solutions for this model. The Uniform cluster and the pair of Weibull and skewed normal clusters are roughly equally distant from the pair of Exponential and Beta clusters hence the two different 3 cluster solutions. Figure 4.17f represents a two clusters solution. Note that although the Gamma cluster is also almost at the same distance from the central four clusters as compared to the Uniform cluster, it has a wider spread, it is more natural for the methods to combine the Uniform cluster with these 4 central clusters rather than the Gamma cluster. However, OSil has always estimated 6 clusters for this model.

(Model 6) Model 6 is a particularly difficult model for the estimation of the number of clusters. Recall that there are 5 clusters in 5 dimensions. Two clusters are very close to each other in all dimensions. The remaining three clusters are also close to each other (but not equally distant) and far from the other two clusters. These clusters contain equal numbers of observations but the spread among the observations within clusters is not the same. PAMSIL and ASW were never able to estimate the correct number of clusters for Model 6. Only OSil with k -means (11%) was able to estimate correct number of clusters. Note that although model-based clustering performed 100% for the estimation of k with BIC, it was never able to estimate the correct k with ASW or OSil. Other methods, namely Gamma, C, and BI were never able to estimate the correct k in combination with any clustering method. KL, Gap, and CVNN were able to estimate the correct number of clusters with a very low performance rate and with only a few clustering methods. CH, PS, in combination with complete, McQuitty and model-based clustering, Jump with ($p/2$ and $p/3$), and model-based clustering with BIC were able to estimate k with 100% rate.

Table C.16 shows the estimated numbers of clusters according to these indices for Model 6. For the majority of the clustering methods there is no agreement over the number of clusters by indices. The estimation of k with the H index with various clus-

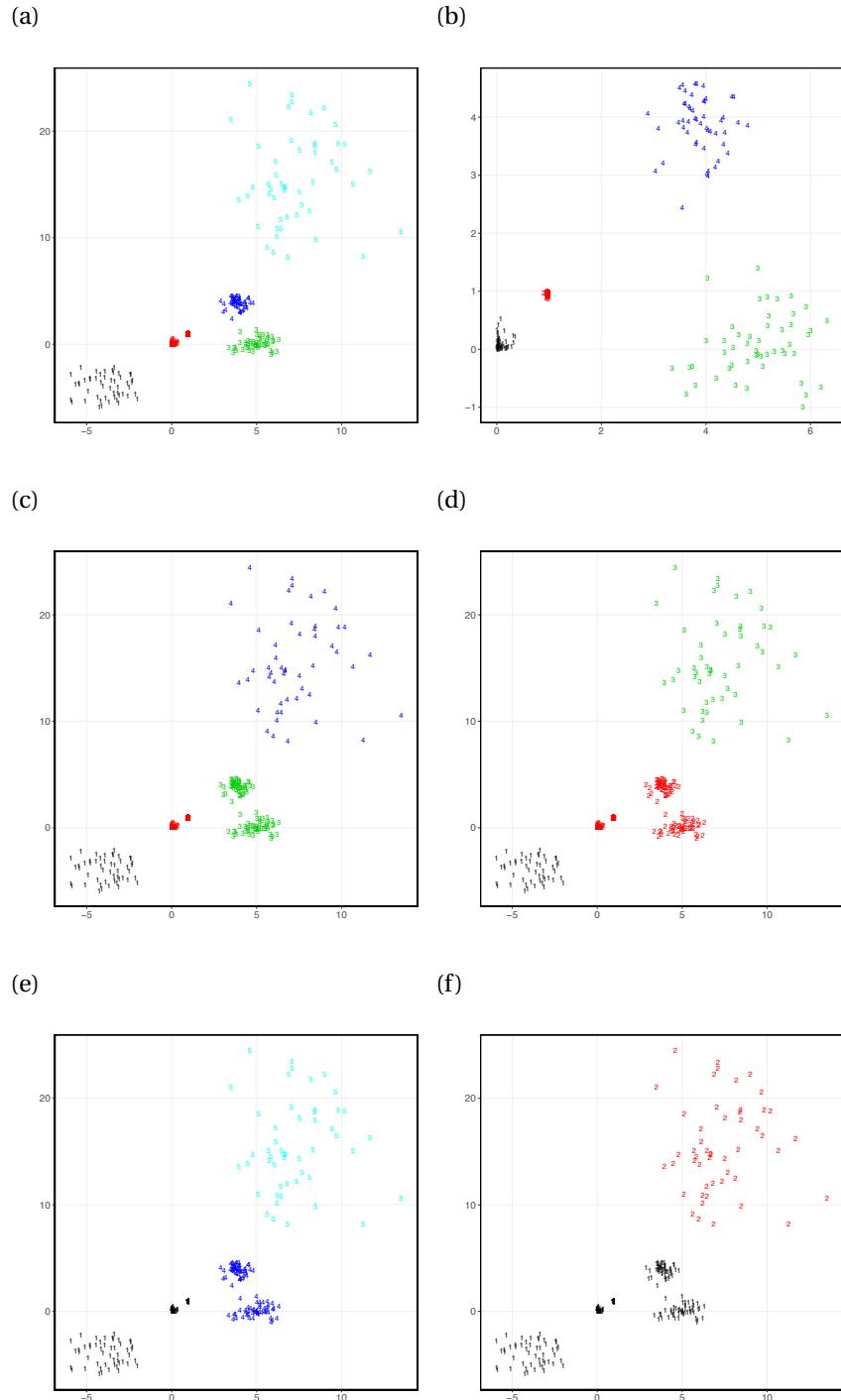


Figure 4.17 Artificially generated clustering solutions for Model 5 other than true number of clusters. Figure (a) represents 5-cluster solution, (b) the middle four clusters, (c) a 4-cluster solutions, (d) and (e) two distinct 3-cluster solutions, and (f) 2-cluster solution.

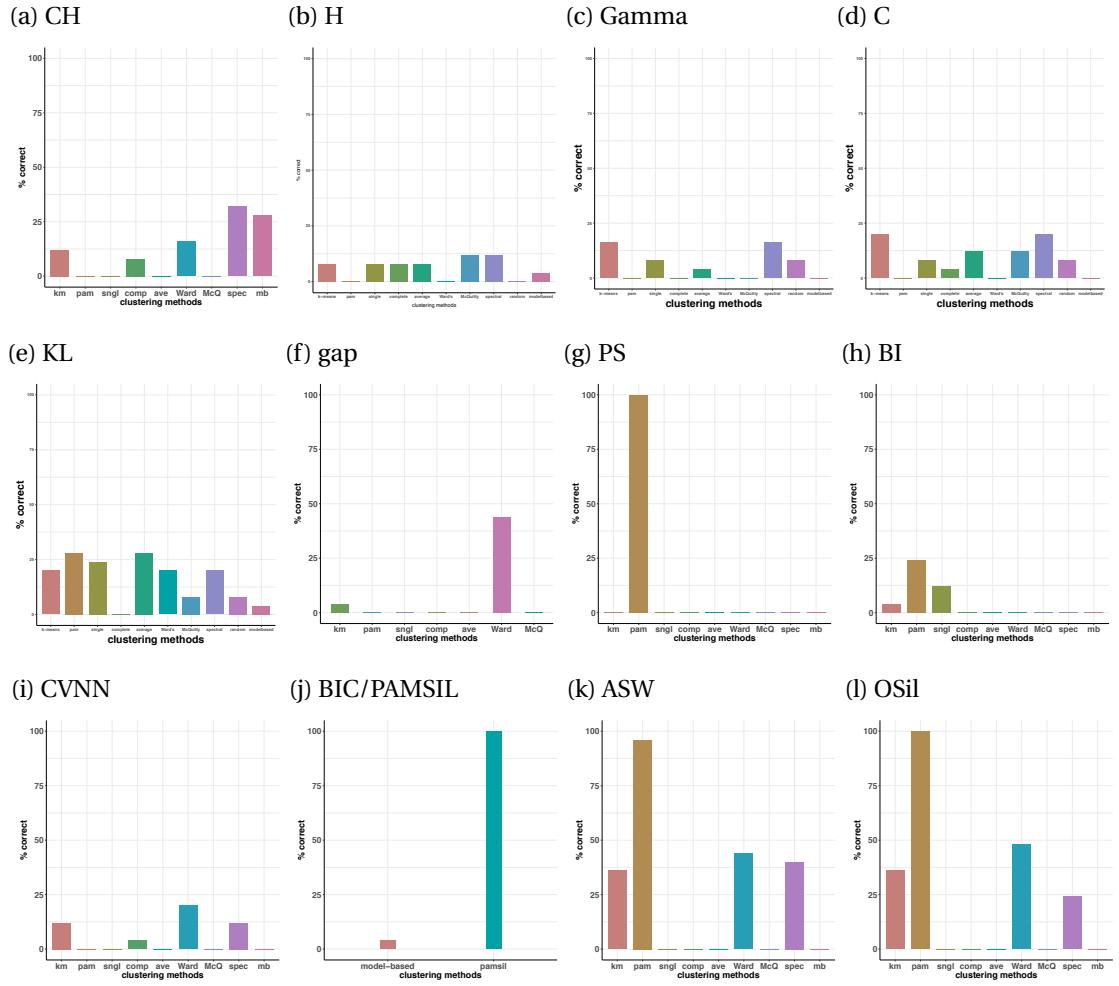


Figure 4.18 Bar plots for the estimation of k for Model 5. The Jump method was never able to estimate correct number of clusters for Model 5.

tering methods varies greatly. Gamma, C and BI always estimated the number of clusters as 2. KL has estimated 2, 3, or 4 clusters. Only Ward's, PAM, and model-based clustering in combination with CH have 100% performance rates. All the other clustering methods with CH have estimated 4, 5, 6, or 7 clusters. With the Gap method the majority of the clustering methods has estimated 4 as the number of clusters. Jump (with $p/2$ and $p/3$), PS (with Ward's, PAM, model-based clustering) and model-based clustering (with BIC) have 100% performance rates. PAMSIL has always estimated 4 as the number of clusters. ASW has estimated 4 as the number of clusters a majority of the times. k -means and spectral clustering with ASW have estimated 2 as the number of clusters 44% of the time each. OSil has always estimated 4 as the number of clusters as well except for some occasional variations with k -means and spectral clustering.

Figure 4.19 shows 2, 3, and 4 cluster solutions for Model 6. As a three cluster solution, any solution from Figures 4.19b or 4.19c or 4.19d can occur because two out of three clusters are always very close to each other resulting in three possibilities. However, for the 4-cluster solution, since two clusters are close to each other as compared to the other three, therefore, 4.19e is the most intuitive possibility.

(Model 7) All clustering methods in combinations with all estimation methods have performed very poorly in estimating 7 clusters for Model 7. Table C.17 shows the estimated number of clusters by each index. The majority of the methods have estimated the number of clusters from 2-5. Figure 4.21 shows the three cluster solution for Model 7.

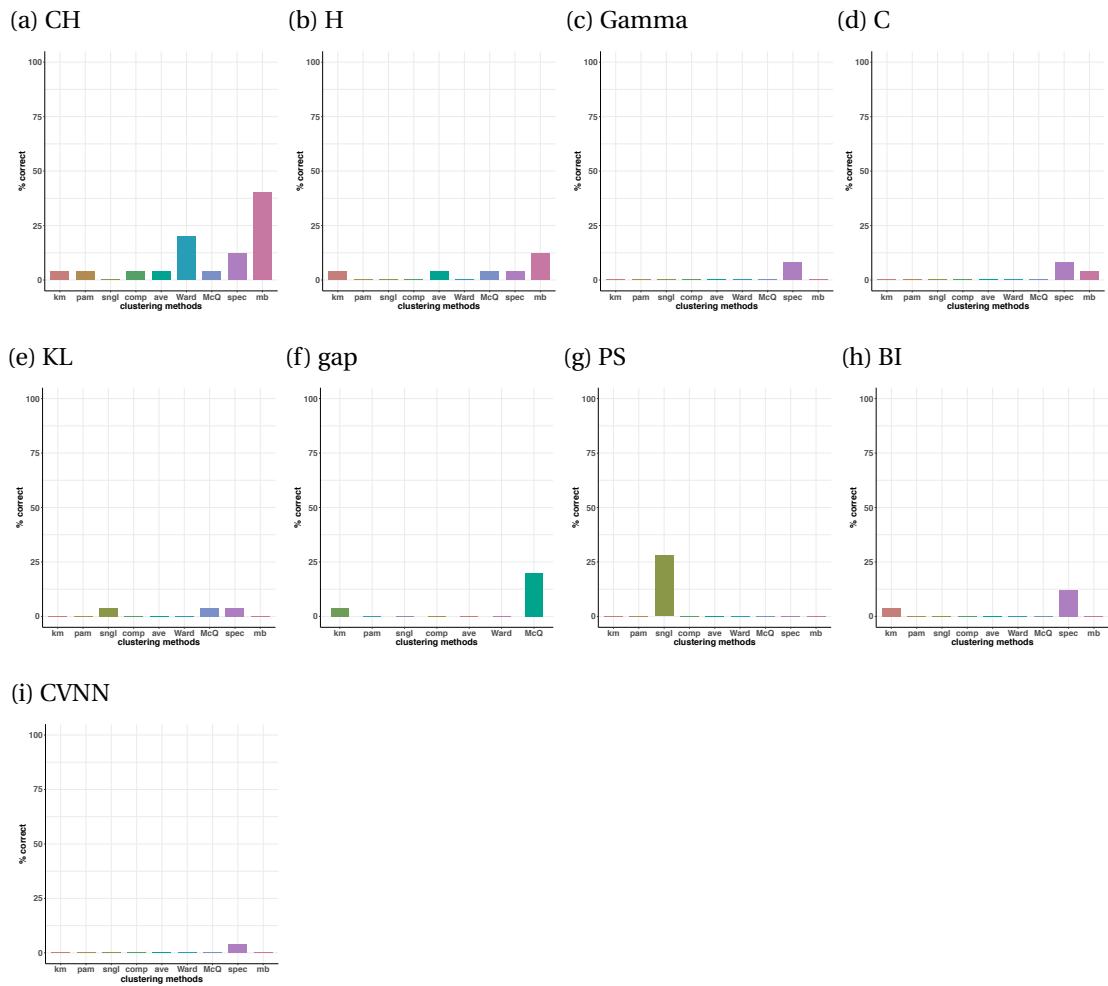


Figure 4.22 Bar plots for the estimation of k for Model 7. The Jump, model-based clustering with BIC, PAMSIL, ASW and OSil were never able to estimate correct number of clusters for Model 7.

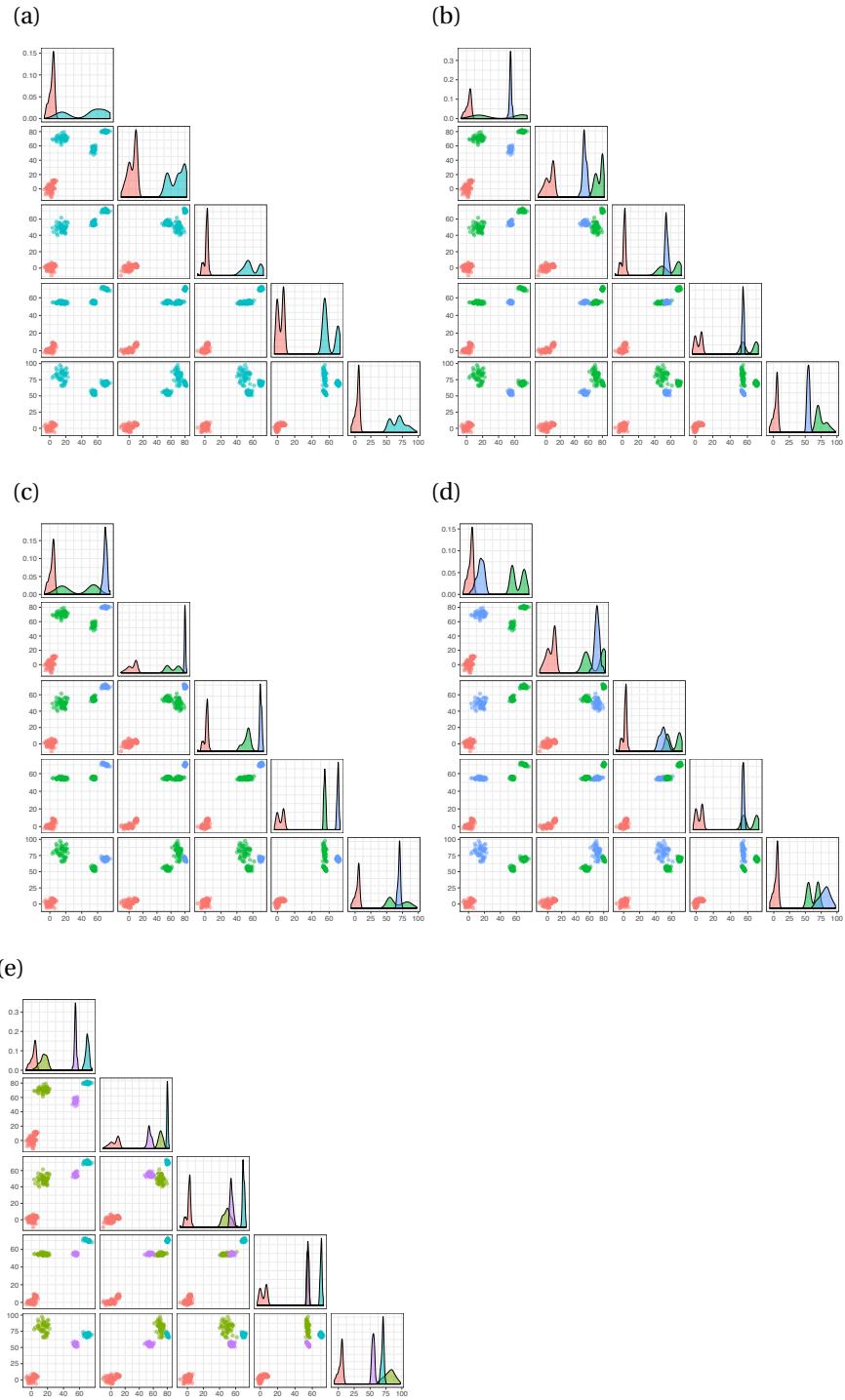


Figure 4.19 Artificially generated clustering solutions for Model 6 for various number of clusters. Figure (a) a 2-cluster solution, (b), (c), and (d) the three different 3-cluster solutions, and (e) a 4-cluster solution.

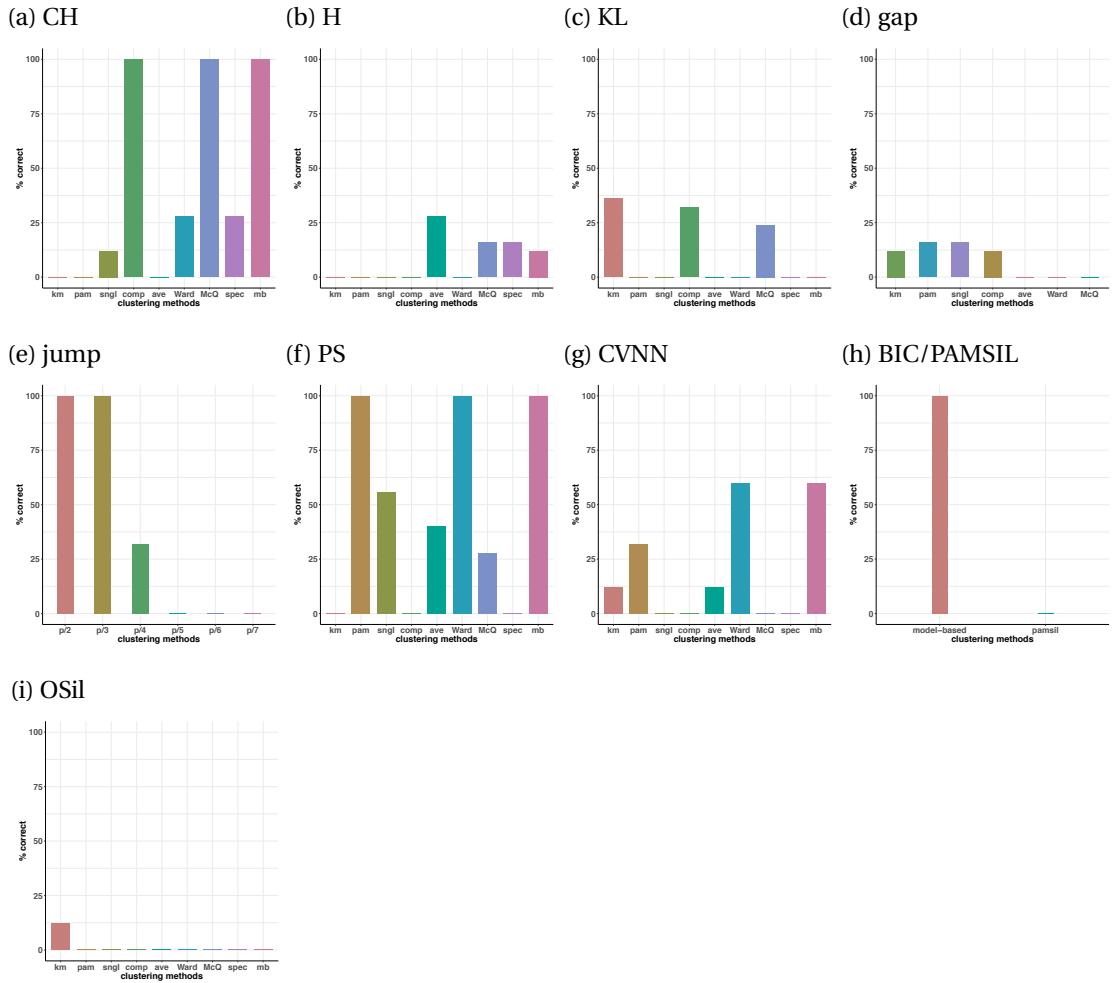


Figure 4.20 Bar plots for the estimation of k for Model 6. The Gamma, C, BI, PAMSIL and ASW were never able to estimate correct number of clusters for Model 6.

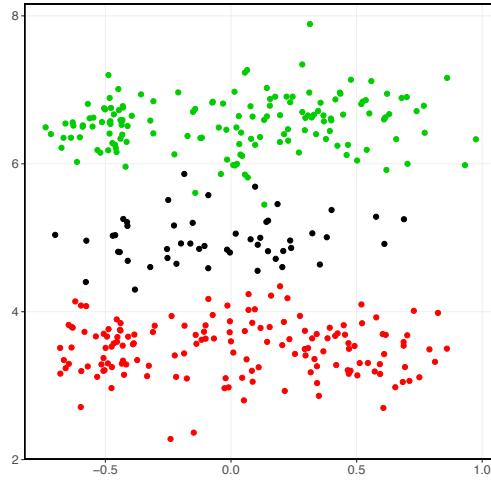


Figure 4.21 An artificially generated 3-cluster solution for Model 7. Shown are the first two dimensions of the data.

(Model 8) This model has 10 clusters. PAMSIL, ASW and OSil (except with k -means and spectral clustering) have 100% estimation rates. KL, gap, jump, CVNN, model-based (with BIC) were never able to estimate the correct number of clusters. CH and BI have performed poorly as well. H and PS (never with k -means, spectral), Gamma and C (never with k -means, and very low with spectral clustering) have estimated the correct number of clusters 100% of times except with those mentioned above. Table C.18 shows the estimated number of clusters with each estimation method in combination with each of the clustering methods with the frequencies for each estimated k . Most of the methods have mostly estimated 8, 9, 11 or 12 clusters. KL has estimated 8 or 9 clusters. CH has estimated 10, 11 or 12 clusters. Jump has always estimated 5 number of clusters. BI has estimated 2 or 4 clusters. Model-based in combination with BIC has estimated 9 clusters. CVNN has shown a very poor performance here and has estimated the number of clusters between 2 to 8.

For Model 8 there are 2 other different numbers of clusters solution possible as shown in Figure 4.23. For this model the design is of such kind that the 2 cluster solution and 4 cluster solution looks intuitive. Since there are two global clusters and the difference between their means is highest in the data therefore 2 number of clusters makes sense here as well. Also since each of these global clusters further is of such kind that the difference between 2 groups of 3 clusters, and 2 groups of 2 clusters are same, methods can estimate 4 clusters as well, as shown in the right panel of the figure.

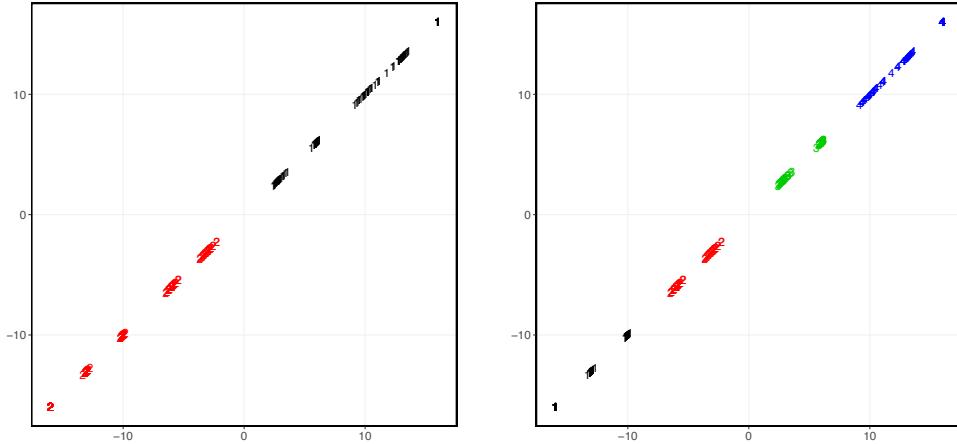


Figure 4.23 Artificially generated clustering solution for Model 8. Left panel a 2-cluster solution, and right panel a 4-cluster solution.

(Model 9) This model has three clusters of equal sizes in 1000 dimensions. Overall the estimation methods have not performed well here. A few methods failed entirely or performed very poorly. The H was never able to estimate three clusters. BI (with k -means, PAM, single and average linkage), CVNN (with spectral method) were never able to estimate 3 clusters. KL, BI and PS have performed poorly. Many indices have shown low performance for the estimation of the correct number of clusters with the k -means clustering methods. Table C.19 displays the details on the estimates of k. A majority of the methods have shown 100% performance except those mentioned.

(Model 10) For this model the majority of combinations agrees. Many methods have estimated the correct k except, the H index has estimated 6 clusters instead of 7. KL has either estimated 5 or 6 clusters a majority of times. Jump has always estimated 5 clusters. BI with k -means has shown a 6-cluster solution a majority of times.

4.6.3 Summary

- The distance between clusters turned out to be a very significant characteristic for many clustering methods and indices to estimate the correct number of clusters. Many clustering methods, especially H, Gamma, C and KL performed badly for the models with unequal difference between clusters' locations. Also, varying spread among the observations between the clusters is hard for many combinations to determined correctly. This includes different shaped clusters in the data like compact as well as wide clusters or other shapes like as generated from Uniform distributions. The Gaussian clusters with different shapes and orientations across the different dimensions were not identified by the methods correctly. Even the methods that are

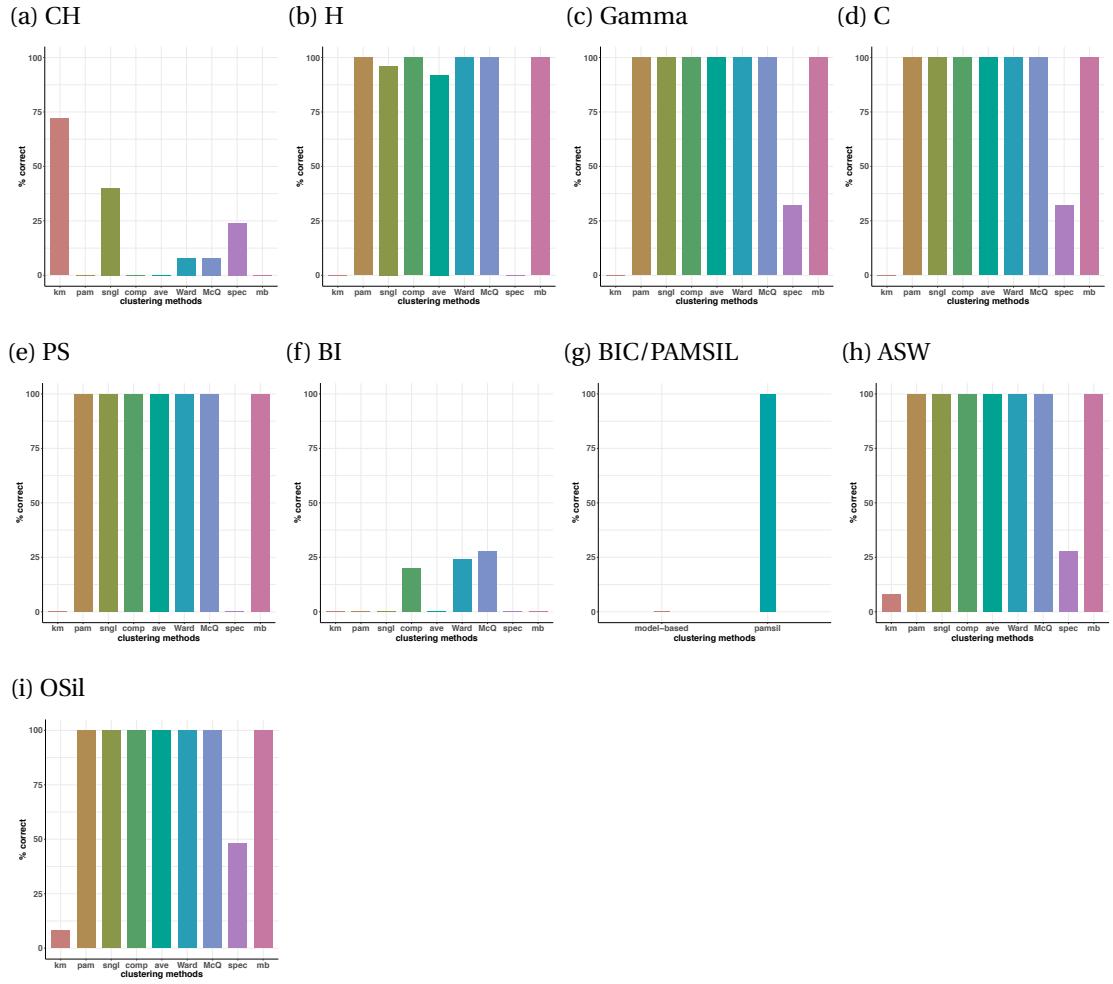


Figure 4.24 Bar plots for the estimation of k for Model 8. The KL, Gap, Jump, CVNN were never able to estimate correct number of clusters for this model.

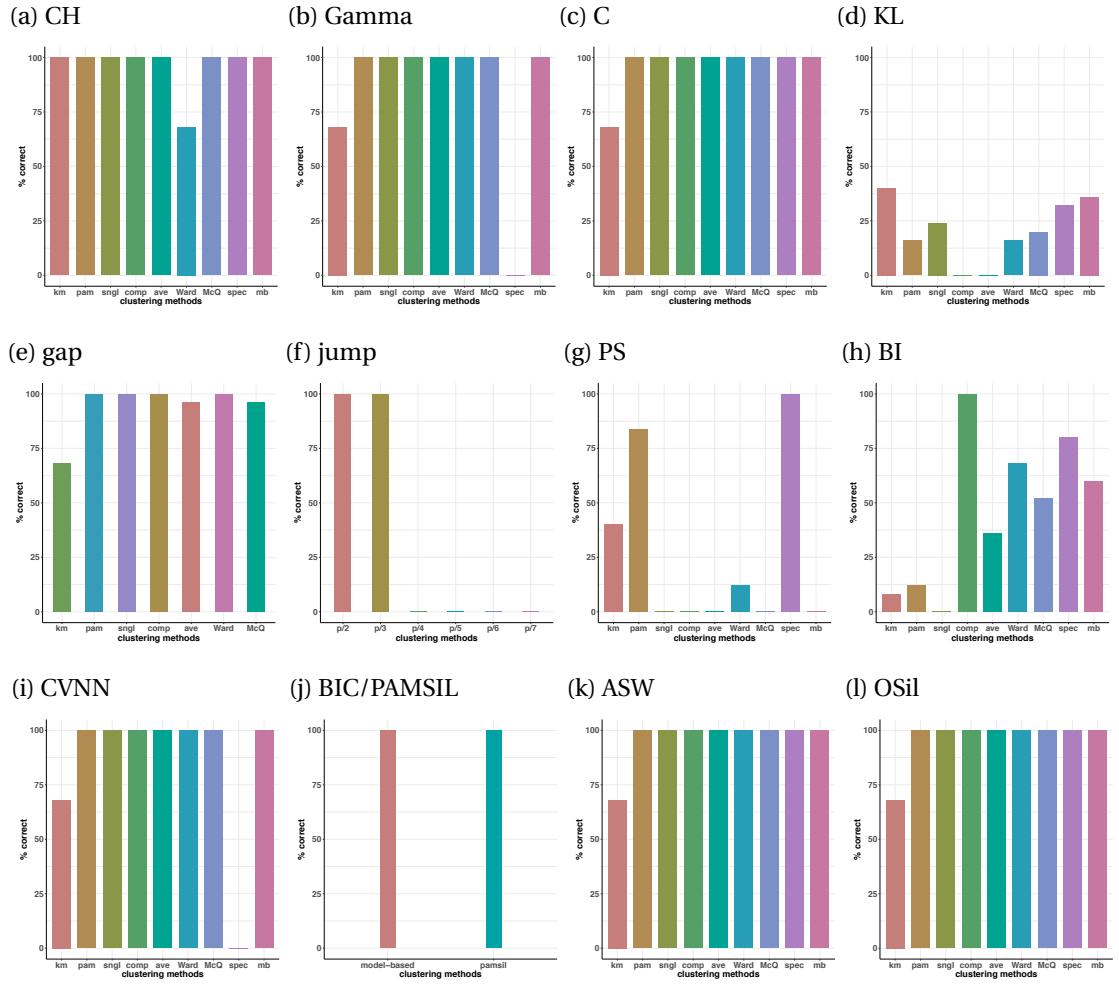


Figure 4.25 Bar plots for the estimation of k for Model 9. The H index was never able to estimate 3 number of clusters for Model 9.

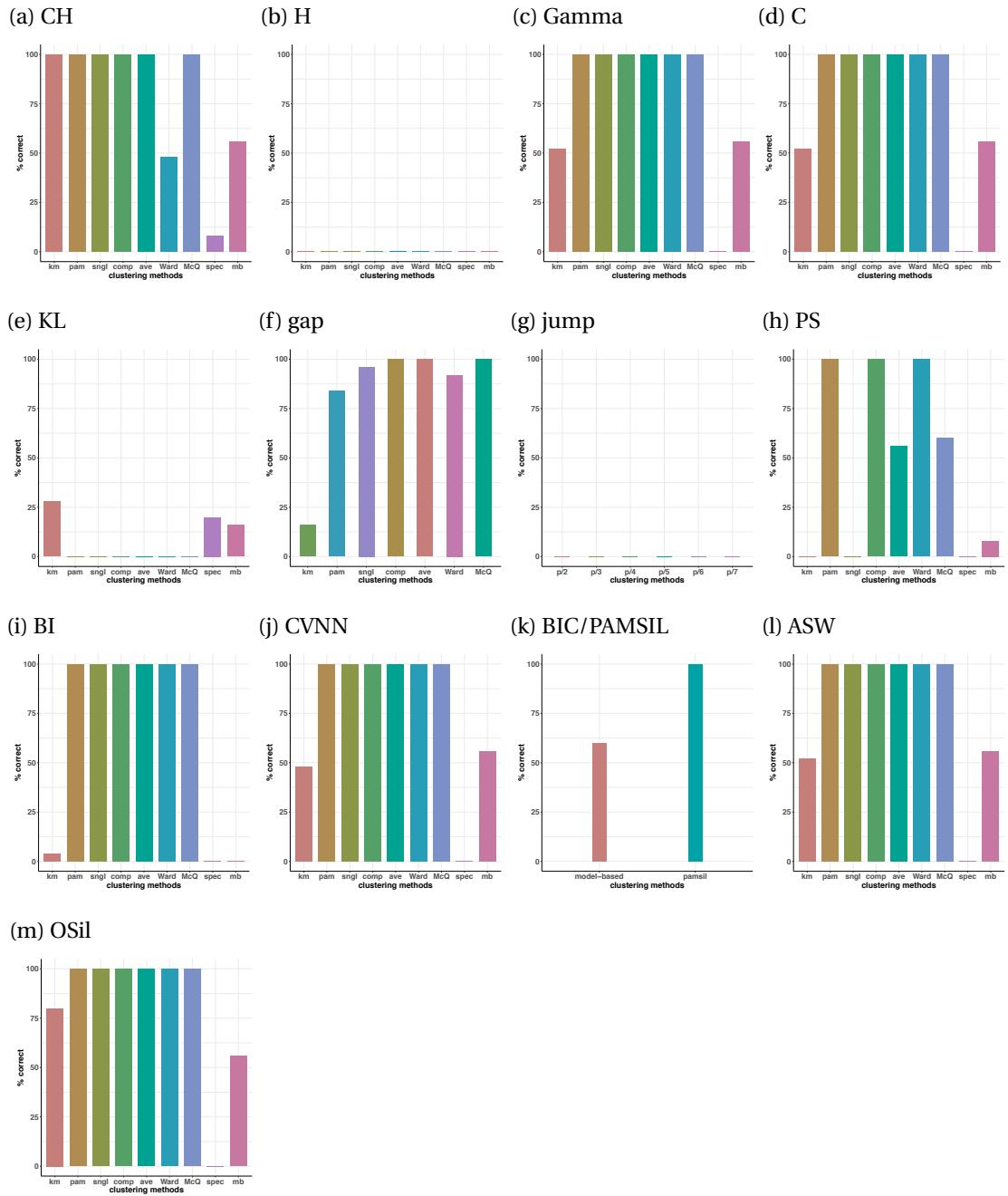


Figure 4.26 Bar plots for the estimation of k for Model 10.

designed especially for Gaussian data failed quite often, for instance, see the model-based clustering results for Models 6, 7, 8, and 9. For ASW based clustering objective functions (including OSil and PAMSIL), the difference between the clusters mean locations turns out to be a challenging situation.

- Another important observation is that when some method can't estimate the known or correct number of clusters for a DGP, they choose the maximum number in the range (i.e., K) as the estimate of k. This is mostly the case for C, Gamma, and CH index.
- A ranking of the indices included in the study is plotted in Figure 4.27. The bars in the plot is made from the block sums of each index in Table C.21. For instance, for CH sum of overall column is 543 and success rate is $(543/25*9*9)*100$. The model-based clustering with BIC, PAMSIL and Jump methods were not added in the figure as they do not corresponding to the same total.

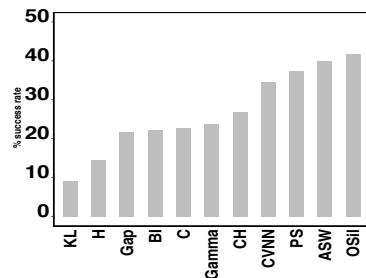


Figure 4.27 Overall results for the indices and clustering methods aggregated for Simulation II across all DGPs.

- The choice of clustering method with the indices matters and has an effect on the estimation of the number of clusters. For instance, for Model 2, PS and BI were never able to estimate the desired number of clusters using PAM, complete, single, Ward's, McQuitty and spectral. On the other hand they were able to estimate clusters at the desired level 100% of the times with k-means and model-based clustering. There is plenty of other such evidence. The performance of each index with one model across these clustering methods differs greatly.

As defined in the above paragraph, various indices has shown good performance with a particular clustering method only. Figure 4.28 shows the % success rate of each clustering index with the top performing clustering method. One best row for each index in Table C.21 was used to construct the plot.

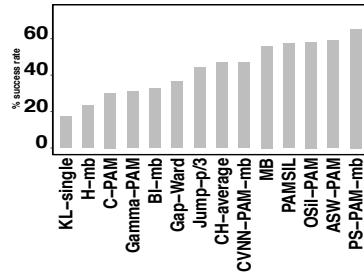


Figure 4.28 Overall results for Simulation II aggregated over all DGPs for the indices in combination with clustering methods.

- Table 4.18 shows the performance of OSil as compared to ASW for all the clustering methods and DGPs for the estimation of number of clusters. The comparison is done using the percentage performances rate (PPR) of estimating the correct numbers of clusters. OSil has either performed the same or better than standard use of ASW with all the clustering methods except at a new occasions.

Table 4.18 Performance comparison of ASW values obtained from OSil as compared to other clustering methods for all DGPs.

DGMs	<i>k</i> -means	PAM	single	complete	average	Ward	McQuitty	model-based	spectral
Model 1	=	=	✓	✓	✓	✓	✓	✓	✗
Model 2	✗	✓	✗	✓	✗	✓	✓	✓	✗
Model 3	✓	=			✓	✓	✓	✓	✗
Model 4	=	=	=	✗	=	=	✓	✓	✗
Model 5	=	✓				✓			
Model 6	✓								
Model 7									
Model 8	=	=	=	=	=	=	=	=	✓
Model 9	★	★		✓		★		★	
Model 10	✓	=	=	=	=	=	=	=	

=, ✓, ✗ represent the same, increase, and decrease in percentage performance of OSil as compared to ASW, respectively. ★ means only OASW was able to estimate the correct number of clusters, whereas an empty box represent that neither OSil nor other clustering methods using ASW were able to estimate the number of clusters at the desired value.

- One important observation regarding the use of ASW with spectral clustering is that the OSil initialized with spectral clustering has slightly decreased the performance rate as compared to ASW obtained from standalone spectral clustering method for Models 1 to 5. ASW has shown a poor performance with the spectral clustering method to estimate the correct number of clusters for Model 3 (8%), has shown good performance for Model 2 (48%), Model 4 (56%), Model 5 (40%) and Model 8 (28%). This combination has a performance rate of 96% for Model 1. However, this combination never worked for Models 6, 7, and 9. OSil has further reduced the performance rate for the estimation of *k* with spectral clustering for Models 1 to 5. This combination also never worked for Models 6, 7, and 9. However, OSil only improved

the performance rate for Model 8 from 28% up to 48%. Overall, ASW approach does not look a good fit for spectral clustering to estimate number of clusters.

- Table C.21 presents aggregated results for all the estimation indices included in the present study together with all the clustering methods across all the DGPs. One thing to keep in mind before using this table is that the purpose of presenting these results is not to compare the overall count calculated by taking into account all the clustering methods of each index across DGPs. These results in this manner are not comparable across indices, because not all of them contain an equal number of clustering methods. Therefore, if an overall ranking of these indices is required, best performing row should be used to create such a ranking rather than the overall count at the end of each block. For instance, for CH the maximum count is with model-based clustering, which is 60, thus only this value should be used to determine the rank of CH among the estimation indices. However, the overall count at the end of each block can be used for overall performance comparison of estimation indices that have the same number of clustering methods. Spectral clustering performed poorly with most of the indices for all DGPs (see Table C.21). It only performed a bit better with OSil (87/225), ASW (93/225) and CH(78/225) as far as estimation of number of clusters is concerned.
- OSil has always increased the frequency count for all the methods to estimate the clusters at the desired value except a slight decay for model-based clustering. The OSil approach is indeed better than the use of ASW with standalone clustering methods for the estimation of the number of clusters, as the overall counts for the two methods with all the clustering methods included are 1061 and 1026, respectively.
- The index that showed the best performance is PS with PAM (168 count) with a 75% success rate. ASW with PAM clustering (158 count) being second with a 70% success rate. OSil with PAM on third rank (156 count) with 69% success rate, and PAMSIL being 4th (154 count) at 68% success rate. Many indices other than these performed closely: model-based with BIC (151), PS with model based clustering (146 count), ASW with Ward (140 count), OSil with Ward (139), CVNN with PAM (131 count), and CVNN with model-based clustering (130 count).
- For the Jump method, $p/3$ is the best transformation power choice for the majority of DGPs. Estimation of the numbers of clusters for the spectral clustering method is the big challenge. Many indices with this clustering method have performed poorly for the estimation of the number of clusters.
- It was observed that PAMSIL has given a higher ASW value for a few models but with too low ARI as compared to OSil. So far we have learnt that OSil is better than PAMSIL for the estimation of the number of clusters. Finally we look into which method

among the ASW family (ASW, OSil and PAMSIL) gives the best ARI for each DGP when the number of clusters are estimated from these. For many DGPs more than one methods have given the same ARI. The purpose here is to see how each of these methods performs in terms of obtaining clustering solutions for the estimated number of clusters. The results are summarized in Table 4.19 below based on the ARI values reported in Tables C.1, C.2, C.3, C.4, C.5, C.6, C.7, C.9, and C.10.

Table 4.19 Best ARI values indication among ASW and OSil methods for the estimation of k for each DGP.

Methods	DGPs									
	1	2	3	4	5	6	7	8	9	10
8 clusterings	✓	✓		✓		✓		✓		✓
PAMSIL					✓	✓		✓		✓
OSil(8 initializations)		✓		✓	✓	✓	✓	✓	✓	✓

The first row in the table above shows the one best ASW value based on the 8 existing clustering methods as mentioned already. OSil produced the best ARI for the maximum number of models. Thus OSil not only produces better value of the ASW, it has also shown a better performance for the number of clusters estimation as well. Also, if we only compare OSil with PAMSIL, it has produced much higher values of ARI as compared to PAMSIL, see for instance, Model 1, 2, 3, 7, and 9. It is also able to achieve the same ARI for the remaining models (4, 5, 6, 8) as PAMSIL.

- We now continue our investigation on model-based clustering left at the end of the Subsection 4.5.2. Table 4.20 below shows the percentage of the correct number of clusters estimated by model-based clustering using the BIC, maximum ASW criterion.

Table 4.20 PPR for model-based clustering using BIC, ASW and OSil.

Methods	DGPs									
	1	2	3	4	5	6	7	8	9	10
BIC	100	100	60	44	4	100	0	0	0	60
ASW	80	52	12	96	0	0	0	100	0	56
OSil	84	36	12	100	0	0	0	100	4	56

OSil has shown 100% performance as compared to BIC (44%) for Model 4, which contains non-Gaussian clusters. A similar result is true for model-based clustering using ASW and OSil initialized with model-based clustering for Model 8, where BIC was never able to estimate correct number of clusters.

For Model 6, OSil has estimated 6 clusters a majority of the times, except for model-based clustering with which it has estimated 5 cluster solution. For model-based clustering it is hard to separate the two small clusters very close to each other as two components.

4.6.4 Some general comments

The purpose of the experiments was not only to compare the performance of the proposed method with the existing competitors, but also to find out whether the idea of OASW clustering is worthwhile at all for the estimation of number of clusters in comparison to the ASW. Another motivation for setting up this experiment is to systematically investigate the behaviour of the existing estimation indices as well. [Milligan \(1981\)](#) conducted a study to evaluate the clusterings obtained from 4 hierarchical clustering methods using 30 internal criterion. The clustering methods used were single link, complete link, group average, and Ward's minimum variance. The data model considered had a strong clustering structure and 4 clusters. The clusters were compact and well separated. The indices included in the study was chosen from the proposed indices during the period of 1967 and 1980. In another study, [Milligan and Cooper \(1985\)](#) has conducted an experiment with 4 artificially generated data sets having 2 to 5 number of clusters. 30 methods to estimate number of clusters were considered with four hierarchical clustering methods mentioned previously. The indices included in the study was chosen from the proposed indices proposed during the period of 1965 to 1983. Recently, [Arbelaitz et al. \(2013\)](#) have conducted the cluster validation study using the 30 indices proposed during the period of 1973 to 2011. For many indices, their several versions were included to compare their performance. The experiment was conducted to estimate the numbers of clusters from each index. They considered three clustering methods, namely average linkage, Ward's method and k -means. The artificial data sets were generated considering five factors, that were number of clusters (allowed values were 2, 4, and 8), dimensions, cluster overlap, cluster density, and noise level. They also considered 20 real data sets ranging from 2 to 15 clusters.

The strength of the experiments done here is that they include a larger number of clustering methods for evaluation with indices covering wide spectrum of methods, and artificial data sets having various clustering structures than those used in previous such studies. The Gap, PS, BI, and CVNN indices never appeared in a comparative study together with other indices in such extensive systematic simulations. The most promising and widely used indices were chosen for the comparative study here. These widely used indices are paired with various fundamentally different clustering methodologies that are in use across disciplines. It has been observed that researchers mostly tend to show the good performances of these indices for a few data sets by pairing them with a clustering method with which they show good performance, and do not discuss at all how these indices will perform in other situations.

Authors have shown a way too rigid focus on ideas and methods for clustering or estimation indices which will perform well on easy challenges or toy data and fail to propose indices which can perform reasonably well for even fairly general situations. Real data sets are complex and demanding. A single data set may contain some clusters that are compact as well as others that have wide variations. The data sets here are designed with the focus that several of the clustering challenges were present in one data set. We have evaluated the performance of these indices for very complex data structures for clustering solutions as well as for the estimation of number of clusters.

[Van der Laan et al. \(2003\)](#) has shown that PAMSIL is good for detecting small sized clusters in presence of bigger sized clusters, where the clusters have same covariances. By small sized clusters they mean number of observations in clusters. They have set an experiment (this is Model 10) where 6 clusters have 25 observations and one cluster has 350 observations. The clusters only differ in mean and are equally distant from each other. The clusters also have the same co-variance matrix. However, we think that this is a too simple clustering challenge and quite often in practice, clusters will not only differ in shapes, and number of observations, they will be non-Gaussian as well as unequally distant. Therefore, we have included PAMSIL for its in-depth analysis based on much diverse data sets generated from Gaussian and other distributions. We have not only included small clusters, as just defined but also other concepts related to small clusters i.e., different spread in clusters and also from different distributions.

4.7 Simulation III: Overlapping data structure

In this section we perform an experimental design study for understanding the behaviour of OSil algorithm. In experimental design, the experimental conditions are called factors and the output performance is called the response. The goal of the experimental design for the simulations was to see how each factor (experimental condition) will affect the response (for instance the value of the ASW) in a systematic manner. For this study we have multiple response variables of interest. We want to judge how each factor will affect these response variables. One purpose of the study is also to identify which of these factors are more important than others and have greater effect on the response. The factors are based on major concerns that can contribute to the final clustering solution. There are various important issues here, which we are interested in investigating. Which initialization method will give the best ASW value? How good is the developed technique in discovering/fitting the simulated data generating structure? How will the sizes and relative sizes of the clusters, the spread of the clusters and clusters coming from different distributions affect the value of ASW and the discovered clustering structure? Evaluation of run time complexity of the algorithm as factor effecting time will grow such as n and k . How many iterations the algorithm will take to converge? This is to observe the increase in number of iterations by increasing the

observations or clusters.

Suppose that there are m factors each at v levels and m' factors at w levels. We have used a $v^m \times w^{m'}$ factorial experiment. We now first list the factors included and their corresponding levels in the experiment below. Each of these are discussed in more detail right after we list them below, together with the details of how to exactly generate the data.

- (i) Number of observations (Levels: $n_1 = 225, n_2 = 625$).
- (ii) Number of dimensions (Levels: $p_1 = 2, p_2 = 30, p_3 = 200$)
- (iii) Number of clusters (Levels: $k_1 = 2, k_2 = 5, k_3 = 10$)
- (iv) relative sizes of clusters (Levels: s_1 (equal): all clusters are of equal sizes. s_2 (one small): one cluster contains 15% of the observations and rest are equally sized. s_3 (one big): one cluster contains 70% observations and the remaining observations are equally sized among other clusters.)
- (v) Cluster separation (distance between means of clusters) (Levels: Ω_1 : overlapping clusters (0.01), Ω_2 : close clusters (0.1), Ω_3 : well separated clusters (0.3)).
- (vi) Covariance structures among clusters. Levels: ζ_1 (small variations): all clusters have equal covariance matrix. ζ_2 (big variations): one cluster has smaller covariance matrix and ζ_3 (mixed variation): one cluster has smaller covariance matrix along dimensions.

The number of observations in each cluster depends upon the levels of factors 1, 3 and 4 above. The corresponding sample size calculation for the design considered here is given Table 4.21. Let $\mu_{1k_1} = (0.5, 1)$, $\mu_{2k_1} = (0, -1)$, $\mu_{1k_2} = (0.5, 1, 3.5, 5, 6.5)$, $\mu_{2k_2} = (0, -1, 0.5, -0.5, 1)$, $\mu_{1k_3} = (0.5, 1, 3.5, 5, 6.5, -0.5, -2, -3.5, -5, -6.5)$, and $\mu_{2k_3} = (0, -1, 0.5, -0.5, 1, 0, -1, 0.5, -0.5, 1)$. For k_1 the cluster means were generated as $(\mu_{1k_1} + \Omega_1)$ in the first dimension and $(\mu_{2k_1} + \Omega_1)$ in the second dimension. For k_2 the cluster means were generated as $(\mu_{1k_2} + \Omega_1)$ in the first dimension and $(\mu_{2k_2} + \Omega_1)$ across all the remaining dimensions. Finally for k_3 the cluster means were generated as $(\mu_{1k_3} + \Omega_1)$ in the first dimension and $(\mu_{2k_3} + \Omega_1)$ across all the remaining dimensions.

Let O_p and I_p represents the null and identity matrices of order p . The size of these matrices will depend upon the dimensions of the data. The covariance matrix for the data sets depends upon the levels of covariance structures, number of dimensions and number of clusters. Lets define the following matrices first:

$$\Sigma_1 = \begin{bmatrix} \Gamma_1 & | & O \\ \hline O & | & I \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} \Gamma_2 & | & O \\ \hline O & | & I \end{bmatrix}, \text{ and } \Sigma_3 = \begin{bmatrix} \Gamma_3 & | & O \\ \hline O & | & I \end{bmatrix},$$

Table 4.21 Sample size calculation in each cluster against 3 levels of number of clusters and relative sizes of clusters considered in the simulation study. Use $n = n_1 = 225$ and $n = n_2 = 625$ for calculations.

No. of Clusters	Relative size of clusters	No. of observations in each cluster
$k_1 = 2$	$s_1 = \text{equal}$	$\frac{n}{2}$
	$s_2 = 1 \text{ small}$	$15\%n, 85\%n$
	$s_3 = 1 \text{ big}$	$70\%n, 30\%n$
$k_2 = 5$	$s_1 = \text{equal}$	$\frac{n}{5}$
	$s_2 = 1 \text{ small}$	$15\%n;$ $\frac{85\%n}{4} (\text{remaining each})$
	$s_3 = 1 \text{ big}$	$70\%n;$ $\frac{30\%n}{4} (\text{remaining each})$
$k_3 = 10$	$s_1 = \text{equal}$	$\frac{n}{10}$
	$s_1 = 1 \text{ small}$	$15\%n;$ $\frac{85\%n}{9} (\text{remaining each})$
	$s_3 = 1 \text{ big}$	$70\%n;$ $\frac{30\%n}{9} (\text{remaining each})$

where

$$\Gamma_1 = \begin{bmatrix} 0.3 \\ 0.1 & 0.3 \\ 0.1 & 0.1 & 0.3 \\ 0.1 & 0.1 & 0.1 & 0.3 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.3 \end{bmatrix}, \quad \Gamma_2 = \begin{bmatrix} 0.7 \\ 0.5 & 0.7 \\ 0.5 & 0.5 & 0.7 \\ 0.5 & 0.5 & 0.5 & 0.7 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.7 \end{bmatrix},$$

and

$$\Gamma_3 = \begin{bmatrix} 0.3 \\ 0.1 & 0.2 \\ 0.1 & 0.2 & 0.3 \\ 0.1 & 0.2 & 0.3 & 0.4 \\ 0.1 & 0.2 & 0.3 & 0.4 & 0.5 \end{bmatrix}.$$

For p_1 only the first 2×2 dimensions of Σ_1 , Σ_2 and Σ_3 were used. For p_2 the dimensions of O and I were O_{25} , I_{25} and for d_3 , O_{195} , I_{195} .

The covariance structures in clusters differ in two ways. Firstly there are basic covariance matrix structures. These are:

- Σ_1 : small observational spread (across the first five dimensions),

- Σ_2 : bigger observational spread (across the first five dimensions),
- Σ_3 : Mixed variations among observations (across the first five dimensions).

Secondly, to achieve these variations observational spreads, not all the clusters were generated with the same covariance matrices:

- Equal covariance matrix: all clusters were generated using small observational spread, i.e., Σ_1
- One cluster has smaller variation than the others' that have bigger observational spread, i.e., all clusters were generated from Σ_2 except one from Σ_1
- One cluster has smaller variation than the others' that have mixed variations among observations , i.e., all clusters were generated from Σ_3 except one from Σ_1

For k_1 and k_2 (number of clusters), the three covariance structures ζ_1 , ζ_2 and ζ_3 were achieved in the same fashion as defined above. To allow other variational patterns like negative correlation within cluster dimensions, more observational variation was added for k_3 . For k_3 (10-clusters) and ζ_1 (small variation among observations within-clusters) all clusters were generated from Σ_1 . For k_3 and ζ_2 nine clusters were generated from Σ_2 and one was generated from Σ_1 . For k_3 and ζ_3 one cluster each was generated from Σ_1 , Σ_4 , Σ_5 , Σ_6 , Σ_7 and the remaining clusters from Σ_3 , where the new Σ 's are defined as under.

$$\Sigma_i = \begin{bmatrix} \Gamma_i & | & O \\ O & | & I \end{bmatrix}, \quad i = 4, 5, 6, 7,$$

where

$$\Gamma_4 = \begin{bmatrix} 0.01 \\ 0.001 & 0.01 \\ 0.001 & 0.001 & 0.01 \\ 0.001 & 0.001 & 0.001 & 0.01 \\ 0.001 & 0.001 & 0.001 & 0.001 & 0.01 \end{bmatrix}, \Gamma_5 = \begin{bmatrix} 0.2 \\ -0.05 & 0.2 \\ -0.05 & -0.05 & 0.2 \\ -0.05 & -0.05 & -0.05 & 0.2 \\ -0.05 & -0.05 & -0.05 & -0.05 & 0.2 \end{bmatrix},$$

$$\Gamma_6 = \begin{bmatrix} 0.5 \\ -0.09 & 0.5 \\ -0.09 & -0.09 & 0.5 \\ -0.09 & -0.09 & -0.09 & 0.5 \\ -0.09 & -0.09 & -0.09 & -0.09 & 0.5 \end{bmatrix}, \Gamma_7 = \begin{bmatrix} 0.3 \\ 0.09 & 0.3 \\ 0.09 & 0.09 & 0.3 \\ 0.09 & 0.09 & 0.09 & 0.3 \\ 0.09 & 0.09 & 0.09 & 0.09 & 0.3 \end{bmatrix}.$$

A complete balanced factorial experiment gives: number of observations (2) \times Number of variables (3) \times number of clusters (3) \times size of clusters (3) \times cluster separation (3) \times covariance structures (3) = $3^5 2^1 = 486$ unique data conditions. $B = 3$ data sets of each type were generated, resulting in 1458 data sets. The clusters are generated independently from the Gaussian distribution. Each of the generated data sets was initialized with seven initialization methods (k -means:with 100 random initialization, PAM, average linkage, Ward's similarity, spectral, model-based). ASW, adjusted rand index (ARI) and standard error (SE) were calculated for each clustering obtained. Each of these seven initial clusterings were then passed to the OSil algorithm to find the ASW. We look at the ASW, SE and ARIs for these as well. The setup is run for a fixed/known number of clusters.

Performance evaluation

To examine the performance of each initialization method 1) we look at the ASW for the initialization and the ASW value obtained from OSil together with their standard errors. Note that as the ASW is a clustering quality index to compare different clustering methods, therefore ASW can also be used here to examine the best fitted solution among all considered methods. 2) For comparison of how good the resulting clustering matches the data generating process, we have calculated the adjusted rand index (ARI). 3) For each case in our study we have computed the time taken by the methods and, number of iterations taken by the algorithm to converge.

4.7.1 Results discussion

Some data sets are plotted in Figures 4.29 to give the impression of what kind of clustering structures are to be identified by the clustering methods in this setup.

Table 4.22 (at the end of this section) represents the ASW values obtained against various clustering methods and ASW when initialized with these clustering methods together with their standard errors. Each cell represents an average counts of $1458/3=486$ data sets except for the number of observations case which has $1458/2=729$.

The best ASW value among the clustering methods is obtained through the average linkage hierarchical clustering method. OSil with all the initialization methods have always improved the value of the ASW obtained from the existing clustering methods. Among these improved values the bold values in Table 4.22 represent the highest values of the ASW obtained. The best values for the ASW were always obtained from the average linkage agglomerative hierarchical clustering initialization passed to OSil. Note that PAMSIL never gave a best value of ASW.

Among the cluster separation factor, ASW was maximised for the well-separated clusters case. Among different co-variance structure factor, the ASW value was maximised for the mixed variation case. As the number of observations increases, the ASW value decreases for all the initialization methods except for model-based clustering, whereas the ASW value decreases only for some initialization methods namely

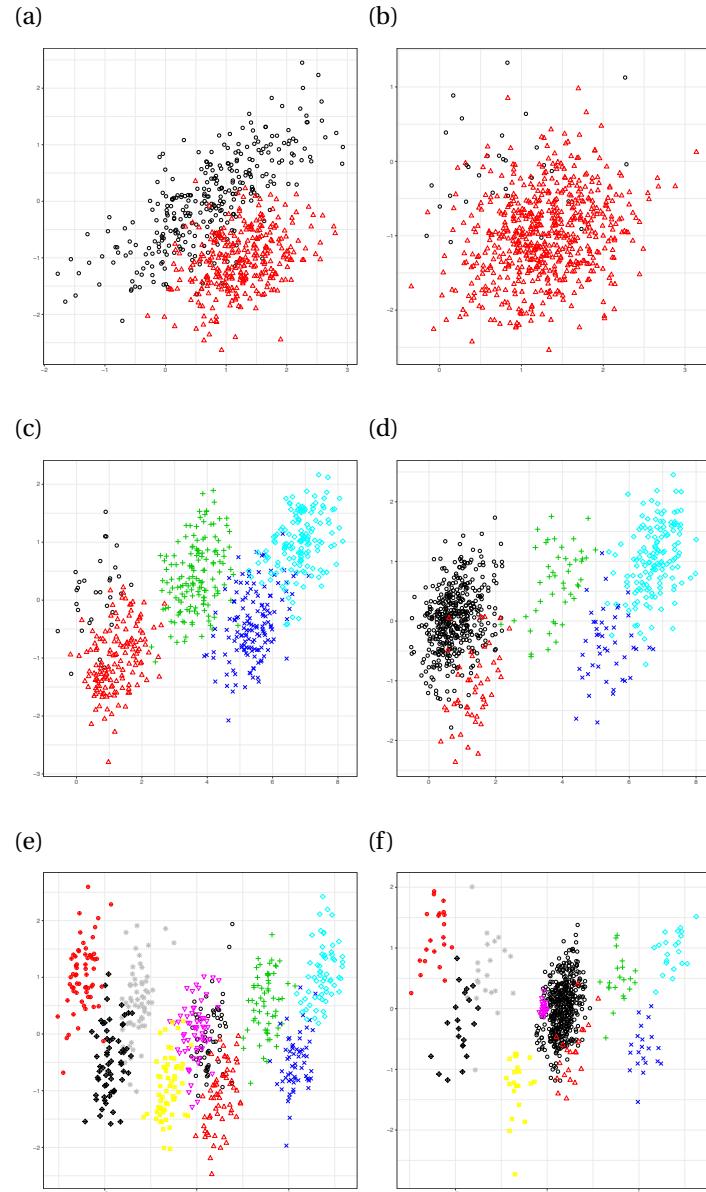


Figure 4.29 Few data condition plots from experimental design study. The factors and their levels are: (a) $p_3 = 200$ (number of variables, shown are the first two dimensions); Ω_1 :overlapping (separation level); $n_2 = 625$ (number of observations); $k_1 = 2$ (number of clusters); s_1 :equal clusters (relative cluster sizes); ζ_2 :big variations (covariance structure). $p_3 = 200$; Ω_1 :overlapping; $n_2 = 625$ in all figures. (b) $k_1 = 2$; s_2 :one small; ζ_1 : small variations. (c) $k_2 = 5$; s_2 :one small cluster; ζ_1 : small variations. (d) $k_2 = 5$; s_3 :one big cluster; ζ_1 : small variations. (e) $k_3 = 10$; s_1 :equal clusters; ζ_1 : small variations. (f) $k_3 = 10$; s_3 :one big cluster; ζ_3 : mixed variations.

k -means and Ward's method. As the number of clusters increases, the ASW value decreases, except for the model-based clustering initialization, where the maximum occurs in the order k_2 , k_1 , and k_3 . Furthermore, OSil initialization by Ward, spectral, and model-based clustering methods never gave the best ASW values.

On the other hand the ARI value for the OSil clustering are never better than for the existing clustering methods. Table 4.23 represents the ARI values obtained for experimental conditions and initialization methods in Table 4.22. The bold values represent the maximum ARI for the existing clustering methods and the OASW clustering schemes. The maximum ARI for the existing clustering methods was attained for k -means clustering method except for two data conditions (k_3 and s_1) where average linkage was predominant.

The best ARI values obtained were never from OSil or PAMSIL, indicating that optimizing ASW might not be a good method for the clustering solution itself for this data structure. The best ARI for OSil clustering was obtained from the Ward's method for agglomerative hierarchical clustering initialization except for two data conditions (k_3 and s_1) where average linkage initialization took the lead.

OSil initialized with PAM (italic values in Table 4.23) gave the better ARI values as compared to the PAM clustering for all the data conditions. Note that PAMSIL has never given a better value of the ARI than OSil initialized with PAM. For k_2 , k_3 and p_1 , OSil initialized with spectral clustering have gave higher ARI values as compared to ARI obtained from just the spectral clustering method.

An important thing to note here is that the best values of ARI are obtained for the clustering method (which is k -means) that has not given the best ASW value for most of the data conditions. For instance, for number of clusters as two the best value of ARI (0.2441) is obtained from the k -means clustering method, whereas the ASW obtained with k -means clustering is not the highest (0.0824). The best ASW value (0.0903) was obtained from the average linkage clustering method for the number of clusters as two.

In the summary, we have learnt that the existing clustering method that gave the best value of the ASW also gave the best value of the ASW when used for initialisation for OSil. The existing clustering method that gave the best value of ASW don't gives the best value of ARI. The proposed clustering method never gave a better value of the ARI as compared to the existing methods for this data structure. This indicates that it's not necessary that a clustering method that gave the best value of ASW will give the best ARI value as well.

Table 4.24 represents the time taken by the OSil algorithm run with each initialization. Each cell is calculated from an average of 162 data sets except for the number of observations factor of the design which have just two levels hence the average is for 243 data sets. The OSil algorithm is computationally more expensive than existing clustering methods including PAMSIL. The time reported here is in seconds and also includes the initialization time. As expected, as the number of clusters increases or the num-

ber of observations increases or the number of dimensions increases the computation time also increases. The OSil initialization with the model-based clustering method, spectral clustering method, PAM algorithm and Ward's clustering methods consumed almost equal time. These are the methods who took the highest time. k -means is the second best in terms of time, whereas average linkage takes the minimal time among these methods.

In conclusion OSil clustering with average linkage initialization performed best in terms of achieving the best ASW value and have the smallest computational time, whereas the highest ARI value for OSil was achieved from Ward's initialization method, keeping in mind that this highest ARI is always lower than the highest ARI value achieved from the clustering methods that gave maximum ASW.

Table 4.22: ASW and its standard error (SE) for the initialization methods and OSil with its SE for the factorial experiment design.

Factors	Initialization Methods													
	k-means		PAM		Average		Ward		Spectral		Model-based			PAMSIL
	init SE	OSil SE	init SE	OSil SE	init SE	OSil SE	init SE	OSil SE	init SE	OSil SE	init SE	OSil SE	init SE	OSil SE
Overall	0.2159 0.0044	0.2378 0.0078	0.1931 0.0060	0.2422 0.0091	0.2288 0.0126	0.2831 0.0095	0.2069 0.0057	0.2466 0.0108	0.1955 0.0169	0.2619 0.0129	0.1934 0.0143	0.2606 0.0144	0.2607 0.0111	
Clusters														
2	0.2471 0.0069	0.2653 0.0087	0.2108 0.0074	0.2658 0.0081	0.2710 0.0157	0.3078 0.0117	0.2410 0.0071	0.2683 0.0102	0.2447 0.0166	0.2924 0.0145	0.2186 0.0119	0.2540 0.0151	0.2936 0.0127	
5	0.2110 0.0033	0.2430 0.0084	0.1948 0.0061	0.2506 0.0116	0.2175 0.0125	0.2885 0.0055	0.2022 0.0101	0.2514 0.0108	0.1965 0.0172	0.2757 0.0130	0.2043 0.0128	0.2762 0.0140	0.2693 0.0088	
10	0.1896 0.0032	0.2051 0.0062	0.1737 0.0046	0.2103 0.0046	0.1978 0.0076	0.2530 0.0094	0.1774 0.0066	0.2202 0.0045	0.1452 0.0113	0.2174 0.0169	0.1574 0.0111	0.2515 0.0183	0.2193 0.0140	
Dimensions														
2	0.2767 0.0078	0.2989 0.0101	0.2568 0.0082	0.2994 0.0099	0.2932 0.0166	0.3487 0.0121	0.2665 0.0084	0.3031 0.0117	0.2449 0.0234	0.3240 0.0164	0.2575 0.0187	0.3242 0.0181	0.3229 0.0144	
30	0.2567 0.0037	0.2710 0.0080	0.2418 0.0062	0.2737 0.0096	0.2683 0.0147	0.3291 0.0091	0.2429 0.0066	0.2825 0.0145	0.2378 0.0217	0.3121 0.0151	0.2327 0.0133	0.2976 0.0167	0.3093 0.0125	
200	0.1143 0.0018	0.1435 0.0053	0.0806 0.0037	0.1537 0.0079	0.1249 0.0063	0.1715 0.0072	0.1113 0.0021	0.1543 0.0062	0.1038 0.0056	0.1495 0.0071	0.0900 0.0109	0.1600 0.0084	0.1500 0.0064	
Observations														
225	0.2167 0.0045	0.2382 0.0075	0.1937 0.0065	0.2422 0.0090	0.2294 0.0124	0.2826 0.0100	0.2074 0.0059	0.2468 0.0107	0.1961 0.0170	0.2603 0.0131	0.1933 0.0142	0.2598 0.0139	0.2606 0.0113	
625	0.2151 0.0044	0.2374 0.0081	0.1925 0.0056	0.2423 0.0093	0.2282 0.0128	0.2837 0.0090	0.2064 0.0055	0.2465 0.0109	0.1948 0.0168	0.2635 0.0127	0.1935 0.0144	0.2614 0.0149	0.2609 0.0110	
Cluster Size														
Equal	0.2237 0.0036	0.2365 0.0070	0.2139 0.0054	0.2427 0.0086	0.2242 0.0109	0.2614 0.0077	0.2133 0.0049	0.2384 0.0075	0.2056 0.0151	0.2524 0.0100	0.1878 0.0125	0.2340 0.0097	0.2523 0.0061	
One small	0.2226 0.0060	0.2460 0.0076	0.1778 0.0067	0.2535 0.0098	0.2349 0.0120	0.2824 0.0112	0.2170 0.0060	0.2495 0.0095	0.1978 0.0179	0.2654 0.0138	0.2098 0.0150	0.2719 0.0168	0.2718 0.0096	
One big	0.2014 0.0038	0.2309 0.0087	0.1875 0.0060	0.2306 0.0091	0.2272 0.0147	0.3055 0.0096	0.1903 0.0062	0.2519 0.0154	0.1831 0.0176	0.2679 0.0149	0.1827 0.0154	0.2758 0.0167	0.2581 0.0177	
Separation														
Well-separated	0.2175 0.0045	0.2385 0.0070	0.1944 0.0062	0.2435 0.0076	0.2312 0.0124	0.2834 0.0094	0.2087 0.0058	0.2467 0.0095	0.1908 0.0185	0.2585 0.0131	0.1937 0.0146	0.2590 0.0150	0.2592 0.0103	
close	0.2163 0.0041	0.2383 0.0077	0.1930 0.0064	0.2426 0.0096	0.2280 0.0124	0.2826 0.0097	0.2066 0.0056	0.2480 0.0119	0.1983 0.0159	0.2616 0.0128	0.1928 0.0144	0.2622 0.0135	0.2610 0.0115	
over-lapping	0.2139 0.0048	0.2366 0.0086	0.1918 0.0054	0.2406 0.0101	0.2272 0.0128	0.2833 0.0094	0.2053 0.0057	0.2452 0.0110	0.1973 0.0162	0.2655 0.0127	0.1938 0.0139	0.2605 0.0146	0.2619 0.0116	
Co-Variance Structure														
small equal variation	0.2158 0.0045	0.2351 0.0068	0.1943 0.0057	0.2381 0.0075	0.2282 0.0134	0.2838 0.0088	0.2073 0.0055	0.2439 0.0109	0.1968 0.0161	0.2614 0.0126	0.1980 0.0136	0.2597 0.0152	0.2556 0.0113	
big unequal variation	0.2099 0.0041	0.2359 0.0086	0.1844 0.0061	0.2404 0.0096	0.2197 0.0118	0.2739 0.0111	0.1989 0.0058	0.2455 0.0110	0.1879 0.0176	0.2564 0.0132	0.1813 0.0157	0.2531 0.0143	0.2614 0.0103	
mixed variation	0.2221 0.0047	0.2424 0.0079	0.2005 0.0062	0.2483 0.0103	0.2385 0.0125	0.2917 0.0086	0.2144 0.0086	0.2504 0.0058	0.2018 0.0104	0.2678 0.0169	0.2010 0.0128	0.2690 0.0136	0.2651 0.0118	

Table 4.23: ARI for initialization methods and OSil clustering obtained for factorial experiment design.

Factors	Initialization Methods													
	k-means		PAM		Average		Ward		Spectral		Model-based		PAMSIL	
	init	OSil	init	OSil	init	OSil	init	OSil	init	OSil	init	OSil	ARI	
Overall	0.7214	0.6323	0.5322	0.6001	0.6700	0.5893	0.7125	0.6520	0.6219	0.5953	0.5846	0.5072	0.5814	
Clusters														
2	0.7324	0.6045	0.5247	0.6051	0.5867	0.4703	0.7287	0.6055	0.6493	0.5632	0.5019	0.3389	0.6000	
5	0.8287	0.7608	0.6025	0.6970	0.7294	0.6761	0.8226	0.7694	0.7096	0.7148	0.6952	0.6609	0.5968	
10	0.6030	0.5316	0.4695	0.4983	0.6940	0.6215	0.5861	0.5811	0.5067	0.5078	0.5565	0.5219	0.5474	
Dimensions														
2	0.6269	0.5783	0.4991	0.5616	0.5905	0.5591	0.6258	0.5907	0.5704	0.5738	0.5611	0.5257	0.5718	
30	0.6815	0.6248	0.5369	0.5903	0.6768	0.6102	0.6684	0.6449	0.5870	0.5726	0.5798	0.4975	0.5610	
200	0.8557	0.6937	0.5607	0.6485	0.7428	0.5986	0.8432	0.7204	0.7082	0.6393	0.6127	0.4985	0.6113	
Observations														
225	0.7248	0.6311	0.5352	0.6025	0.6721	0.5929	0.7131	0.6513	0.6253	0.5991	0.5862	0.5185	0.5750	
625	0.7179	0.6334	0.5293	0.5978	0.6680	0.5857	0.7118	0.6527	0.6184	0.5915	0.5829	0.4959	0.5878	
Cluster Size														
Equal	0.8270	0.7739	0.7070	0.7351	0.6252	0.6196	0.7813	0.7753	0.7040	0.6811	0.5358	0.5104	0.6364	
One small	0.7689	0.5890	0.4614	0.5403	0.6620	0.5314	0.7575	0.5885	0.6507	0.5501	0.6328	0.4913	0.5387	
One big	0.5682	0.5339	0.4283	0.5249	0.7229	0.6169	0.5985	0.5922	0.5109	0.5545	0.5850	0.5200	0.5691	
Separation														
Well-separated	0.7281	0.6325	0.5317	0.6000	0.6701	0.5765	0.7143	0.6472	0.6251	0.5953	0.5779	0.5141	0.5843	
close	0.7266	0.6332	0.5360	0.6005	0.6674	0.5888	0.7119	0.6532	0.6227	0.5947	0.5806	0.5025	0.5774	
over-lapping	0.7094	0.6312	0.5290	0.5999	0.6725	0.6027	0.7112	0.6556	0.6177	0.5958	0.5951	0.5051	0.5825	
Co-Variance Structure														
small equal variation	0.7448	0.6638	0.5580	0.6278	0.6941	0.6141	0.7326	0.6826	0.6446	0.6189	0.6006	0.5273	0.5948	
big unequal variation	0.6613	0.5650	0.4743	0.5422	0.6267	0.5377	0.6584	0.5874	0.5811	0.5452	0.5485	0.4550	0.5426	
mixed variation	0.7580	0.6681	0.5645	0.6303	0.6893	0.6161	0.7464	0.6860	0.6400	0.6217	0.6045	0.5394	0.6067	

Table 4.24: Time (in seconds) taken by the OSil algorithm including initialization time in the factorial experiment design setup.

Factors	Initialization Methods						
	<i>k</i> -means	PAM	Average	Ward	Spectral	Model-based	PAMSIL
Clusters							
2	7.82	20.08	2.95	14.28	8.60	17.85	1.54
5	114.15	243.59	84.93	153.07	221.23	162.19	13.07
10	327.93	501.96	158.52	489.56	455.21	451.47	36.86
Dimensions							
2	3.12	5.63	1.96	4.21	5.39	6.85	1.06
30	69.93	86.79	89.14	189.02	161.60	172.45	21.20
200	376.84	673.21	155.30	463.68	518.06	452.21	29.21
Observations							
225	142.96	251.82	78.34	205.30	216.34	198.91	16.15
625	156.96	258.60	85.93	232.65	240.35	222.10	18.16
Cluster Sizes							
Equal	136.14	195.59	60.00	139.47	180.10	194.81	13.76
One small	123.41	212.01	50.56	136.68	170.73	160.80	13.56
One big	190.34	358.04	135.84	380.76	334.21	275.90	24.16
Separation							
Well-separated	122.93	220.65	51.13	133.42	160.32	142.56	10.02
close	163.18	263.02	91.35	234.02	246.17	215.93	19.22
over-lapping	163.78	281.97	103.91	289.48	278.54	273.02	22.24
Co-variance structure							
small equal variation	128.36	238.59	83.50	190.19	204.61	188.13	16.96
big unequal variations	198.65	290.30	94.48	273.83	278.69	245.16	18.61
mixed variations	122.87	236.74	68.42	192.90	201.74	198.22	15.90

4.8 Runtime complexity

The runtime complexity of OSil is high. The formal complexity is done for the final algorithm. For a given initial clustering, each observation $i \in C_r$, $r = 1, \dots, k$, is shifted to every other cluster. Hence there are $n \times (k - 1)$ possibilities to run at each iteration till convergence. The practical runtime of the algorithm will depend on the number of iterations required to optimize ASW. Table 4.25 represents the runtime for a few combinations of the number of clusters and the number of observations in the data. The data was generated from the Gaussian distribution. In the beginning two clusters were generated having equal number of observations centred at $(0,0)$ and $(0,1)$. For other values of k clusters were added one by one. The total number of observations were always divided equally among clusters. The highest value in the table is corresponding to $n = 1000$ and $k = 10$. For this case the data is shown in Figure 4.30.

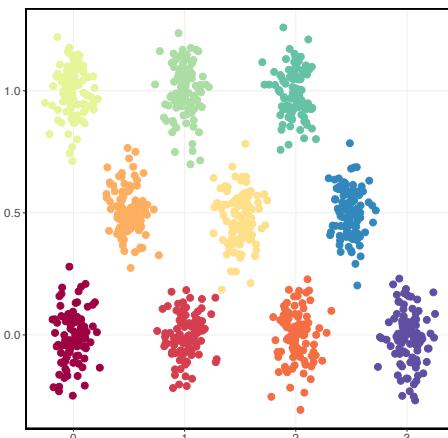


Figure 4.30 Data set generated for $n = 1000$ and $k = 10$ for runtime evaluation.

The runtime reported in the tables are in seconds and are based on just a single data sets. It is quite possible that in other runs the algorithm does not take that long and converges much faster than this, or can take even longer based on the number of iterations. Also from the reported values for runtime for DGPs considered in Simulation I and II in the tables, it is evident that the algorithm is taking longer than other methods. This is an indication that the algorithm is slow and alternative ways for speeding it up are necessary to explore.

Table 4.25 Time taken by *OSil* algorithm for various combinations of n and k .

k	n	100	200	300	400	500	600	700	800	900	1000
2	0.024	0.151	4.872	2.544	10.65	29.63	2094	118	28.04	380	
3	0.049	0.614	0.750	1.817	4.36	83.94	64.86	72	293	293	
4	0.113	1.002	8.795	17.89	29.53	195	182	618	294	1948	
5	0.081	1.049	14.93	18.77	82.44	114	462	519	581	2428	
6	0.081	1.266	3.486	36.54	55.94	200	148	400	486	948	
7	0.076	2.126	22.53	22.95	160	224	572	211	579	2142	
8	0.109	1.258	12.98	51.67	61.19	233	200	386	1386	1924	
9	0.160	1.971	8.478	25.33	183	218	319	1554	1257	2596	
10	0.219	1.857	4.346	51.08	72.12	279	220	1232	1556	4321	

4.9 Best OASW algorithm selection

In this section we will present the final version of OSil algorithm based on the best initialization methods. From Simulation I, II, and III we have identified the best clustering methods for the initialization based on their performance to optimize the ASW. We initialize the algorithm every time with each of these best performing initialization methods and use only one of these based on maximum ASW to initialize the OASW clustering algorithm. The algorithm is named as $OSil_1$. The $OSil_1$ algorithm can estimate the numbers of clusters as well. The algorithm starts with $k = 2$ clusters and finds out the clusterings from several clustering methods listed below, where we present the algorithm formally. It then chooses one best initialization based on the maximum ASW value to start performing OASW clustering. The next step is the PAM style swapping phase to find out the best possible value of ASW. The algorithm then repeats these steps to estimate the number of clusters.

Let the dataset denoted by \mathcal{X} of size n is to be clustered into k clusters. Let the number of clusters be estimated from the range $2, \dots, K$. Recall all notations from section 2.6. In addition a few more notations are needed. Let $l_j(\mathcal{X}, k) = (l_j(1), \dots, l_j(n))$ represents the clustering labels for $j \in \{1, 2, 3, 4, 5\}$, where each value of ‘j’ represents a clustering method namely, k -means, PAM, average linkage, Ward’s method, and model-based clustering, respectively. We now present the $OSil_1$ algorithm.

OSil₁ algorithm

Set number of clusters $k = 2$ and maximum number of clusters K .

Initialize

- (i) Calculate $d(x_i, x_h), \forall i \neq h \in \mathbb{N}_n$.
- (ii) Calculate the clustering using k -means, PAM, average linkage, Ward's method and model-based clustering and initialize the five clustering label vectors obtained from these methods with k clusters as $l_j(\mathcal{X}, k) = (l_j(1), \dots, l_j(n)), j \in \{1, \dots, 5\}$, respectively.
- (iii) Calculate $f^{(0)} = \arg \max_j f(l_j(\mathcal{X}, k), d)$, $j = \{1, \dots, 5\}$ as defined in (4.3).
- (iv) Set $q = 1$.

Swap

- (i) For all pairs (i, r) such that $x_i \notin C_r$, for $r \in \mathbb{N}_k$, assign $x_i \in C_r$ and denote the new label set as $l_{(i,r)}^*(\mathcal{X}, k) = (l^*(1), \dots, l^*(n))$.
- (ii) Compute $f_{(i,r)} = f(l_{(i,r)}^*(\mathcal{X}, k), d)$.
- (iii) $(h, s) = \arg \max_{(i,r)} f_{(i,r)}$, $f^{(q)} = f_{(h,s)}$, $l^{(q)}(\mathcal{X}, k) = l_{(h,s)}^*(\mathcal{X}, k)$.

Stop

If $f^{(q)} \leq f^{(q-1)}$. Else $q = q + 1$. Repeat *Swap*: Step (i)-(iii).

Repeat

- (i) Assign $f_{(k)} = f^{(q)}, l_{(k)}(\mathcal{X}, k) = l^{(q)}(\mathcal{X}, k)$. $k = k + 1$.

Repeat all steps from **Initialize**-(ii) to **Repeat** until $k = K$.

Estimate k

$$\hat{k} = \arg \max_{k=2, \dots, K} f_{(k)}$$

Return

$\hat{k}, f_{(\hat{k})}$ and $l_{(\hat{k})}(\mathcal{X}, k)$.

Recall that one of the results of the Simulation I (fixed k case) was an indication that it is not necessary that an initialisation methods which gave the maximum ASW value will only give the maximum ASW value for OSil clustering. In fact, we have observed for model 6 and 7 that an initialization with a lesser value of ASW can also give maximum ASW value for OSil. Based on this finding we now present another version of OSil algorithm named as OSil₂. This version of the algorithm does not decide a best initialization of the data based on ASW but instead it takes all of these 5 initializations to the swapping phase and produces 5 different OASW clusterings. One best clustering out of these 5 OASW clusterings is chosen at the end based on the maximum ASW value. It then repeats this whole process to estimate the best number of clusters. We

now formally present this algorithm.

OSil₂ Algorithm

Set number of clusters $k = 2$ and maximum number of clusters as K . Set $q = 1$.

Initialize

- (i) Calculate $d(x_i, x_h), \forall i \neq h \in \mathbb{N}_n$.
- (ii) Calculate the clustering using k -means, PAM, average linkage, Ward's method and model-based clustering and initialize the five clustering label sets with k clusters as $l_j(\mathcal{X}, k) = (l_j(i), \dots, l_j(i)), j = \{1, \dots, 5\}$, respectively.

Swap

- (i) For all pairs (i, r) such that $x_i \notin C_r$, for $r \in \mathbb{N}_k$, assign $x_i \in C_r$ and denote the new label vector as $l_{j,(i,r)}^*(\mathcal{X}, k) = (l_j^*(1), \dots, l_j^*(n))$, for all j .
- (ii) Compute $f_{j,(i,r)} = f(l_{j,(i,r)}^*(\mathcal{X}, k), d)$.
- (iii) $j, (h, s) = \arg \max_{(i,r)} f_{j,(i,r)}$, $f_j^{(q)} = f_{j,(h,s)}$, $l_j^{(q)}(\mathcal{X}, k) = l_{j,(i,r)}^*(\mathcal{X}, k)$.
- (iv) $f^{(q)} = \arg \max_j f_j^{(q)}$

Stop

If $f^{(q)} \leq f^{(q-1)}$. Else $q = q + 1$. Repeat *Swapping*: Step (i)-(iv).

Repeat

- (i) Assign $f_{(k)} = f^{(q)}, l_{(k)}(\mathcal{X}, k) = l^{(q)}(\mathcal{X}, k)$. $k = k + 1$.

Repeat all steps in **Initialize**-(ii) up to **Repeat** until $k = K$.

Estimate k

$$\hat{k} = \arg \max_{k=2, \dots, K} f_{(k)}$$

Return

$\hat{k}, f_{(\hat{k})}$ and $l_{(\hat{k})}(\mathcal{X}, k)$.

We now compare OSil₁ and OSil₂ performance. We considered first 4 DGPs defined in Subsection 4.4.1. All the setup is same as before, i.e., we consider 25 runs. The number of clusters k , were estimated from the range $1, \dots, K$, where we set $K = 12$. Let the estimated number of clusters be denoted by \hat{k} . The results for Model 1 to 4 are presented in Table 4.26 for the two proposed versions.

The column named “ $\hat{k} = k$ count” in Table 4.26 represents the count for the correct estimated k by OSil₁ and OSil₂. The time reported in the tables is in seconds and initialization time is included. For the true/known k both versions gives the same ASW, but for the estimation of k , OSil₂ gave a little higher value with a worse ARI value. Also OSil₁ shows a better performance in terms of the estimation of numbers of clusters.

First thing to note here is that OSil₂ has only given a slightly higher value of ASW

Table 4.26 OSil₁ and OSil₂ results comparisons for Models 1, 2, 3, and 4 for true k and estimated \hat{k} .

True k (Model 1)			True k (Model 2)			
Method	ASW	SE	ARI	ASW	SE	ARI
OSil ₁	0.6683	0.0055	0.8536	0.7098	0.0032	0.8571
OSil ₂	0.6683	0.0054	0.8579	0.7098	0.0032	0.8592
$\hat{k} = k$ (Model 1)						
Method	ASW	SE	ARI	time(s)	\hat{k} count	
OSil ₁	0.6701	0.005	0.8335	0.4452	23	
OSil ₂	0.6704	0.005	0.8213	4.2546	22	
\hat{k} (Model 2)						
Method	ASW	SE	ARI	time(s)	$\hat{k} = k$ count	
OSil ₁	0.7219	0.0022	0.8080	1.822	8	
OSil ₂	0.7227	0.0021	0.8056	17.906	8	
True k (Model 3)				True k (Model 4)		
Method	ASW	SE	ARI	ASW	SE	ARI
OSil ₁	0.6986	0.0031	0.9078	0.8354	0.002	0.9956
OSil ₂	0.6986	0.0031	0.9078	0.8354	0.002	0.9956
\hat{k} (Model 3)						
Method	ASW	SE	ARI	time(s)	$\hat{k} = k$ count	
OSil ₁	0.765	0.0027	0.3212	4.45	0	
OSil ₂	0.765	0.0027	0.3212	48.5923	0	
\hat{k} (Model 4)						
Method	ASW	SE	ARI	time(s)	$\hat{k} = k$ count	
OSil ₁	0.8354	0.002	0.9956	7.603	25	
OSil ₂	0.8354	0.002	0.9956	74.098	25	

twice. The gain in the ASW values obtained from it is very small (0.0002 for Model 1 and 0.0008 for Model 2), which is insignificant. Second, OSil₁ has shown the better performance in term of estimation of k only once (for Model 1). Third, note that ARI values, both OSil₁ and OSil₂ has shown larger values of ARI twice, rest being the same. For the data sets which have well-separated clusters like Model 4 both versions of the algorithm will give the same results. However, a clear advantage which OSil₁ has is its less computational time. From the table it is clear that OSil₁ is much faster than OSil₂. In general the results for OSil₁ and OSil₂ were consistent with each other and showed the same trend without any major difference.

As mentioned above there are four things here to consider before choosing one out of two versions, ASW value, estimation of k performance, ARI values, and runtime. The ASW is a clustering quality measure and the number of clusters can be estimated

from it using the maximum ASW value. So one should first look which algorithm is giving the best ASW value, and then accept whatever performance it is giving for the estimation of number of clusters for that maximum value. One could also argue that, it is not convincing to obtain a higher value of ASW with a bad clustering i.e., lesser ARI value. As the true clustering and number of clusters are not well-defined and in general debatable, algorithm selection based on ARI and the number of clusters is also debatable. In this work we are interested in investigating, what ASW do? Therefore we are concerned in obtaining best ASW values. Since the difference in ASW values obtained were not significant, we decide to proceed with OSil₁, provided that it is much more faster than other.

4.10 Fast version

As we have already observed that OSil is computationally expensive, so we worked on speeding up OSil₁. We have tried several ideas and compared them to identify one best fast version. In the following section, we first propose a fast version, and then briefly talk about its other legitimate variants that we tried. We then make the computational comparison between the proposed fast versions to decide one best. We then compare the performance of OSil₁ and its best fast version together in the section to follow.

4.10.1 FOSil algorithm

We take the approach of the random sampling to speed up the OSil₁ algorithm. This is a common and widely developed idea in literature. Clustering is first done only for a smaller subset of data or the objective function is optimized only using slices/sample of data, example includes, mini-batch algorithm to scale k-means to big data sets ([Baraldi and Blonda \(1999\)](#), [Sculley \(2010\)](#)). Other ideas of using the sampling approaches to speed up the algorithms appeared in [Guha et al. \(1998\)](#), [Zhang et al. \(1996\)](#), and [Karypis et al. \(1999\)](#).

The idea behind proposing a fast version of OSil₁ is not to run the OSil₁ algorithm on the entire data set directly, but to run it instead on a sample from the actual data. A random sample of size s is first drawn from the data and clustered by OSil₁, which we refer to as the partial clustering of the data. The final data clustering is obtained using OSil₁ partial clustering result. The remaining data points are then assigned to the partially clustered data based on the maximum ASW value. For each data point, k clusterings are defined by assigning each data point to all clusters. ASW is calculated for these k clusterings and point's membership is chosen based on maximum ASW. We call this clustering as FOSil₁ clustering, where 'F' stands for fast. We don't just take one random sample from the data set, but several of these to calculate the partial clustering

solution using OSil₁. One out of these partial clustering is chosen based on the best ASW value to perform full clustering.

The cluster memberships for all the points in the remaining data set are first decided one by one, and then assigned all together at once to the clusters to get the final clustering. The clusters were not updated each time separately for the remaining points. If we add the remaining points one by one to clusters, the silhouette widths and the overall ASW will be sensitive to the ordering of the remaining data. The clusters' average silhouette width for that cluster which got a new point will change every time. This will also change the silhouette width of other clusters for which this cluster was selected as nearest cluster due to the “min” in the definition of $b(i)$. This will cause the ASW of the entire clustering to change every time. In this scenario, the clustering will be sensitive to the ordering of the remaining data and with different ordering, one can get different clusterings.

We now recall some notation to present the FOSil algorithm. Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be the data set with n points and $\mathcal{C}_k = \{C_1, \dots, C_k\}$ is a k -clustering identified by some clustering function f_k on \mathcal{X} as usual. Let the clustering labels be $(l(1), \dots, l(n)) \in \mathbb{N}_k^n$ determined by $l(i) = r$, $r \in \mathbb{N}_k$, $i \in \mathbb{N}_n$. Let us define (2.12) for the five initialization methods separately as

$$\bar{S}_j(\mathcal{C}_k, d) = \frac{1}{n} \sum_{i=1}^n S_i(\mathcal{C}_{(k,j)}, d), \quad (4.5)$$

for $j = 1, 2, 2, 4, 5$, where each value of j represents k -means, PAM, average linkage, Ward's method, and model-based clustering methods, respectively, and $\mathcal{C}_{(k,j)}$ denote the clustering from each of the j methods.

Rewriting (4.5) for the label vector $l(\mathcal{X}, k)$ instead of clusterings set \mathcal{C}_k , to get the equivalent representation of (4.2) for j methods, gives:

$$\bar{S}_j(l_j(\mathcal{X}, k), d) = \frac{1}{n} \sum_{i=1}^n S_i(l_j(\mathcal{X}, k), d), \quad (4.6)$$

where $l_j(\mathcal{X}, k)$ is the clustering label vector for each j .

Let $\delta \in \mathbb{R}^+$ denote the proportion of the actual data to sample, such that the sample size(number of points) is $s = \delta \times n$. Further assume that the sampled data is denoted by S and remaining data by S' . Let the number of clusters k be to be estimated from the range 2 to K, where K is the maximum number of clusters allowed for estimation. Let the number of random samples of size s be denoted by $M \in \mathbb{N}$. Let m be an index for the M samples such that $m \in \{1, \dots, M\}$. Let $l'(S, k)$ denotes the clustering label vector for the best clustering selected from the five initialization clustering methods based on the maximum ASW for each m . Let $l^{(m)}(S, k)$ representation the clustering label vector corresponding to each sample M . Let $l''(S, k)$ represent the best clustering label vector

obtained from $l^{(m)}(S, k)$ from M samples.

The number of data points in S' will be $(n-s)$. Let $x'_h \in S'$, where $h = 1, \dots, (n-s)$ and $c(S', k) = (c(x'_1), \dots, c(x'_{(n-s)}))$, where $c(h) = r'$, $r' \in \{1, \dots, k\}$ and $h \in S'$ be the clustering label vector for S' . Let \mathcal{X}' represents the new ordering of the data set as $\mathcal{X}' = (S, S')$.

The numbers of clusters estimated from FOSil₁, is based on the sample data clustering using the ASW i.e., before performing the final data clustering. Before presetting the FOSil₁ algorithm formally, we present the steps of algorithm as follows:

- (i) Take a sample from the data. Let $m = 1$
- (ii) For this sample calculate the 5 clusterings to initialize.
- (iii) Choose one best out of these clusterings based on maximum ASW.
- (iv) Let $m = m + 1$ and repeat (ii) to (iv) until $m = M$.
- (v) Choose one best clusterings out of M clusterings based on maximum ASW. Call this as *initial clustering*.
- (vi) Pass the *initial clustering* to OSil₁.
- (vii) For $k = 2, \dots, K$, perform OSil₁ step and get one OSil₁ clusterings for each value of k . These will be $(K - 1)$, OSil₁ clusterings in total.
- (viii) Choose \hat{k} from $(K - 1)$ OSil₁ clusterings based on best ASW value. Call the clustering corresponding to \hat{k} as *partial clustering*.
- (ix) Assign remaining points to the *partial clustering* based on maximum ASW.

We experimented with several value of K , M and δ . We recommend $m = 25$. We tried 25, 50, 75, and 100 samples and found that $m = 25$ is good for the models included. We did not observe further improvement in results from other values of m . We observed that s between 2% to 20% of the the actual size of data gave good performance.

From experiments it was observed for data sets that have well separated clusters, that there will be no difference between ASW values obtained from OSil₁ and FOSil₁, and also that multiple sampling is not needed. Multiple sampling is good for the data sets that have overlapping, close clusters or widespread clusters.

There are several other ideas which we tried in the hope to further improve FOSil₁ clustering. For instance, to assign the remaining data to the partial clusters, we could decide their cluster memberships based on the maximum ASW value instead of the maximum ASW value. But in doing so, as the size of the sampled data s will approach the actual data size n , we will have the same computational complexity issue.

FOSil₁ algorithm

Choose K , M and δ . Set $k = 2$ and $m = 1$.

Sampling

- (i) Take a random samples of size s from \mathcal{X} . Let sample data be S and remaining data be S' .

Initialize

- (i) Calculate the pairwise dissimilarities $d(x_i, x_h)$, between all pairs of objects $(x_i, x_h) \in S$.
- (ii) Calculate the clustering of S using k -means, PAM, average linkage, Ward's method and model-based clustering, and initialize the five clustering label sets with k clusters as $l_j(S, k) = (l_j(1), \dots, l_j(s))$, $j \in \{1, \dots, 5\}$ for each of the five clustering methods, respectively.
- (iii) Calculate $j' = \operatorname{argmax}_j \bar{S}_j(l_j(S, k), d)$, $j \in \{1, \dots, 5\}$, where $\bar{S}_j(l_j(S, k), d)$ is defined in (4.6). Let $l'(S, k) = l_{j'}(S, k)$. Let $m = m + 1$.
- (iv) Calculate $f^{(m)} = \bar{S}(l'(S, k), d)$, where $\bar{S}(\cdot)$ is defined in (4.2). Assign $l^{(m)}(S, k) = l'(S, k)$.
- (v) Repeat **Sampling**-(i) and (iv) until $m = M$.
- (vi) $f^{(0)} = \max f^{(m)}$. Let $l''(S, k) = l^{(m)}(S, k)$ be the corresponding labels belonging to $f^{(0)}$.

Estimate \mathbf{k}

- (i) Calculate $f^{(k)} = f(l''(S, k), d)$, with $f(\cdot)$ as defined in 4.3. Let the resulting clustering be denoted by $\mathcal{C}_k = \{C_1, \dots, C_k\}$.
- (ii) $k = k + 1$, repeat all steps from **Sampling** up to now until $k = K$.
- (iii) $\hat{k} = \operatorname{argmax}_{k=2, \dots, K} f^{(k)}$.

Let the resulting clustering be denoted as $\mathcal{C}_{\hat{k}}$ be called **Partial Clustering**. Let $p(S, \hat{k}) = l''(S, k)$ be the clustering label vector belonging to $\mathcal{C}_{\hat{k}}$. Note that the full label vector is written as $p(S, \hat{k}) = (p(1), \dots, p(s))$.

Remaining Cluster Labels

To calculate the cluster membership for the points in S' using maximum ASW. Let $c^*(h)$ denotes a candidate label for a data point h in S' . Find the clustering label vector $c(S', k) = (c(x'_1), \dots, c(x'_{(n-s)}))$ for S' as:

- (i) For each pair (h, r') , where $h \in \{1, \dots, (n-s)\}$ and $r' \in \{1, \dots, \hat{k}\}$, assign $c^*(h) = r'$. Generate a label vector for $(s+1)$ points as $l_{(h, r')}^*((S, h), \hat{k}) = (p(1), \dots, p(s), c^*(h))$.
- (ii) Compute $f_{(h, r')} = f(l_{(h, r')}^*((S, h), \hat{k}), d)$, where $f(\cdot)$ defined in (2.12).
- (iii) $(h^*, r^*) = \operatorname{argmax}_{(h, r')} f_{(h, r')}$.
- (iv) Assign the label as $c(h^*) = r^*$.
- (v) Return $c(S', \hat{k}) = (c(x'_1), \dots, c(x'_{(n-s)}))$.

Final Clustering

- (i) Assign $\mathcal{X}' = (S, S')$ and $l(\mathcal{X}', \hat{k}) = (p(S, \hat{k}), c(S', \hat{k}))$.
- (ii) Calculate $f_{\mathcal{X}'} = f(l(\mathcal{X}', \hat{k}), d)$, with $f(\cdot)$ as defined in (4.3).

Return

$\hat{k}, f_{\mathcal{X}'}$ and $l(\mathcal{X}', \hat{k})$.

We tried to improve FOSil₁ clustering further by following two separate ideas. We describe them one by one. Considered the clustering obtained from FOSil₁ algorithm. Now from each cluster take a random sample of data points of size q , so that we have a sample of size $q \times k$ in total. For each of these sampled points change their cluster membership to other clusters to see if we get a further improvement in ASW values. We do this by calculating the ASW after performing swaps. A swap is good if ASW is higher for this clustering as compared to the ASW value we noted earlier. When tried out this idea on data sets, we have actually observed a decrease in ASW results. The reason is that if the sampled points were among those points that were located in the center or dense areas of the cluster, then the ASW is decreased. This idea could work better for the points on cluster edges because there is a high chance that they are misclassified. One reason for this could be that the clusters are updated at once at the end, and not one by one. If the clusters are updated point-wise, then ASW will be updated each time, and these points might be assigned to other clusters. Therefore, recalculating the ASW values after swapping all the points on the edges of clusters or doing so only after sampling from points on the edges, if there are many of these could further improve the ASW. However, we didn't try this.

It is advisable to keep s much smaller than n to keep the time complexity as low as possible, but the smaller size could work fine only for the data sets that have compact and well separated clusters, for the identification of correct clusters. For the data sets that have overlapping or wide and less dense clusters, the numbers of clusters may be estimated wrongly based on sample data, because in sample data the less dense region of a same clusters might appear as separate clusters rather than one cluster. To tackle this, some kind of re-evaluation for the estimated number of clusters from FOSil₁ based on the entire data set might help in identifying correct clusters. One idea is, once the FOSil₁ clustering is found, split each cluster into $(\psi \times k)$ sub-clusters, where ψ can be any natural number. For instance ψ between 2 and 5 is reasonable. The idea is to re-estimate the number of clusters k from the range 2 and $(\psi \times k)$. Calculate the ASW for this clustering. Next choose $q\%$ representative points from each of the $(k \times \psi)$ clusters. Only swap these representative points to other clusters and calculate ASW i.e., estimate k from $\underset{k \in \{2, \dots, (\psi \times k)\}}{\operatorname{argmax}} \bar{S}(\mathcal{C}_k, d)$ as defined in (2.12). However, we didn't try this idea.

In FOSil₁ we have estimated the number of clusters based on sample data only. FOSil₁ performance was poor for the estimation of k (results will be presented later). We now present another idea for FOSil clustering which we named as FOSil₂. FOSil₂ will estimate the number of clusters using the OASW clustering on the entire data set.

FOSil₂ performs clustering on the entire data set for 2 to K clusters. It stores clustering label vectors, the new data ordering and ASW values from 2 to K numbers of clusters. FOSil₁ estimates the number of clusters on the sample data only and needs much less storage space. Whereas, FOSil₂ estimates the number of clusters on the entire data and will take more computational time than FOSil₁. We now compare the two

FOSil₂ algorithm

Choose K , M and δ . Set $k = 2$ and $m = 1$.

Sampling

Take a random samples of size s from \mathcal{X} . Let sample data be S and remaining data be S' .

Initialize

- (i) Calculate the pairwise dissimilarities $d(x_i, x_h)$, between all pairs of objects $(x_i, x_h) \in S$.
- (ii) Calculate the clustering of S using k -means, PAM, average linkage, Ward's method and model-based clustering, and initialize the five clustering label sets with k clusters as $l_j(S, k) = (l_j(1), \dots, l_j(s))$, $j \in \{1, \dots, 5\}$ for each of the five clustering methods, respectively.
- (iii) Calculate $j' = \arg \max_j \bar{S}_j(l_j(S, k), d)$, $j \in \{1, \dots, 5\}$, where $\bar{S}_j(l_j(S, k), d)$ is defined in (4.6). Let $l'(S, k) = l_{j'}(S, k)$. Let $m = m + 1$.
- (iv) Calculate $f^{(m)} = \bar{S}(l'(S, k), d)$, where $\bar{S}(\cdot)$ is defined in (4.2). Assign $l^{(m)}(S, k) = l'(S, k)$.
- (v) Repeat all steps from **Sampling** till **Initialize**-(iv) until $m = M$.
- (vi) $f^{(0)} = \max f^{(m)}$. Let $l''(S, k) = l^{(m)}(S, k)$ be the labels belonging to $f^{(0)}$.

Partial clustering

- (i) Calculate $f^{(k)} = f(l''(S, k), d)$, with $f(\cdot)$ as defined in (4.3). Let the resulting clustering be denoted by $\mathcal{C}_k = \{C_1, \dots, C_k\}$. Note that the full label vector is written as $l''(S, \hat{k}) = (l''(1), \dots, l''(s))$.

Remaining Cluster Labels

To calculate the cluster membership for the points in S' using maximum ASW. Let $c^*(h)$ denotes a candidate label for a data point h in S' . Find the clustering label vector $c(S', k) = (c(x'_1), \dots, c(x'_{(n-s)}))$ for S' as:

- (i) For each pair (h, r') , where $h \in \{1, \dots, (n-s)\}$ and $r' \in \{1, \dots, k\}$, assign $c^*(h) = r'$. Generate a label vector for $(s+1)$ points as $l_{(h, r')}^*((S, h), k) = (l''(1), \dots, l''(s), c^*(h))$.
- (ii) Compute $f_{(h, r')} = f(l_{(h, r')}^*((S, h), k), d)$, where $f(\cdot)$ defined in (2.12).
- (iii) $(h^*, r^*) = \arg \max_{(h, r')} f_{(h, r')}$.
- (iv) Assign the label as $c(h^*) = r^*$.
- (v) Return $c(S', k) = (c(x'_1), \dots, c(x'_{(n-s)}))$.

Final Clustering

- (i) Assign $\mathcal{X}' = (S, S')$ and $l(\mathcal{X}', k) = (l''(S, \hat{k}), c(S', \hat{k}))$.
- (ii) Calculate $f^k = f(l(\mathcal{X}', k), d)$, with $f(\cdot)$ as defined in (4.3).
- (iii) $k = k + 1$, repeat all the steps from **Sampling** up to now until $k = K$.
- (iv) $\hat{k} = \arg \max_{k=2, \dots, K} f^{(k)}$.

Return

\hat{k} , $f^{(\hat{k})}$ and $l(\mathcal{X}', \hat{k})$.

versions of FOSil. Table 4.27 represents the results for the comparisons. For Model 1, Model 2, and Model 3 it is clear that FOSil₂ is performing much better than FOSil₁. However, the following table also shows if the clustering structure is clear, FOSil₁ is also able to estimate the correct number of clusters based on the sample data. For instance see the results for Model 4 in Table 4.27.

Table 4.27 FOSil₁ and FOSil₂ comparisons for Model 1, 2, 3, and 4, for true k and estimated k.

True k (Model 1)			True k (Model 2)			
Method	ASW	SE	ARI	ASW	SE	ARI
FOSil ₁	0.6685	0.0057	0.6623	0.709	0.0038	0.6296
FOSil ₂	0.668	0.0059	0.7942	0.7062	0.0067	0.6856
\hat{k} (Model 1)						
Method	ASW	SE	ARI	time(s)	\hat{k}	count
FOSil ₁	0.6306	0.0109	0.7227	2.5933	13	
FOSil ₂	0.6691	0.0054	0.8159	3.0303	23	
\hat{k} (Model 2)						
Method	ASW	SE	ARI	time(s)	\hat{k}	count
FOSil ₁	0.6991	0.0052	0.7572	1.6217	9	
FOSil ₂	0.716	0.003	0.8334	2.2963	14	
\hat{k} (Model 3)				\hat{k} (Model 4)		
Method	ASW	SE	ARI	ASW	SE	ARI
FOSil ₁	0.701	0.0033	0.4958	0.8206	0.0019	0.9174
FOSil ₂	0.7031	0.0027	0.5036	0.8207	0.0018	0.9888
\hat{k} (Model 3)						
Method	ASW	SE	ARI	time(s)	\hat{k}	count
FOSil ₁	0.7652	0.0041	0.3687	1.9187	1	
FOSil ₂	0.7693	0.0021	0.3203	2.9747	0	
\hat{k} (Model 4)						
Method	ASW	SE	ARI	time(s)	\hat{k}	count
FOSil ₁	0.81	0.0062	0.9728	2.1163	21	
FOSil ₂	0.8207	0.0018	0.9888	3.4095	25	

Finally, we apply two more ideas in order to seek improvement in the ASW value for FOSil₂. From Table 4.27 for Model 1 and 2, especially for the estimation of k case, we observed that FOSil₂ was giving slightly smaller values of the ASW than OSil₁, which is fine because FOSil₂ is an approximation, but we still tried to further improve this approximated value. First, in the “Initialize” phase of the algorithm one could proceed differently after step (iii) of the FOSil₂ algorithm. Step (iii) decides the best initial-

ization method based on the maximum value of the ASW. After choosing one best initialization from the five methods, this best initialization could be then passed to OSil₁. Thus the best initialization chosen from M samples is based on best ASW instead of just ASW. We call this FOSil₃ algorithm which is different than FOSil₂ just in the initialization phase. Note that all the steps are same, except **Initialize(iv)** is now calculated as follows:

FOSil₃ algorithm

- (iv) $f^{(m)} = f(l'(S, k), d)$, with $f(\cdot)$ as defined in (4.3).
-

The second idea is similar to the initialization as done in OSil₂. We have already observed from OSil₂ that passing all initialization to OSil to perform clustering is way more expensive than just passing one best initialization based on ASW. But doing so on the sample data will not incur the same expenses, and we can get a better ASW value than what we currently get from FOSil₂. We call the new algorithm formed by this as FOSil₄, which is basically passing all the 5 initializations to OSil₁ and then decide one best at the end based on maximum value of ASW for the sampled data. This algorithm is different from FOSil₂ in the **Initialize** phase only. After (ii) of the **Initialize** in FOSil₂, we pass all the five initializations to OSil₁ separately. We now present **Initialize** for FOSil₄ formally as follows:

FOSil₄ algorithm

Initialize

- (i) Calculate the pairwise dissimilarities $d(x_i, x_h)$, between all pairs of objects $(x_i, x_h) \in S$.
- (ii) Calculate the clustering of S using k -means, PAM, average linkage, Ward's method and model-based clustering, and initialize the five clustering label sets with k clusters as $l_j(S, k) = (l_j(1), \dots, l_j(s))$, $j \in \{1, \dots, 5\}$ for each of the five clustering methods, respectively.
- (iii) Calculate $f^{(j)} = f(l_j(\mathcal{X}, k), d)$, where $f(\cdot)$ as defined in (4.3). Let $l'_j(S, k)$ be the corresponding label vectors for $f^{(j)}$. Let $m = m + 1$.
- (iv) Calculate $f^{(m)} = \max f^{(j)}$, and $l^m(S, k)$ be the label set belonging to $f^{(m)}$.
- (v) Repeat all steps from **Sampling** up to now until $m = M$.
- (vi) $f^{(0)} = \max f^{(m)}$. Let $l''(S, k) = l^m(S, k)$ be the labels belonging to $f^{(0)}$.

Continue from the **Partial Clustering** step of FOSil₂.

In a nutshell, FOSil₁ differs from FOSil₂ in terms of estimation of number of clusters only. Since FOSil₁ performed poorly for the estimation of k , we decide to work further with FOSil₂. In an attempt to improve the performance further we introduces FOSil₃

and FOSil₄ that differ from FOSil₂ in the **Initialize** phase only. In the **Initialize** phase, first the five clustering methods are used to get the five clusterings. The remaining initialization steps for the three algorithms are summarized as follows. For **FOSil₂**

- (i) Choose one best out of 5 clusterings using maximum ASW
- (ii) Repeat this for M samples
- (iii) Choose one best out of M clusterings based on maximum ASW

For **FOSil₃**

- (i) Choose one best out of 5 clusterings using maximum ASW
- (ii) Pass this best to OSil₁
- (iii) Repeat this for M samples
- (iv) Choose one best out of M clusterings based on maximum ASW

For **FOSil₄**

- (i) Pass 5 clusterings to OSil₁
- (ii) Choose one best out of 5 OSil₁ clusterings based on maximum ASW
- (iii) Repeat this for M samples
- (iv) Choose one best out of M clusterings based on maximum ASW

We now compare the results for FOSil₂, FOSil₃ and FOSil₄. Table 4.28 represents the results for these models. From the Table 4.28 it is clear that FOSil₂ produces high values for ASW for the majority of models for fixed and estimated k. However, note that for a few cases FOSil₃ produces better values of the ARI and shows better estimates of the number of clusters for two models, but this is not a drastic improvement as compared to FOSil₂. Among these three faster approximations, FOSil₂ takes the least time and consistently produces the highest ASW values. FOSil₄ takes much more time than FOSil₂ and does not produce as high ASW values as FOSil₂.

Table 4.28 FOSil₂, FOSil₃ and FOSil₄ comparisons for Models 1 to 4.

Method	True k (Model 1)			True k (Model 2)		
	ASW	SE	ARI	ASW	SE	ARI
FOSil ₂	0.6636	0.0032	0.7454	0.6999	0.0056	0.6477
FOSil ₃	0.6634	0.0032	0.7827	0.7013	0.0049	0.6923
FOSil ₄	0.6636	0.0032	0.7125	0.7024	0.004	0.6379
\hat{k} (Model 1)						
Method	ASW	SE	ARI	time(s)	\hat{k} count	
	0.6641	0.0031	0.8182	3.5673	22	
FOSil ₃	0.6635	0.0032	0.8249	5.5071	23	
FOSil ₄	0.6641	0.0032	0.8001	20.1853	21	
\hat{k} (Model 2)						
Method	ASW	SE	ARI	time(s)	\hat{k} count	
	0.7112	0.0026	0.8116	2.4304	13	
FOSil ₃	0.7085	0.0023	0.8107	2.6666	16	
FOSil ₄	0.7111	0.0025	0.8238	4.0578	12	
True k (Model 3)						
Method	ASW	SE	ARI	ASW	SE	ARI
	0.7025	0.0028	0.4988	0.8203	0.002	0.9924
FOSil ₃	0.6993	0.0044	0.5013	0.8201	0.002	0.9924
FOSil ₄	0.6952	0.0064	0.4817	0.8189	0.0024	0.9702
\hat{k} (Model 3)						
Methods	ASW	SE	ARI	time(s)	\hat{k} count	
	0.7689	0.0023	0.3204	2.8454	0	
FOSil ₃	0.7687	0.0022	0.3221	3.2928	0	
FOSil ₄	0.7688	0.0022	0.3201	5.9061	0	
\hat{k} (Model 4)						
Methods	ASW	SE	ARI	time(s)	\hat{k} count	
	0.8205	0.002	0.996	8.4993	25	
FOSil ₂	0.8203	0.002	0.9924	3.2187	25	
FOSil ₃	0.8201	0.002	0.9924	3.956	25	
FOSil ₄	0.8189	0.0024	0.99	8.2414	24	

4.11 OSil and FOSil comparison

Based on our comparisons in the last two sections OSil₁ and FOSil₂ are better in optimizing the ASW. In this section we will compare the performance of the fast version to the regular version as well as with the existing clustering methods. Here we will present

results for the comparisons for DGPs defined in Subsection 4.4.1.

Results of the simulations for the Models 1-5 are in Tables 4.29 to 4.33. The time mentioned in the tables is in seconds. Since FOSil₂ is an approximation, it is not surprising that it gave ASW and ARI values smaller than OSil₁. In fact it has performed very close to OSil₁. PAMSIL has performed as good as OSil₁ in terms of ARI values, whereas OSil₁ takes less computation time and reaches higher values of the ASW as compared to PAMSIL.

Table 4.29 Comparison of OSil₁ and FOSil₂ with existing methods for Model 1.

Methods	True k			\hat{k}				
	ASW	SE	ARI	ASW	SE	ARI	time(s)	\hat{k} count
k-means	0.6693	0.0038	0.8434	0.6693	0.0038	0.8434	0.0031	25
PAM	0.6696	0.0037	0.8556	0.6696	0.0037	0.8556	0.0156	25
average	0.6059	0.0198	0.6385	0.6562	0.0048	0.8605	0.0011	18
Ward's	0.6626	0.0045	0.9191	0.6626	0.0045	0.9191	0.0012	25
model-based	0.6471	0.0052	0.9904	0.6477	0.0052	0.9763	0.218	24
spectral	0.6538	0.0088	0.95	0.6611	0.0043	0.9147	0.5788	24
BIC-mb	-	-	-	0.6444	0.0055	0.9889	0.1954	24
PAMSIL	0.6705	0.0036	0.8721	0.6707	0.0036	0.8656	0.6628	24
OSil ₁	0.6707	0.0036	0.8721	0.6707	0.0036	0.8656	0.4923	24
FOSil ₂	0.6644	0.0054	0.8272	0.6655	0.0047	0.844	1.8845	24

Table 4.30 Comparison of OSil₁ and FOSil₂ with existing methods for Model 2.

Methods	True k			\hat{k}				
	ASW	SE	ARI	ASW	SE	ARI	time(s)	\hat{k} count
k-means	0.7107	0.003	0.8417	0.7214	0.0028	0.8075	0.0037	12
PAM	0.7105	0.003	0.8521	0.7223	0.0023	0.8011	0.0266	11
average	0.6782	0.0117	0.8097	0.7142	0.0034	0.8204	0.0013	6
Ward's	0.6981	0.0038	0.9257	0.7111	0.0028	0.8444	0.0012	10
model-based	0.6782	0.0044	0.9936	0.6876	0.0039	0.9294	0.2977	12
spectral	0.6014	0.0363	0.8643	0.7093	0.0029	0.827	1.1047	6
BIC-mb	-	-	-	0.6782	0.0044	0.9936	0.2901	25
PAMSIL	0.7113	0.003	0.8539	0.7257	0.0021	0.792	1.8846	7
OSil ₁	0.7254	0.003	0.853	0.7254	0.0021	0.7952	1.6739	8
FOSil ₂	0.7076	0.0032	0.6389	0.7178	0.0027	0.8133	2.5061	12

Table 4.31 Comparison of OSil₁ and FOSil₂ with existing methods for Model 3.

Methods	True k			\hat{k}				
	ASW	SE	ARI	ASW	SE	ARI	time(s)	\hat{k} count
k-means	0.6734	0.0099	0.8194	0.7584	0.003	0.3171	0.0042	0
PAM	0.7028	0.0028	0.9146	0.7594	0.0026	0.3172	0.061	0
average	0.6424	0.0038	0.638	0.7574	0.0028	0.3238	0.0023	0
Ward's	0.6911	0.0041	0.9749	0.7572	0.0028	0.3264	0.0033	0
model-based	0.6786	0.0043	0.9984	0.748	0.0037	0.3298	0.2977	0
spectral	0.5177	0.0505	0.3276	0.7557	0.0028	0.8498	2.8976	0
BIC-mb	-	-	-	0.65	0.0106	0.9614	0.2608	14
PAMSIL	0.7037	0.0028	0.9185	0.7623	0.0024	0.3216	4.5714	0
OSil ₁	0.7623	0.0028	0.9174	0.7623	0.0024	0.3216	3.931	0
FOSil ₂	0.7012	0.0031	0.5031	0.761	0.0026	0.3195	3.0282	0

Table 4.32 Comparison of OSil₁ and FOSil₂ with existing methods for Model 4.

Methods	True k			\hat{k}				
	ASW	SE	ARI	ASW	SE	ARI	time(s)	\hat{k} count
k-means	0.6853	0.0275	0.8415	0.7702	0.0105	0.9349	0.0054	12
PAM	0.8194	0.002	0.9948	0.8194	0.002	0.9948	0.0818	25
average	0.8187	0.002	0.9944	0.8187	0.002	0.9944	0.0067	25
Ward's	0.8185	0.0021	0.9944	0.8185	0.0021	0.9944	0.004	25
model-based	0.8108	0.0034	0.9838	0.8108	0.0034	0.9838	0.3174	25
BIC-mb	-	-	-	0.7688	0.0107	0.9604	0.2461	12
PAMSIL	0.8195	0.002	0.9964	0.8195	0.002	0.9964	6.8964	25
OSil ₁	0.8195	0.002	0.9964	0.8195	0.002	0.9964	10.0383	25
FOSil ₂	0.819	0.0021	0.9952	0.819	0.0021	0.9952	3.4667	25

Table 4.33 Comparison of OSil₁ and FOSil₂ with existing methods for Model 5.

Methods	True k			\hat{k}				
	ASW	SE	ARI	ASW	SE	ARI	time(s)	\hat{k} count
k-means	0.6844	0.0128	0.828	0.7253	0.0044	0.7036	0.0061	11
PAM	0.7422	0.0024	0.9572	0.7437	0.0019	0.971	0.0986	20
average	0.5724	0.0065	0.3025	0.6978	0.0021	0.1908	0.0039	0
Ward's	0.7155	0.0019	0.7757	0.7232	0.0026	0.7614	0.0048	14
model-based	0.6934	0.0038	0.7795	0.7304	0.0023	0.8166	0.4022	1
BIC-mb	-	-	-	0.7279	0.0029	0.8164	0.3278	4
PAMSIL	0.7468	0.002	0.9872	0.7475	0.0018	0.989	8.9691	21
OSil ₁	0.7473	0.0024	0.9603	0.7473	0.0019	0.9819	9.6599	22
FOSil ₂	0.7453	0.0023	0.9714	0.7462	0.0021	0.9868	3.9418	24

For Model 6 and Model 7, FOSil₂ and OSil₁ produced same ASW value. This ASW value was also same as ASW values obtained from other clustering methods except for model-based and spectral clustering methods. All the methods failed for the estimation of number of clusters here. For Model 6 clustering methods estimated 4 number of

clusters not 5. For Model 7, the methods have estimated 3 number of clusters instead of 7.

For models 8, 9, and 10 OSil₁ and FOSil₂ gave exactly same results for ASW values and ARI. They have estimated correct number of clusters for these models as well as produced the correct clustering. The other clustering methods also gave the best ASW value equivalent to the ASW value except *k*-means and spectral clusterings methods for Model 8 and Model 9, and *k*-means, spectral and model-based clustering methods for model 10. For Model 8, all the methods have estimated correct number of clusters except for *k*-means and spectral clustering methods. For Model 9, all the methods have estimated correct number of clusters except for *k*-means clustering methods. For Model 10, all the methods have estimated correct number of clusters except the *k*-means, spectral and model-based clustering methods.

4.12 Closing remarks for simulations

Our purpose for setting the simulations was not to show the performance of the proposed algorithms for strong or clear structures, but rather to find out, how tough clustering challenges the proposed method can handle and how existing methods will perform in these situations. Because real life applications might not contain strong clustering structures and they are not as easy as the ones, one can set in simulations. There are several other well-separated, coherent existing data sets where OSil and FOSil will perform as good as existing methods.

This study is an exploration of a wide range of well-reputed clustering methods in practice. OSil has improved the performance of many clustering methods in two respects. The first is to find the correct clustering and the second is to estimate the desired number of clusters. The biggest benefit was observed regarding the estimation of number of clusters through McQuitty, complete and single linkage as observed in Simulation II.

We now report the clustering results of OSil₁ and FOSil₂ for the DGPs considered in Chapter 3. From 17 DGPs considered for simulations in Section 3.4, of Chapter 3, seven models were initially chosen for experiments in this chapter, plus we defined a few more DGPs. Note that these 7 models covered a wide range of clustering characteristics. Model 2 in Chapter 3 is similar to Model 1 in this chapter. Similarly, Model 4, Model 8, Model 10, Model 11, Model 16 and Model 17 of Chapter 3 are same as Model 2, Model 3, Model 4, Model 5, Model 8 and Model 9 of this chapter, respectively. Although, the seven DGPs' structures are same in both chapters, however, the parameters are not exactly the same. In fact, in this chapter these models were made more challenging. For instance, by bring the cluster's means further close to each other or by increasing the within clusters observational spread. The OSil₁ and FOSil₂ performs well for the parametric choices in this chapter for these 7 DGPs, since these choices made here

are more challenging, clearly OSil₁ and FOSil₂ will also perform well for the parametric choices in previous chapter.

We now report the clustering results of OSil₁ and FOSil₂ for the remaining DGPs of Chapter 3. We also consider PAMSIL algorithm. We run OSil₁, FOSil₂ and PAMSIL algorithms on one data sets generated for the DGPs to find out whether they will be able to produce the desired clustering or not.

Model 1 OSil₁, FOSil₂ and PAMSIL were able to estimate the correct number of clusters as well as produced the correct clusterings. This performance is similar to the HOSil performance observed earlier in Chapter 3.

Model 3, Model 5 OSil₁, FOSil₂ and PAMSIL didn't estimate three number of clusters but two for both models. They were also not able to produce the desired clustering results even for the fixed known number of clusters. This performance is not similar to HOSil, which estimated correct number of clusters for both models as well as produced correct clustering.

Model 6 OSil₁ and FOSil₂ estimate the correct number of clusters as well as produce the correct clustering for both fixed known k as well as for the estimated k. This performance is same as HOSil. However, PAMSIL estimates the correct number of clusters but returns the clustering for the fixed known k and for the estimated k with many misclassified points.

Model 9 OSil₁, FOSil₂ and PAMSIL were able to estimate the correct number of clusters as well as produced the correct clusterings. This performance is similar to HOSil.

Model 12 OSil₁ produced the correct clustering for the fixed known k, however, it estimated 5 number of clusters instead of 6. FOSil₂ and PAMSIL didn't produce the correct clustering for the fixed known k and didn't estimate the correct number of clusters. HOSil performed same as that to OSil₁.

Model 13 OSil₁ produced the correct clustering for the fixed known k, however, it estimated 5 number of clusters instead of 14. FOSIL₂ estimate 4 number of clusters and didn't produced the correct clustering even for the fixed known k cases. PAMSIL also fail in producing the correct clustering even for the fixed known k. It estimates the maximum number provided as the number of clusters. We try the estimation of number of clusters by PAMSIL using maximum number K as: 16, 20, and 24. For these three values PAMSIL estimates k as 16, 20, and 24. HOSil has performed best for this model. It not only estimate the correct number of clusters but also produce the desired clustering.

Model 14, Model 15 OSil₁, FOSil₂ and PAMSIL estimate correct number of clusters and produce the correct clustering for fixed known k and estimated k. This performance is similar to HOSil performance observed earlier in Chapter 3.

We now report the result for the DGPs considered in Section 3.6 of Chapter 3

Model 18, Model 19, Model 20 OSil₁, FOSil₂ and PAMSIL estimate correct number of clusters and produce the correct clustering for fixed known k and estimated k. This performance is aligned with HOSil performance.

Model 21 OSIL₁, FOSIL₂ and PAMSIL are not able to produce the correct clustering for the known fixed k case. They estimate 14 number of clusters. HOSil also performed in this fashion.

Four Shapes, Diamonds, Tetra OSIL₁, FOSIL₂ and PAMSIL estimated the correct number of clusters and correct clustering for both fixed known k as well estimated k. This performance is aligned with HOSil performance.

Smiley OSil₁ and FOSil₂ produce the correct clusters for the fixed k, however PAMSIL did not. The three methods estimated number of clusters as 6. This performance is aligned with HOSil performance.

Aggregation, Lsun The three methods fail here, both in terms of estimation of number of clusters as well as producing clustering for the fixed known number of clusters. This performance is aligned with HOSil performance.

4.13 OSil₁ complexity

OSil₁ takes as an input the data matrix of size $n \times p$, where n is number of points in p -dimensional space. The first step is to calculate the pair wise distance matrix between observations. There are $n(n - 1)/2$ unique entries in the proximity matrix, which gives the complexity as $O(pn(n - 1)/2)$. We begin calculating the complexity of the algorithm for fixed k case. The OSil₁ algorithm can be divided into two parts where the first part is comprised of the **Initilize** step and second part is comprised of **Swap, Stop, Repeat** steps. In the **Initilize** phase the five clusterings of the data sets are computed to initialize the algorithm. These 5 clusterings are k -means, PAM, average, Ward, Model-based clustering. The k -means algorithm by [Lloyd \(1982\)](#) or [Kaufman and Rousseeuw \(1987\)](#) has time complexity $O(nkpq)$, where k is number of clusters and q is the number of iteration for convergence ([Garey et al. \(1982\)](#), [Mahajan et al. \(2009\)](#), [Aloise et al. \(2009\)](#)). The PAM by [Kaufman and Rousseeuw \(1987\)](#) has the time

complexity of $O(k(n - k)^2 q)$ ([Schubert and Rousseeuw \(2018\)](#)). The hierarchical clustering algorithms has $O(n^3)$ time complexity ([Day and Edelsbrunner \(1984\)](#), [Firdaus and Uddin \(2015\)](#)). The EM algorithm for model-based clustering has the complexity $O(npq)$ ([Firdaus and Uddin \(2015\)](#)). The total complexity from the 5 initialization is $O(nkpq) + O(k(n - k)^2 q) + 2O(n^3) + O(npq)$. One best out of these 5 clusterings is chose to pass to the second part.

The second part of the OSil₁ algorithm is implemented using two functions named as *sil_lab_swap()* and *clustyanlys()*. They second function is invoke once the first has finished. The final complexity of the algorithm will be decided by adding the complexity of these two functions. We now give the expression for the complexity of each of these as follows. The function *sil_lab_swap()* calls another function named *sil_lab()*. This function further calls two functions named as *grab()* having $O(1)$ complexity, *hpsort()* is sorting of a vector of length n from smallest to largest having $O(n^2)$ (quadratic) complexity. Thus the complexity of *sil_lab()* is given as under: $L_1 = 3O(nk) + O(nk)O(n^2) + c_1O(n) + c_2O(1)$, where $c_1, c_2 \in \mathbb{N}$ are some constant number of operations having linear and constant time complexities respectively. L_1 can be simplified as $L_1 = O(nk) + O(n^3k) + O(n) + O(1) \Rightarrow O(n^3k)$.

The complexity of *sil_lab_swap()* is as under:

$L_2 = L_1 + 2O(n) + q \times [nkL_1 + O(n^2k) + O(n) + O(1)]$, where q is the number of iteration taken by the algorithm to converge.

The complexity of *clustyanlys()* is: $L_3 = O(k) + O(n) + 2O(nk)$.

The total complexity of the algorithm is $O(n(n-1)/2p) + O(nkpq) + O(k(n-k)^2 q) + 2O(n^3) + O(npq) + L_2 + L_3$. If in a program there are operations involved that have various O complexities, then a superior bound defines the final complexity. Solving this gives the complexity of the OSil₁ algorithm as $O(qn^4k^2)$ where q is the number of iterations, n is number of data points, and k are number of clusters.

This complexity is for the fixed k only. If number of clusters are also being estimated then the complexity raise to $O(qn^4k^2K)$, where K is the maximum number of clusters tried out.

4.14 FOSil₂ complexity

The FOSil₂ algorithm has **Sampling**, **Initialize**, **Partial clustering**, **Remaining Cluster Labels**, **Final Clustering** parts in its implementation. The most computationally expensive are the first three parts. We begin to calculate the complexity for the algorithm for fixed k first. The new part in FOSil₂ algorithm as compared to OSil₁ is **Sampling**. The computational complexity of the **Sampling** phase is as under: The random sampling is done without replacement using the “sample()” function in R. The sample of size $d < n$ is taken without replacement. This has the $O(d \log d)$ quasilinear complexity ([Walker \(1974\)](#), [Becker et al. \(1988\)](#)). The random sampling is not done only once

but several times. Let M represents the number of times the random sampling is done. This gives the **Sampling** complexity as $O(Md \log d)$. The **Initialize** phase is carried out in the same way as described for the OSil₁ in Section 4.13 with the only difference that the initial clustering is performed only for a reduced data size which is d . The complexity is $O(dkpq) + O(k(d - k)^2q) + 2O(d^3) + O(dpq)$. The **Partial clustering** call OSil₁ clustering. This has the complexity of $O(qd^4k^2)$. The **Remaining Cluster Labels** has complexity of $((n - d)^5k^2)$. The **Final Clustering** has the time complexity $O(nk)$ complexity. The overall complexity of FOSil algorithm is: $O(Md \log d) + O(dkpq) + O(k(d - k)^2q) + 2O(d^3) + O(dpq) + O(qd^4k^2) + ((n - d)^5k^2) + O(nk)$. This simplifies to $O(qd^4k^2)$. For the estimation of k this complexity becomes: $O(qd^4k^2K)$.

4.15 OASW clustering: Compact and well-separated clusters

By definition the average silhouette width (ASW) is an average of all the silhouette widths for the individual data points. This means that for each data point the bigger silhouette width is better. A large silhouette width can be obtained if $a(i)$ is as small as possible and $b(i)$ as large as possible. A small $a(i)$ means small within cluster distance resulting in compact clusters. A large $b(i)$ leads to large between cluster distances resulting clusters as separate as possible. The clustering methods based on optimization of ASW will try to find the best silhouette width values for all the data points i.e., it will try to achieve $a(i)$ as small as possible and $b(i)$ as large as possible. The resulting clustering based on the optimization of ASW criterion will produce *compact* and *well separated* clusters.

For the illustration consider a data set with 2 clusters generated from the Gaussian distribution parametrized as $\mu_1: (0, 0)$ and $\mu_2: (2, 2)$ with common covariance matrix I_2 . Each cluster contains 10 points. Figure 4.31 represents a data plotting against true cluster labels and OSil₁ clustering. Let x and y represents the first and second dimension respectively. The silhouette widths calculations for this data based on the true labels is given in Table 4.34. The ASW for the known data labels is 0.252. A few points have a negative silhouette width (column 5 of table) in the data. The silhouette width for these points can be improved by changing their membership to the neighbouring clusters. Column 6 in Table 4.34 present labels obtained from OSil₁. The ASW for these labels are 0.487. The optimum average silhouette width works for compact and well separated clusters.

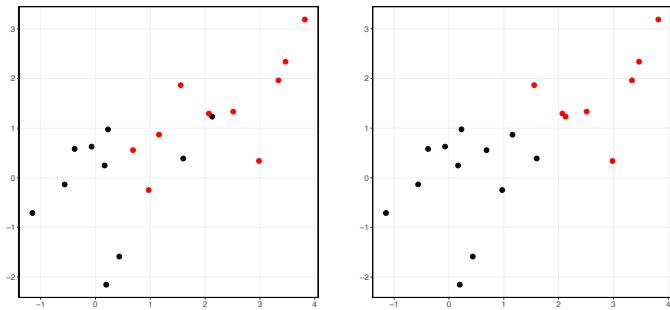


Figure 4.31 Left is the plot of data against true labels, and right is plot of data against a set of optimum labels obtained from $OSil_1$ clustering.

Table 4.34 SW calculations for true labels and a set of optimum labels returned from $OSil_1$ clustering algorithm.

x	y	True labels	neighbor	SW	$OSil_1$ labels	neighbor	SW
2.127	1.231	1	2	-0.519	2	1	0.500
-0.380	0.580	1	2	0.471	1	2	0.589
0.167	0.246	1	2	0.451	1	2	0.617
1.600	0.388	1	2	-0.185	1	2	0.043
0.196	-2.155	1	2	0.390	1	2	0.477
0.228	0.975	1	2	0.283	1	2	0.471
-1.149	-0.711	1	2	0.492	1	2	0.570
-0.071	0.628	1	2	0.442	1	2	0.586
-0.562	-0.136	1	2	0.528	1	2	0.625
0.434	-1.588	1	2	0.408	1	2	0.514
2.068	1.293	2	1	0.408	2	1	0.493
1.554	1.865	2	1	0.324	2	1	0.337
3.466	2.338	2	1	0.507	2	1	0.640
1.159	0.867	2	1	-0.025	1	2	0.145
0.971	-0.248	2	1	-0.364	1	2	0.480
3.818	3.188	2	1	0.436	2	1	0.551
0.685	0.556	2	1	-0.339	1	2	0.477
2.511	1.332	2	1	0.479	2	1	0.597
3.337	1.961	2	1	0.519	2	1	0.655
2.982	0.338	2	1	0.340	2	1	0.382

As the backbone of the optimization function presented in this work is the ASW, we now summarize the factors which will affect the ASW the most. The most important factors are, (i) the distances between the means of clusters (this affects $b(i)$), (ii) the spread of observation within the clusters (this affects $a(i)$) and (iii) the number of observations in clusters (this affects both $a(i)$ and $b(i)$ due to the involvement of the averages). For the explanation we will make use of a small data set with 2 clusters having 10 observations each generated from the Gaussian distribution with the two dimensional mean vectors as μ_1 : (0, 0) and μ_2 : (2, 2) respectively, with common covariance matrix as $0.25 \times I_2$. Let this be called Condition O. The following three operations were performed on condition O to generate other data conditions for comparison.

- (i) Condition A: Change the location of one cluster such that it comes closer to the other as compared to the original data. For this let $a = 1$ be a scalar added to all the values of cluster 1 such that the mean of the cluster becomes $\mu_1 + a$.
- (ii) Condition B: Change the spread of one cluster. For this let $b = 4$ be a scalar multiplied by all the observations in cluster 1. Note that this will automatically affect the mean of this cluster as well. As a result to this the mean of the cluster 1 will be $b \times \mu_1$ and co-variance matrix will be $b^2 \times 0.25 \times I_2$.
- (iii) Condition C: Change the spread of both clusters and bring cluster mean location closer. For this multiply all the observations in cluster 1 and 2 by the above defined scalar b . Define a new scalar quantity $a_1 = a - 2 \times b$. In addition to multiplying all observation of cluster 2 with b add a_1 to all the observations of the second cluster as well. This will locate the second cluster closer to first in terms of cluster means.

The graphical representation of these data are given in Figure 4.32.

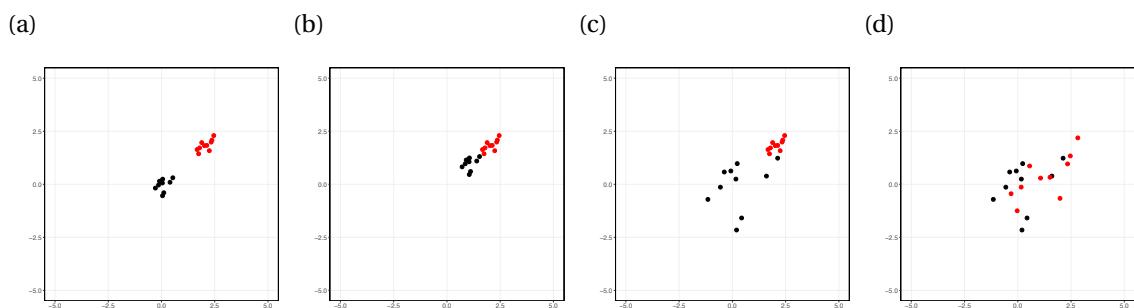


Figure 4.32 Panels (a)-(b) shows data plots using true data labels against Condition O, A, B and C respectively.

Table 4.35 SW for each data point for condition O, A and C.

Data	SW_O	SW_A	SW_B	SW_C	$SW_{O'}$
1	0.693	0.130	-0.746	-0.379	0.927
2	0.862	0.722	0.451	0.247	0.956
3	0.875	0.738	0.455	0.148	0.961
4	0.793	0.503	-0.231	-0.338	0.944
5	0.798	0.633	0.429	0.117	0.923
6	0.842	0.657	0.204	0.068	0.954
7	0.835	0.699	0.502	0.249	0.938
8	0.866	0.722	0.418	0.205	0.959
9	0.867	0.743	0.533	0.263	0.955
10	0.824	0.667	0.450	0.096	0.938
11	0.868	0.724	0.869	0.074	0.959
12	0.844	0.675	0.846	-0.122	0.951
13	0.839	0.707	0.838	0.262	0.939
14	0.821	0.581	0.825	-0.274	0.951
15	0.730	0.282	0.738	-0.291	0.934
16	0.793	0.642	0.792	0.212	0.914
17	0.769	0.414	0.776	-0.347	0.941
18	0.870	0.738	0.870	0.177	0.958
19	0.851	0.722	0.850	0.268	0.945
20	0.821	0.637	0.819	0.132	0.944

The ASW values using the true data labels for all of these conditions are given as follows:

Conditions	ASW
O	0.823
A	0.617
B	0.534
C	0.038

The silhouette width for each individual data point in the condition A, B and C has decreased as compared to condition O, meaning that $a(i)$ has increased and $b(i)$ has decreased under conditions A, B and C affecting the SW for these points negatively. The SW calculations for these data conditions are given in the Table 4.35. The last column for the table is referred later in the discussion.

A comparison of condition O with A also reveals that as the clusters move farther from each other, the ASW increases. To see another example, we define a condition O' by adding a constant $a' = 8$ to all the observations of cluster 1. For the resulting data the ASW with two clusters is 0.944. The SW for each point in this data is mentioned in the last column of Table 4.35.

This explains the reason for the HOSil estimating number of clusters as 2 for Model 6.A in Chapter 3. Even though ASW based optimization is capable of estimating the correct clustering here for the fixed true number of clusters, the maximization of the ASW will not give the desired estimate for the number of clusters. We recall the model definition here first for the explanation. Cluster 1 was generated from a Gaussian distribution with means $(1.5, 5)$ with covariance matrix $\Sigma = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.7 \end{bmatrix}$. Cluster 2 was generated from mean $(0, 5)$ with covariance matrix $0.5I_2$. Cluster 3 was generated from mean $(-1.5, 7)$ with co-variance matrix $0.1I_2$. Each cluster contains 50 observations. The ASW for the three cluster solution for a data set generated from this model is 0.6423. The ASW for two cluster solution, which is formed by putting cluster 1 and 2 together, is much higher (0.6863) than for the 3-cluster solution. This is due to the fact that the SW for many data points in cluster 3 will increase due to the bigger $b(i)$ for them (which is now based on all the data points in cluster 1 and 2) resulting in a bigger ASW for the 2-clusters solution. As the relative locations of the clusters matter for the ASW, moving cluster 1 further from cluster 2 (for instance using mean $(2.5, 5)$ instead of $(1.5, 5)$) will result in number of clusters estimate as 3 from ASW, HOSil, OSil and FOSil. Or one could bring cluster 3 closer to other two clusters, for instance (using mean $(0.5, 7)$ or as in Model 6 instead of $(-1.5, 7)$) will result in number of clusters estimate as 3. Similar reasons hold (involvement of the factors explained above) for HOSil estimating other than true number of clusters for Model 3, 8, 12 in Chapter 3 and OSil for Model 3, 6 and 7 for Chapter 4.

It is not necessary that the methods based on the optimization of the ASW will only find spherical clusters. As we have seen, they are capable of finding uniform clusters and moon like shapes (smiley data). Also the method has a capability of finding clusters that are not linearly separable at least for the fixed k case. The methods based on optimization of the ASW are capable of finding the clusters with different spread among the observations as well as with different number of observations in the clusters.

4.16 Distance metric comparison

The clustering results can be sensitive to the distance metric used. The choice of appropriate distance metric for a given data is important because with various metric used to cluster data the results can differ substantially ([Jain and Dubes \(1988b\)](#), [Jain](#)

et al. (1999), de Amorim and Komisarczuk (2012), Cordeiro De Amorim and Komisarczuk (2012), de Amorim and Hennig (2015)). The distance metric has an impact on the clustering algorithm's output and not all the metrics can handle all data structures. For instance Euclidean distance can capture the spherical and compact clusters present in the data but are not suitable for the complex or irregular shaped data sets (Newton et al. (1992)). Some metric are known for the data sets from a specific domain, for instance, Pearson's correlation coefficient, cosine angle distance or Spearman's rank-order correlation coefficient for the gene expression clustering (see Jiang et al. (2004)). Mimmack et al. (2001) conducted a study to analysis the effect of two distance metrics on climate data sets and concluded that the clustering of station data or grid points is highly sensitive to the distance metric used. This section is devoted to understand the influence of various clustering methods on the algorithms proposed in the current thesis.

4.16.1 Simulation scenario

We have done experiments with two different clustering structures. One of these structures have equal number of observations in clusters, the clusters have same within cluster variations, and the cluster' means are equally distant from each other. However, the other data structure is opposite to this. We now define the data structures and their results one by one below.

For the experiment we have first considered Model 7 defined in Chapter 3. The clustering structure is of such kind that the four clusters are equally distant from each other having equal number of observations. We have considered the three distance metrics namely Manhattan, Euclidean and Minkowski to observe the differences in the clustering results obtained by the proposed algorithms together with the existing methods. The Minkowski distance was run with the power 3. Each cluster contains 50 observations and 50 data sets were generated and clustering were calculated from k -means, PAM, average, Ward, model-based, spectral, PAMSIL, HOSil, OSil₁, and FOSil₂. The results are reported in Table 4.36. All the values reported in the table are for the estimated k .

4.16.2 Results

Overall, from all the methods the optimization performance gained from Minkowski metric is the highest. The ASW values obtained for all the methods showed same trend and the values obtained from the Minkowski metric were greater than the Euclidean metric and the values obtained from the Euclidean metric were greater than the Manhattan metric. The overall best ASW value obtained among all the clustering methods was from PAMSIL with Minkowski metric.

There is no clear trend for one distance metric in terms of the clustering performance. For different clustering methods different metrics gave the highest ARI values. The best ARI for PAM, Ward, model-based, PAMSIL and OSil₁ was obtained from Manhattan distance. However, the best ARI values for k -means, average and HOSil was obtained from Minkowski distance among the three metric. The overall best ARI among all clustering methods and three distance were achieved by PAMSIL with Manhattan distance. For the estimation of number of clusters all metric performed same. Spectral clustering has the lowest PPR among all clustering methods including all of the three metrics. It performed relatively better with the Euclidean distance as compared to other two.

However, these results are not generalisable as the performance of the distance metrics depend upon the clustering structures. This is evident from the results calculated for the Model 5 of Chapter 3. The simulation setup was same as described above except that the data structure now contains 3 clusters of unequal sizes and different within cluster variations. The clusters are also not equally distant from each other. For this model the largest clustering optimization values were obtained from Manhattan distance for all the methods except for k -means and model-based clusterings where the Euclidean metric outperforms. The best ASW value among all the clustering method was obtained from OSil₁ clustering with Manhattan distance. In terms of clustering performance Manhattan metric gave the largest ARI values among the three distance metrics for all the clustering methods always. The best ARI value among all the clustering methods was obtained from OSil₁ clustering using Manhattan distance. For the estimation of number of clusters Manhattan metric outperforms the other two and Minkowski metric performed the lowest among the three.

Table 4.36: Comparison of ASW values obtained from the distance metrics for various clustering methods.

	Manhattan				Euclidean				Minkowski (<i>power</i> =3)			
	ASW	SE	ARI	PPR	ASW	SE	ARI	PPR	ASW	SE	ARI	PPR
<i>Model 7: k=4, n=200, p=2, B=50</i>												
<i>k</i> -means	0.5747	0.0040	0.9588	90	0.6033	0.0022	0.9634	98	0.6124	0.0027	0.9642	98
PAM	0.5820	0.0023	0.9671	100	0.6042	0.0021	0.9605	100	0.6137	0.0023	0.9629	100
average	0.5756	0.0025	0.9405	100	0.5967	0.0026	0.9338	100	0.6064	0.0028	0.9415	100
Ward	0.5759	0.0025	0.9391	100	0.5962	0.0024	0.9313	100	0.6042	0.0033	0.9376	100
model-based	0.5823	0.0022	0.9671	100	0.6047	0.0021	0.9647	96	0.6140	0.0023	0.9661	96
spectral	0.5558	0.0063	0.8581	68	0.5778	0.0068	0.8669	76	0.5792	0.0078	0.8583	68
PAMSIL	0.5840	0.0020	0.9691	100	0.6076	0.0021	0.9686	100	0.6166	0.0022	0.9689	100
HOSil	0.5982	0.0044	0.9336	100	0.5745	0.0043	0.9286	98	0.6082	0.0040	0.9362	100
OSil ₁	0.5823	0.0022	0.9660	100	0.6047	0.0021	0.9653	100	0.6140	0.0023	0.9653	100
FOSil ₂	0.5820	0.0022	0.6822	100	0.6043	0.0021	0.6837	100	0.6138	0.0023	0.6829	100
<i>Model 5: k=3, n=150, p=2, B=50</i>												
<i>k</i> -means	0.6171	0.0057	0.8437	60	0.6192	0.0070	0.7837	36	0.6026	0.0072	0.7521	26
PAM	0.6217	0.0051	0.8737	72	0.6206	0.0068	0.8137	46	0.6047	0.0069	0.8044	42
average	0.6170	0.0046	0.9002	70	0.6103	0.0059	0.8486	46	0.5916	0.0062	0.8326	34
Ward	0.6180	0.0045	0.9262	78	0.6091	0.0057	0.8694	50	0.5917	0.0059	0.839	38
model-based	0.5999	0.0055	0.8628	42	0.6014	0.0058	0.8284	18	0.5855	0.0057	0.8378	28
spectral	0.6111	0.0054	0.9360	70	0.604	0.0058	0.8902	44	0.5875	0.0059	0.9004	38
PAMSIL	0.6141	0.0051	0.9032	76	0.5992	0.0062	0.8610	62	0.5958	0.0065	0.8388	52
HOSil	0.6043	0.0055	0.9066	74	0.5903	0.0057	0.8996	68	0.5903	0.0057	0.8996	68
OSil ₁	0.6237	0.0051	0.9730	72	0.6219	0.0067	0.9714	46	0.6060	0.0068	0.9398	36
FOSil ₂	0.6234	0.0051	0.3995	72	0.6211	0.0067	0.3527	46	0.6053	0.0068	0.3446	34

4.17 Applications

4.17.1 Tetragonula bee's data revisited

We now reconsider the Tetragonula bees data presented in Section 3.9.1 in Chapter 3. We have applied PAMSIL and OSil₁ algorithm to this data. For comparison we have applied *k*-means, PAM, average linkage, Ward's method, model-based clustering to estimate number of clusters using ASW. In addition we have estimated number of clusters from BIC using model-based clustering method. The results are presented in the Table 4.37. HOSil results from Section 3.9.1 are also recalled in the table.

Table 4.37 Clustering results for the bee's data.

Methods	True k		Estimated k		
	ASW	ARI	ASW	ARI	\hat{k}
True lab	0.4754				
k-means	0.2500	0.6359	0.2500	0.6359	9
PAM	0.1396	0.152	0.4131	0.6661	3
average	0.4754	1	0.4832	0.9515	10
Ward's	0.4708	0.8491	0.4711	0.8599	11
model-based	0.1234	0.5121	0.3003	0.3772	3
BIC-mb	-	-	0.1234	0.5121	9
PAMSIL	0.4715	0.9386	0.4839	0.9447	10
OSil ₁	0.4787	0.9994	0.4847	0.9440	10
HOSil	0.4800	0.91082	0.4841	0.9148	10

Average linkage, OSil₁ and PAMSIL has estimated 10 clusters with very close ARI values. *k*-means with ASW and model-based clustering with BIC have estimated number of clusters as 9, but with too low ARI values. For *k*=9, the best ARI values were achieved from average linkage, OSil₁ and PAMSIL respectively, with 0, 1, and 38 misclassified points, respectively. All the other methods gave very low ARI values for *k*=9. The data clustering from PAMSIL and OSil₁ for true *k* and estimated *k* is plotted in Figure 4.33 using 2-dimensional classical MDS plot. OSil₁ has performed better than HOSil here both in terms of clustering as well as significant reduction in time.

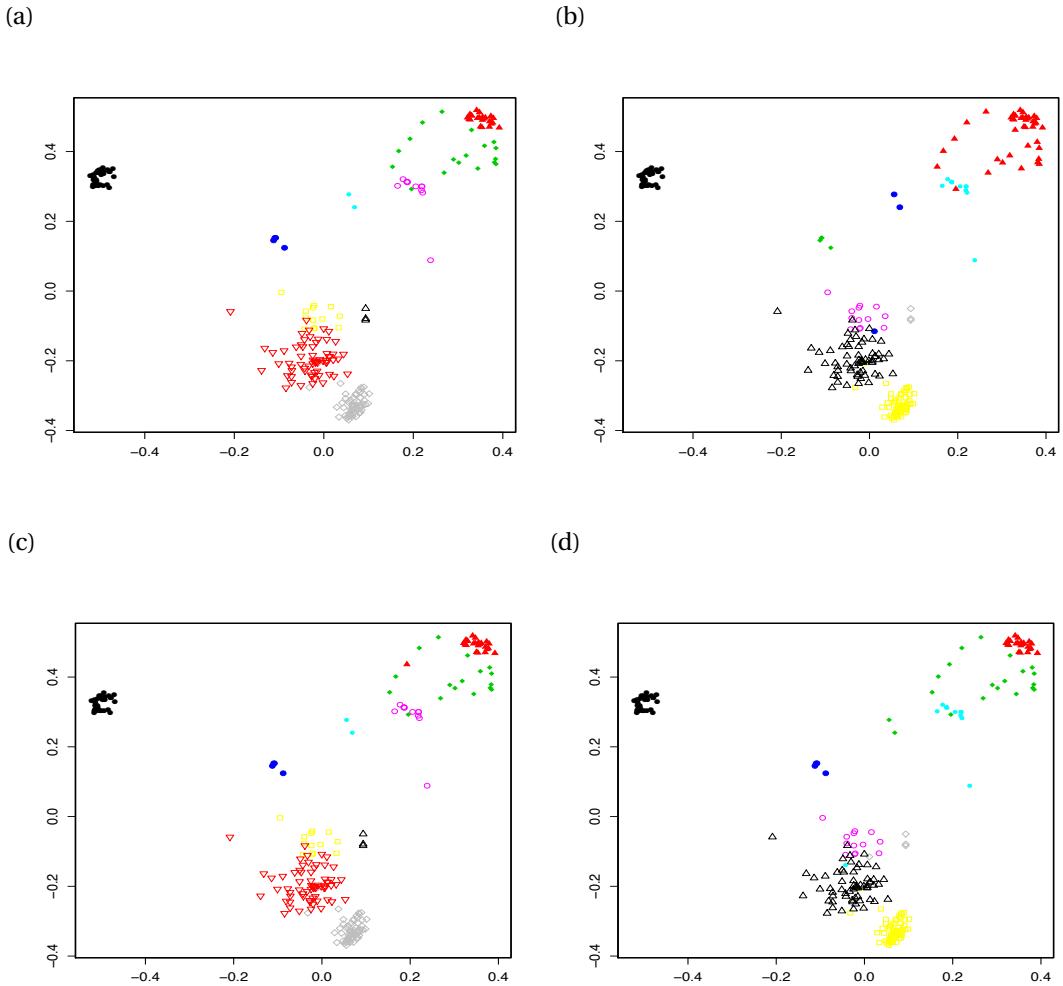


Figure 4.33 Classical MDS plot for the Bee's data. (a) OSil₁ clustering results for $k=10$, (b) OSil₁ clustering results for $k=9$, (c) PAMSIL clustering results for $k=10$, and (d) PAMSIL clustering results for $k=9$.

4.17.2 France rainfall data revisited

The OSil₁ clustering result for the French weather station data considered in Section 3.9.2 is given in the Figure 4.34. OSil₁ has put together Bastia and Perpignan together in one cluster and all other stations in other cluster for number of clusters $k=2$. For the number of clusters $k=3$, the north-east region is separated from the rest of the north. The north-east region with three mountain ranges of Aedennes, Vosges and Jura was separated from the north-west region. Bastia in the north-east of the Corscia island is

mapped together with the north-east climate region of France and Ajacco at the west coast of the island is put together with the west and south region of France. For the number of clusters $k=4$, Corsica island is clustered together with the north-east cluster of France instead of the south cluster. For $k=5$, $k=6$, and $k=7$ the clustering solutions were coherent with the geographical locations. For instance, the 5-cluster solution is coherent with the local mountain regions. OSil₁ classified the Armorican mountain series in the north-west together (blue cluster in Figure 4.34d), the Aedennes, Vosges and Jura in the north-east together (purple cluster), the central mountain series Morvan (with its northern extension) together (yellow cluster), the Alps and the Medeterian coastal region together (red cluster), and separated Pyrenees in the south (green cluster). In terms of the number of clusters, the best ASW was obtained for $k=2$ (with ASW at 0.1865). The second best is $k=3$ (with ASW at 0.1581) and the third best is obtained for $k=7$ (with ASW at 0.1253). The clustering produced by OSil₁ for the number of clusters $k=2$ does not look much convincing, whereas the clustering obtained for either 5, 6 or 7 number of clusters appear more coherent with the rainfall patterns in the country.

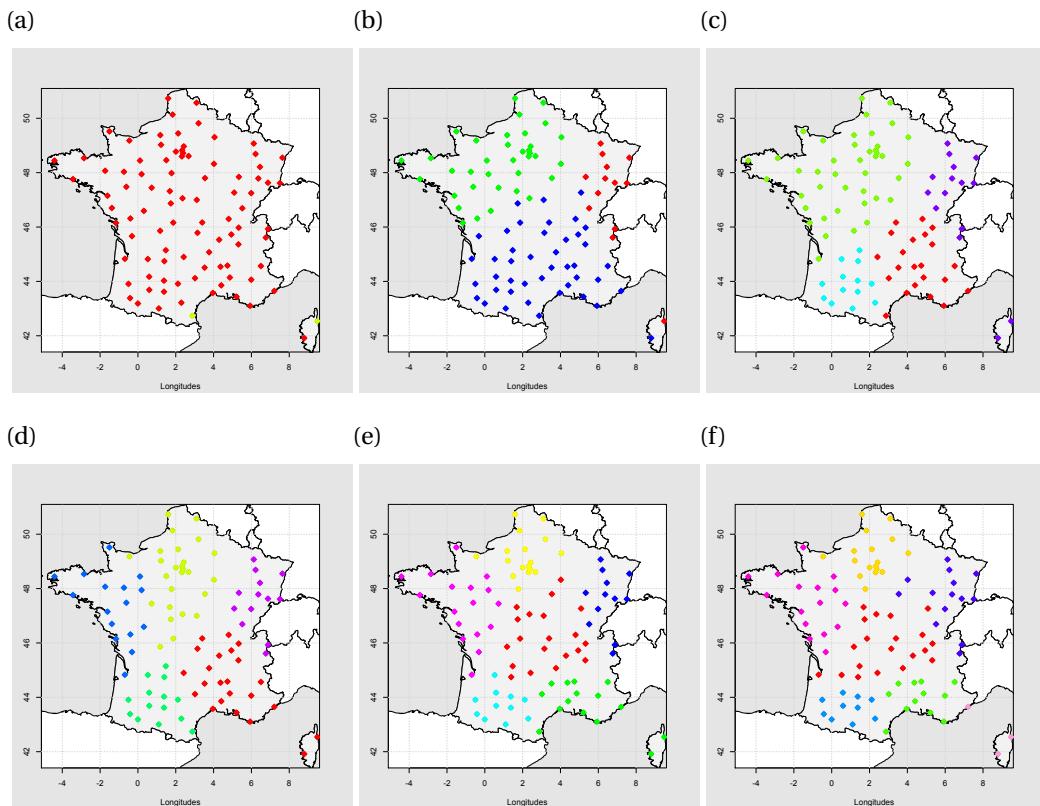


Figure 4.34 Clustering results from OSil₁ algorithm. Panels from (a) - (f) denote clustering against $k = 2$ to $k = 7$.

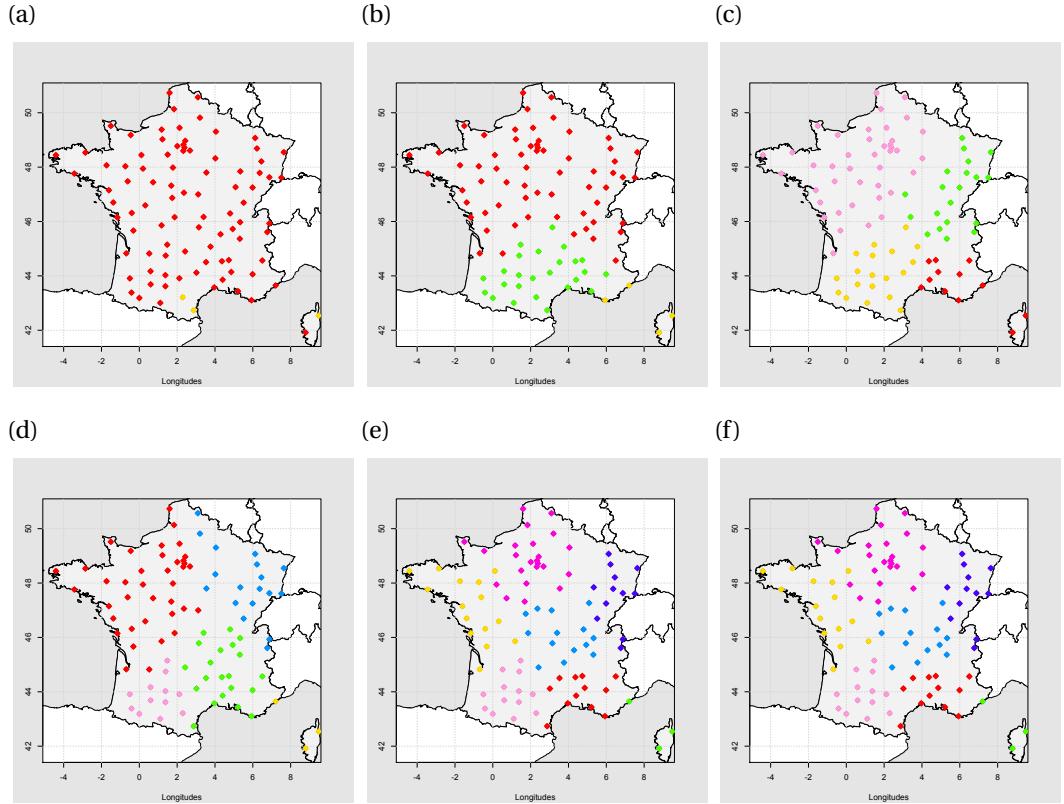


Figure 4.35 Clustering results from PAMSIL. Panels from (a) - (f) denote clustering against $k = 2$ to $k = 7$.

The PAMSIL clustering results for the numbers of clusters 2 to 7 are plotted in Figure 4.35. It's hard to find a climate justification for the clusters produced by PAMSIL. For instance for the numbers of clusters two, Bastia is put together with Perpignan and Carcassonne rather than Nice or Toulon and the upper Alps region is put together with the northern cluster rather than the southern cluster. In terms of the estimation of the number of clusters PAMSIL gives the highest ASW at $k=2(0.1684)$. The second best is $k=3(0.1281)$ and third best is $k=4(0.1224)$. PAMSIL has consistently separated the Alps regions into two parts. For 4-cluster clustering the upper Alps are put together with the north-eastern region and the lower Alps with the Mediterranean region. The east part of France with the Alps must be together with the Nice, Toulon and Corsica or the south cluster rather than with the north cluster. For a number of clusters higher than 4 similar inconsistencies can be observed.

4.17.3 Genetics background

There are many different kinds of cells in Eukaryotic organisms, for instance skin cells, muscle cells, blood cells, nerve cells, or stem cells. These cells are specialised to perform certain functions, except stem cells. The stem cells are unique cells that have the ability to divide and provide new cells to the body as it grows. These cells can divide to more stem cells or other specialized cells like blood cells or muscles cells etc.

Inside the cell, there is nucleus. Inside the nucleus there are many thread like structures called chromosomes. The chromosomes are formed as strings of DNA (Deoxyribonucleic acid). DNA is a molecule that contains all the instructions to build a living organisms and its consistent growth and functioning. A DNA is never ending long intertwined double (two strands) helix structure. Chromosomes keep this never ending structure of DNA by wrapping it compactly around a protein called Histone. Proteins carry many important functions in cells for instance act as a messenger to communicate between cells, tissues and organs to perform or maintain their functions.

The double helix structure can be unwind and flattened to understand the chemical structure inside it. Each strand is called a polynucleotides consists of many simpler units called nucleotides. A nucleotide has three components namely a sugar group, a phosphate group and one of the four possible nitrogenous basis called *Adenine* (A), *Thymine* (T), *Guanine* (G) and *Cytosine* (C). The phosphate groups are bound together to combine all the nucleotides to each other to make the single strand and the hydrogen bound between the bases, pairs the two strands to form the double helix. The order of these bases defines the unique instruction or genetic code to make a certain kind of protein. T can only pair with A and G can only pair with C.

Within the DNA are section called genes. A gene can be seen as a set of letters (some thing like: GTCACGATT). About 99% of DNA contains the non-coding genes. The non-coding genes do not contain instructions for making protein. The non-coding DNA is active and relatively new and less popular research domain. About just only 1% DNA encode proteins meaning that each part (a gene) of the DNA contains a set of instructions to make a protein. The complete set of genes is called genome. The genome is spilt between 23 pairs of chromosomes. A genome is a complete set of instruction to build an entire organism. For every person the arrangement of genes in the genome is same but small differences in the sequence of the bases make them unique.

Broadly speaking there are three types of information known so far that are encoded in genome. Firstly genes that encode protein, secondly regulatory regions that control when genes are expressed/active, what's the activation level and thirdly the regions of genes that encode RNA (Ribonucleic acid essential for expression of genes) molecules.

Gene expression is a process by which genetic code (the nucleotide sequence: GT-CACGATT) from a gene is used by enzymes for the synthesis of protein for construction of cell structure. DNA sequencing allows the researchers to determine the order of

bases in a DNA sequence. This sequence of genome is divided into pieces to read the letters by genome sequencing machines.

To produce proteins DNA must be readout or copied into RNA, i.e., DNA is transcribed into RNA, which is then translated into protein. The genes readouts are called transcripts and a transcriptome is a collection of all the genes readout from a cell. Thus the transcriptome contains all the RNA molecules within a cell. Each cell in the multi-cellular organism carries same DNA or genome but transcriptome varies widely across cell types and functions and can tell many things about genes activity.

Not all the genes are expressed/active all the time. We need to know when a gene is expressed/active. A gene is active when it makes a copy of itself, also the activation level of a gene can vary from cell to cell. The activation of the genes determines certain traits are present in the individual or presence of a disease. Genes are not only active (on) or not active (off) but also, they have certain level of activation. There are several factor which can turn on a gene. It is also possible that gene(s) relate to a certain disease is present in a human but this does not necessary mean the person will get a disease. This gene can be active to do other things.

There are various gene expression profiling techniques probably the most famous one and active in research are serial analysis of gene expression (SAGE), DNA microarray, RNA sequencing and more recently single cell RNA sequencing among others. These are the ways to measure the activation level of a gene in samples. Through these techniques one can learn how genes activation level is varied due to external stimulus or before or after drug dosages. These techniques can also be used to compare the gene of cells that are affected by a disease versus the genes of cells that are not affect by the disease by examining which genes were expressed and how much they are transcribed? This comparison between the normal cells with the mutated cells tells about the genetic mechanism causing the difference. The process can identify structural variations and detect novel genes. They can also be used to study the difference between the cells at different stages of a disease.

DNA microarray is more common choice of researcher when conducting transcriptional profiling experiments. The reason is for RNA-seq the data analysis is complex, need more data storage and expensive in terms of time. However, gene transcripts profiling through both DNA microarray and RNA-seq uses the bulk of cell samples simultaneously. They can't give the cell-to-cell heterogeneity.

4.17.4 Introduction to scRNA-seq technique

The most recently known technique, single cell RNA sequencing (scRNA-seq) is a way to analyse an individual cell and it can inform how each cell is different from other. Single-cell RNA sequencing is a technique to profile the transcriptome of individual cells. The technology was first published in [Tang et al. \(2009\)](#). Since scRNA-seq is a way to analyse an individual cell, it can inform how each cell is different from other.

This has brought revolution in many areas of medical sciences. One important example is cancer research. Identification of the composition of a tumor is crucial in order to understand its biological process. This is also crucial for targeting a cure. For many decades it was believed that a tumor is made up of identical cell populations with identical characteristics. More recently, it is known that cancer tumor is not made up of only one identical cell population but several sub-populations each having different characteristics. These sub populations can substantially differ in terms of their properties. It is very difficult to study the sub-population of tumor because using the earlier mentioned two techniques the tumor can be analysed as a pool of cells only. scRNA-seq is a way to look at the individual cells within a tumor. Through scRNA-seq we can constructed gene expression profile of each cell within the tumor. This will give enough information about the properties and characteristic of each individual cell and can give the architecture of complex tumor composition. By identifying the individual sub-population within the tumor we can actually identify the sub-population, which propagates the tumor growth. By studying the properties of tumor propagated population it becomes much more easier to design a target therapy for the suffering patients.

4.17.5 Identification of cell population

In this thesis we will apply the proposed algorithm to the identification of cell population using scRNA-seq data. The cells are fundamental units in biology. Identification of cell types in several tissues and organs from the mass of heterogeneous cells is an important task in cell biology. This is considered as the first step in the biological analysis of single-cell RNA sequencing (scRNA-seq) data. Using scRNA-seq many different studies have already been conducted on various organs either during development or at fixed time. For instance, in early embryonic development ([Biase et al. \(2014\)](#), [Goolam et al. \(2016\)](#)) or various regions of brain ([Zeisel et al. \(2015\)](#)).

4.17.6 scRNA-seq data analysis workflow

There are various technical steps involved through out the sequencing process (see [Hwang et al. \(2018\)](#), [Shapiro et al. \(2013\)](#) and [Mardis \(2008\)](#) for the pipeline). The process can be broadly divided into three main steps/categories each of which involves several steps. The first step is to prepare the sequencing library, the second is to sequence and third is data analysis. At a high level description, first, solid tissues are dissociated into single cells. Then after cells are isolated, the messenger RNA (mRNA) is separated and reverse transcribed to complementary DNA (cDNA) for high throughput sequencing. Once the sequencing library is prepared the scRNA-seq data is obtained through sequencing.

Sequencing generates the raw data, which is a large collection of the cDNA reads. As a first step this is ensured that the reads are of high quality using standard tools. Af-

ter trimming the low quality reads, they are mapped to the reference genome and the quality of the mapping is checked. The next step is the qualification of the reads i.e., to quantify the expression level of each gene for each cell. The units of measurements of the gene expressions depend on the protocol used. Although scRNA-seq technology has principle steps but within each steps the methodologies differ. The standard gene expression quantification methods used in scRNA-seq are read counting and unique molecular identifier (UMI). The quantified gene expressions are summarized as an expression matrix. Each row of the matrix represents gene and each column represents a cell. The expression matrix is then consider for cleaning. Poor quality cells are removed because they introduce noise in the expression matrix.

The expression matrix is normalized to eliminate the technical variation which is introduced in the gene expression during the sequencing process so that the biological difference of interest are not masked. Depending upon the normalization strategies used the data have other units. Other units are R/CPM (Reads/Counts Per Million), RPKM (Reads Per Kilobase Million), FPKM (Fragments Per Kilobase Million), or TPM (Transcripts Per Million). The normalization strategy used can have a strong affect on downstream analysis. Many normalization methods in scRNA-seq are adhere from bulk RNA sequencing. [Vallejos et al. \(2017\)](#) discussed in detail about the technical considerations while normalizing and showed the difference in results between commonly used normalization methods.

After sequencing, the process of obtaining raw data and transforming it to expression matrix is known as low-level analysis, whereas the further biological analysis is known as downstream analysis. For the efficient storage, quality control (QC) and normalization of the expression matrix software tools has been developed. “SingleCellExperiment” [Lun and Risso \(2017\)](#) is an R package to store the scRNA-seq data. For the low-level analysis, the R libraries (QC, normalization) of scRNA-seq data are “scater” [McCarthy et al. \(2017\)](#) and “scran” [Lun et al. \(2016b\)](#) available through Bioconductor. A step by step workflow to conduct the low-level analysis using these libraries is presented in [Lun et al. \(2016a\)](#).

4.17.7 Clustering scRNA-seq

The development of the novel clustering methods for scRNA-seq is of vital importance. scRNA-seq data clustering is of interest at its own or can be of interest to be used as first step for further analysis. Since much of the downstream analysis is based on clustering the final conclusions may be strongly affected by clustering. Definition or discovery of a new cell type via clustering is an important area of research in the field, for instance, [Villani et al. \(2017\)](#) discovered several new putative cell sub-populations using novel clusters.

There are a few clustering methods specifically designed for scRNA-seq data. A list of these can be found in the Table 1 of [Kiselev et al. \(2019\)](#). Each of these suffers

from some kind of limitations. Some of them are specifically designed for a purpose, example includes identification of rare cell types. Many of them are not scalable to big data sets or for the estimation of large number of clusters. For instance, “SC3” ([Kiselev et al. \(2017\)](#)) is not scalable to big data sets and “Seurat” ([Butler et al. \(2018\)](#)) can handle big data sets but it performs poorly for small data sets as reported in [Kiselev et al. \(2017\)](#).

There are a few challenges while clustering scRNA-seq data ([Kiselev et al. \(2019\)](#)). One of these is the high dimensionality. The total number of genes measured in the experiment is known as dimensionality, that is often at least a few thousands. The two main approaches to deal this issue is to use only a subset of genes or to project the data to some low dimensional space.

Before considering scRNA-seq data clustering SC3 package has been reviewed which offers a specific clustering methods for scRNA-seq data. The SC3 package uses principle component analysis method for the dimensionality reduction. We start the review from it now.

Principal component analysis is a standard linear dimensionality reduction method. The principal components are found by calculating the eigenvectors and eigenvalues of the covariance matrix of data. The eigenvectors with the corresponding eigenvalues gives the directions in which the data has some proportion of the variance of data. The eigenvector with the largest eigenvalue define the direction of the maximum variation of data and hence known as the first principal component of the data set.

Let X be the data matrix having n observations and p dimensions and Σ be the sample covariance matrix of X of order $p \times p$. The first step of PCA is to calculate the eigenvalues and eigenvectors from the covariance matrix. For a square matrix of order $p \times p$ there will be p eigenvalues. Let $\lambda_1, \lambda_2, \dots, \lambda_p$ represents the p eigenvalues of Σ and y_1, y_2, \dots, y_p represents the p eigenvectors. Note that each y_j is a column vector of length p . The roots of the characteristics equation ($|\Sigma - \lambda I| = 0$) gives the eigenvalues of Σ , where λ is a scalar and I is an identity matrix of order $p \times p$. For $p \times p$ matrix Σ there will be p roots of the equation resulting in p eigenvalues as $\lambda_1, \lambda_2, \dots, \lambda_p$. Let \mathbb{O} represents the null column vector of length p . The p eigenvectors are then calculated from $(\Sigma - \lambda_j I)y_j = \mathbb{O}$, $j = 1, \dots, p$. For each value of λ the equation just mentioned will result in a system of p equations to solve simultaneously to get the p co-ordinates of each eigenvector.

Let Λ be a diagonal matrix of order p whose diagonal elements are eigenvalues of

Σ . Let Φ be a $p \times p$ matrix, whose columns are the eigenvectors of Σ , written as follows:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_p \end{bmatrix}, \Phi = \begin{bmatrix} y_1 & y_2 & \cdots & y_p \end{bmatrix}.$$

The Φ matrix is orthogonal matrix i.e., $\Phi\Phi^t = \Phi^t\Phi = I$, the eigenvectors are normalized to unit magnitude, such that $\Phi^t\Sigma\Phi = \Lambda$. The Φ matrix is the linear transformation of data points where the transformed data variables in the new coordinate system are uncorrelated. The correlation matrix of the new data is Λ having all off diagonal elements as zero.

The next step is to order the eigenvectors by eigenvalues from largest to smallest. This gives the components in order of importance. Suppose we want $d < p$ dimensional data, the first d eigenvectors will give the reduced dimensional representation of the original data. Let Φ^* be the matrix made up of the first d eigenvectors such that its order is $(p \times d)$. Let $X^* = X - \mu_j$, $j = 1, \dots, p$, where μ_j is the means of the columns in X . Find the d -dimensional new data as: $(\Phi^*)^t(X^*)^t$. Note that the new data matrix will have the order $(d \times n)$ such that the dimensions are in rows and the observations are along columns now.

SC3 clustering package SC3 stands for single cell consensus clustering. It takes as an input the expression matrix where genes are stored in rows and cell are stored in columns. It produces the clustering using PCA and k -means clustering method. There are seven steps involved in SC3 clustering and at each stage several parameters are needed, which the package sets automatically. It first filters the gene from the expression matrix. The distance measure is then calculated using the filtered gene matrix, which is then followed by the transformation step. At the transformation the linear PCA is applied to get the d eigenvectors. k -means clustering is performed on each of these eigenvectors. A consensus matrix is constructed from these d clusterings, which is then used to produce final clustering. We now describe in detail each of these steps.

SC3 takes as an input the SingleCellExperiment object. It uses both counts (for gene filtering) and logcounts (both normalised and log-transformed expression matrix for clustering). **Gene filter** removes two types of genes, rare genes and ubiquitous genes. The rare genes are those that are expressed in less than Y% of cells where the ubiquitous gene are those that are expressed in (100-Y)% of cells. These genes are not informative for clustering and removing these reduces the dimensionality greatly (see [Kiselev et al. \(2017\)](#)).

The **Distance calculation** is done using three methods namely, Euclidean, Pearson and Spearman on the filtered expression matrix. The resulting distance values are

stored in matrix form. For each distance matrix calculated **transformation** is done using two methods namely, PCA and eigenvectors of the graph Laplacian ($L = I - \tilde{D}^{-\frac{1}{2}}A\tilde{D}^{-\frac{1}{2}}$, see Section 2.4.4 of Chapter 2). The clustering is performed on the columns of the resulting matrices stored in ascending order with respect to their associated eigenvalues. **k-means** clustering is performed on the first d eigenvectors using the `kmeans()` function in R with the Hartigan and Wong algorithm ([Hartigan and Wong \(1979\)](#)), the number of starts set to 1,000, and the maximum number of iterations is set to 10^9 .

Finally a **Consensus** matrix is computed using cluster-based similarity partitioning algorithm (CSPA, see [Strehl and Ghosh \(2002\)](#)). **CSPA** is a binary similarity matrix based on the intuition that the two objects have similarity of 1 if they are in same cluster otherwise, their similarity is 0. If there are n cells (after gene filtration) to cluster the similarity matrix is of order $n \times n$. For each clustering obtained a binary similarity matrix is constructed for cells and a consensus matrix is calculated by averaging all individual similarity matrices. The resulting consensus matrix is clustered using hierarchical clustering with complete linkage to produce final clustering. User specified number of clusters are used or otherwise SC3 estimates it.

Estimation of number of clusters is done from the log-transformed filtered gene expression matrix. A z-score transformation is performed on this matrix i.e., from each column of log-transformed matrix, means are subtracted and then divided by the standardised deviations. Let the z-transformed matrix is denoted by Z . The eigenvalues of $A = Z^tZ$ are calculated. The number of clusters are estimated by the number of eigenvalues that are significantly different than Tracy–Widom distribution (see [Patterson et al. \(2006\)](#) for detail).

We now consider the scRNA-seq data clustering. For this we consider already published data sets for which the true cell types are originally identified by the authors. In addition we have also considered SC3 clustering.

4.17.7.1 [Yan et al. \(2013\)](#) data

Study type: human embryonic development. The author has defined the development stages (cell types) of the cells. The data set contains 90 cells and 20214 expressed gene. The authors have identified 7 cell types as oocyte (3 samples), zygote (3 samples), 2-cell (6 samples), 4-cell (12 samples), 8-cell (20 samples), lateblast (30 samples), and morula (16 samples). The data is available from Gene Expression Omnibus under accession number GSE36552.

The QC and normalization were performed using “scater” with default settings, for this and the all other scRNA-seq data sets considered in this section. We have used “runPCA()” function of “scater” for dimension reduction using principal component

analysis. Euclidean distances between cells were used for clustering for all data sets. The maximum number allowed for the estimation of number of clusters is 10.

Three principal components were used for clustering. The PCA plot of the data is shown in Figure 4.36. The 1st, 2nd and 3rd components define 53%, 26% and 3% variances respectively.

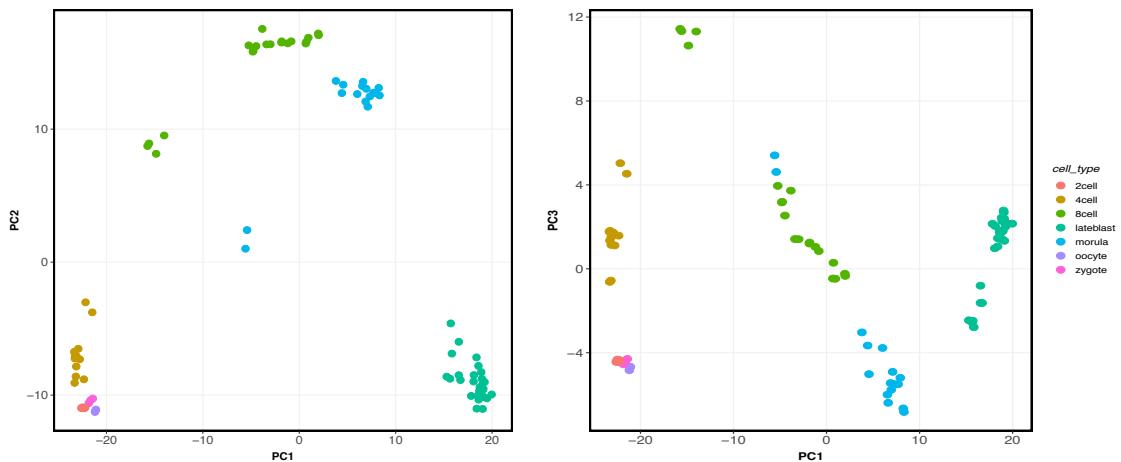


Figure 4.36 Data plots of first three principal components.

Table 4.38 [Yan et al. \(2013\)](#) data clustering results.

Methods	True k		Estimated k		
	ASW	ARI	ASW	ARI	\hat{k}
True labels	0.6557				
k-means	0.7265	0.7554	0.9145	0.6850	3
PAM	0.7916	0.8939	0.7916	0.8939	7
average	0.7916	0.8939	0.8028	0.8773	8
Ward's	0.7916	0.8939	0.7916	0.8939	7
model-based	0.6807	0.7176	0.7554	0.6850	3
spectral	0.3676	0.7224	0.7762	0.7905	5
BIC-mb	-	-	0.6976	0.7288	8
SC3	0.8859	0.6408	0.9713	0.6212	3
PAMSIL	0.7916	0.8939	0.8028	0.8773	8
OSil ₁	0.7916	0.8939	0.8028	0.8773	8
HOSil	0.7916	0.8939	0.7963	0.7956	6

The results for this data are reported in Table 4.38. The ASW using the known classification (true labels) was calculated using the distance between the data obtained from principal components. Only Wards and PAM clustering methods have estimated correct number of clusters. Average linkage, PAMSIL, HOSil and OSil₁ have shown the same performance. SC3 produced higher value of ASW with too low ARI.

4.17.7.2 Biase et al. (2014) data

Study type: the cell fate decision during early embryo development. This data set contains 49 cells and 3 cell types. There are 1-cell (9 samples), 2-cell (20 samples), and 4-cell(20 samples) embryos. The data is available through the accession number GSE57249 from NCBI Gene Expression Omnibus (GEO; <http://www.ncbi.nlm.nih.gov/geo/>). The data has FPKM unit and 25737 genes.

For this data set the two principal components were used. The 1st and 2nd principle components defines 37% and 14% variance respectively. The data is plotted in Figure 4.37 with colours representing the true cell types classification by the authors.

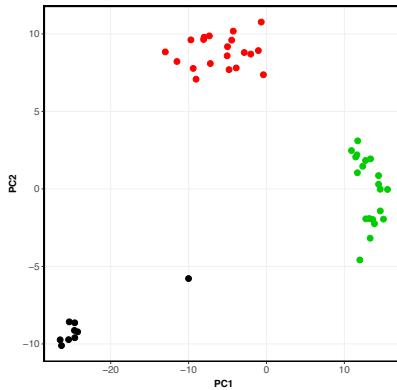


Figure 4.37 Data plots of first two principal components. The cell-types are: 1-cell (black), 2-cell (green) and 4-cell (red).

Table 4.39 Biase et al. (2014) data clustering results.

Methods	True k		Estimated k		
	ASW	ARI	ASW	ARI	\hat{k}
True labels	0.8052				
k-means	0.4673	0.051	0.6917	0.5091	4
PAM	0.8052	1	0.8052	1	3
average	0.8066	0.9483	0.8066	0.9483	3
Ward's	0.8052	1	0.8052	1	3
model-based	0.8052	1	0.8052	1	3
spectral	0.8066	0.9483	0.8066	0.9483	3
BIC-mb	-	-	0.5773	0.5756	6
SC3	0.9270	0.9483	0.9270	0.9483	3
PAMSIL	0.8066	0.9483	0.8066	0.9483	3
OSil ₁	0.8066	0.9483	0.8066	0.9483	3
HOSil	0.8066	0.9483	0.8066	0.9483	3

Table 4.39 shows the result for Biase et al. (2014) data clustering using all the clustering methods considered earlier in this work. FOSil₂ was not applied due to small

number of cells. The performance for PAMSIL, OSil₁ and HOSil is same. These methods have estimated correct number of clusters but have miss classified one cell. Average linkage and spectral clustering methods has also performed equivalent to these methods. However, PAM, Ward's and Model-based (with ASW) clustering gave the best results. They have not only estimated the correct number of clusters but also do not miss classified any points. The performance of *k*-means was poor among all methods for this data. Overall, SC3 gave the highest value of ASW with ARI performance equivalent to other methods.

4.17.7.3 Goolam et al. (2016) data

Study type: pre-implantation development. The data has 124 cells. There are 5 distinct cell types. 2-cell(16 samples), 4-cell(64 samples), 8-cell(32 samples), 16-cell(6 samples) and 32-cell(6 samples). The data is available from under accession number E-MTAB-3321 from ArrayExpress (<https://www.ebi.ac.uk/arrayexpress>).

The data was reduced to three components and Euclidean distances were used between cells to perform clustering. The data set was projected to 3 principled components plotted in Figure 4.38. The colour represents the true cell classification by the authors. The principal components covered 39%, 7%, and 6% variance respectively.

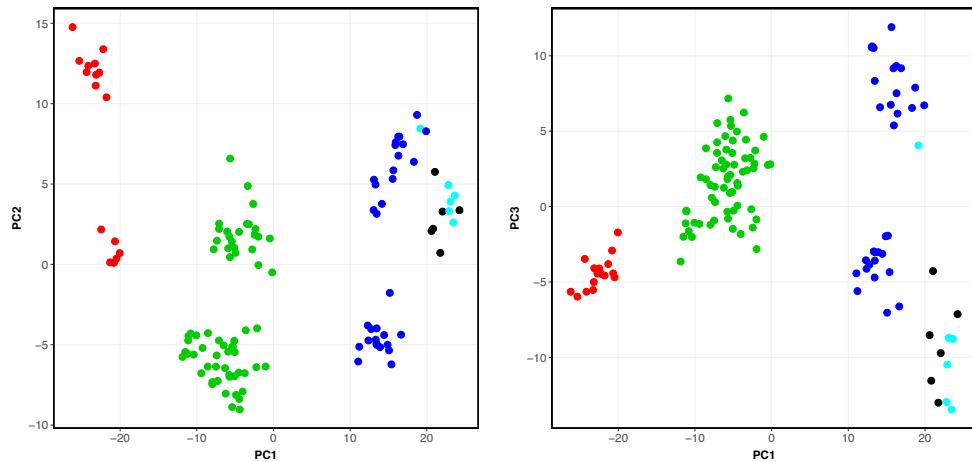


Figure 4.38 Data plots of first three principal components. Shown in red are 2-cell, green are 4-cell, blue are 8-cell, black are 16-cell and light blue are 32-cell stages.

Table 4.40 shows the results for this data. PAMSI, OSil, HOSil, average linkage and Wards clustering methods estimated correct number of clusters and gave the highest ARI value. Although SC3 has produced higher ASW values than all other clustering methods included in the study but it gave low ARI value.

Table 4.40 Goolam et al. (2016) data clustering results.

Methods	True k		Estimated k		
	ASW	ARI	ASW	ARI	\hat{k}
True label	0.4905				
k-means	0.5502	0.5439	0.5995	0.8831	3
PAM	0.5502	0.5439	0.6365	0.8602	4
average	0.6668	0.9097	0.6668	0.9097	5
Ward's	0.6668	0.9097	0.6668	0.9097	5
model-based	0.5502	0.5439	0.6365	0.8602	4
spectral	0.2617	0.6365	0.329	0.8602	4
BIC-mb	-	-	0.5925	0.475	8
SC3	0.8968	0.6874	0.9793	0.6299	2
PAMSIL	0.6668	0.9097	0.6668	0.9097	5
OSil ₁	0.6668	0.9097	0.6668	0.9097	5
HOSil	0.6668	0.9097	0.6668	0.9097	5

4.17.7.4 Kolodziejczyk et al. (2015) data

Study type: mouse embryonic stem cell growth under different culture conditions. The data has 704 cells. The three culture conditions are serum (250 cells), 2i(295 cells) and 2ai(159 cells). The number of clusters are three, where each cluster correspondence to a culture condition. There are sub-populations within each culture condition. The serum grown cells have 3 sub-populations, cell grown under 2i has 4 sub-populations and lastly cell grown under a2i has 2 sub-populations. The data is available from ArrayExpress under accession number E-MTAB-2600.

The data was projected onto 3 principle components shown in Figure 4.39. The 1st, 2nd and 3rd components defined 14%, 9%, and 4% variance respectively.

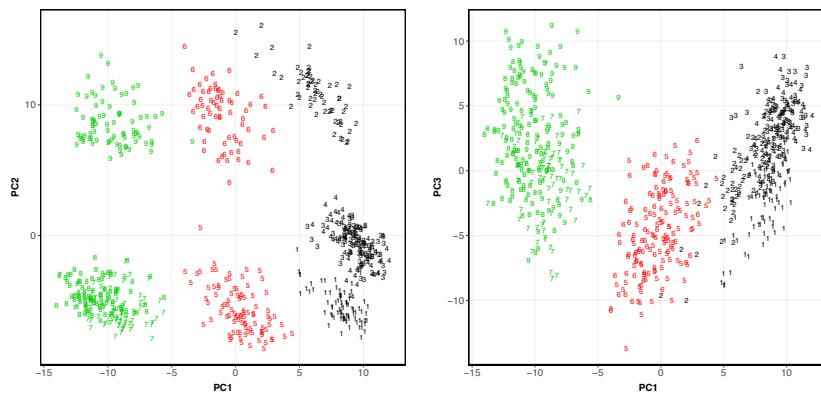


Figure 4.39 Data plots of first three principal components. Shown in black are 2i cells, red are 2ai cells and green are serum cells. The labels shows sub-population within each cell type.

Table 4.41 [Kolodziejczyk et al. \(2015\)](#) data clustering results.

Methods	True k		Estimated k		
	ASW	ARI	ASW	ARI	\hat{k}
True label	0.3659				
k-means	0.4754	0.4366	0.5346	0.5042	7
PAM	0.4704	0.4493	0.5346	0.5042	7
average	0.4632	0.5727	0.5239	0.6090	5
Ward's	0.4742	0.3895	0.5300	0.5316	7
model-based	0.446	0.6165	0.5205	0.5006	6
spectral	0.4742	0.3895	0.5333	0.5225	7
BIC-mb	-	-	0.3382	0.3852	9
SC3	1	1	0.92	0.8317	5
PAMSIL	0.4760	0.4442	0.5354	0.5105	7
OSil ₁	0.4760	0.4442	0.5353	0.5077	7

Table 4.41 shows the clustering results. HOSil clustering was not applied to this data due to greater numbers of cells. The ARI values and true ASW reported in the tables were calculated using data labels for three clusters. There is strong separation between the sub-populations of clusters/cell-types therefore, the methods have estimated number of clusters more than 3. We have also done all calculations using number of clusters as 9. The ARI values were higher with $k=9$ as compared to $k=3$. None of the methods estimated number of clusters as 3 here. SC3 has performed better than other methods here. For the fixed number of clusters (i.e., 3) it gave the true known classification of data correctly, however, it has also not estimated the number of clusters at desired.

Chapter 5

Theoretical foundation

“The generalizations are true, only within a limited scope.”

Fatima Batool

5.1 Background discussion

There has been some advancement in developing a general theoretical framework for clustering functions and algorithms. The aim is to characterize the clustering methods by identifying the mathematical properties which are reasonable to claim as the axioms for clustering. This will allow to understand the task of the clustering independently from the data structures, clustering algorithms or objective functions Kleinberg (2003). Due to numerous clustering applications, a wide range of clustering methods and algorithms have been proposed in literature. Clustering methods are data driven and motivated by a specific data problem. Therefore, many clustering heuristics are not guaranteed to be the optimal in a situation outside the scope for which they were initially developed. In practice it is usually not trivial which clustering method to employ to solve a given problem in hand. The users are always in trouble to choose a clustering method that will fit best in a particular application. It is usually impossible to determine the best clustering procedure as in practice there is a little information available to compare the appropriateness of massive set of algorithms.

There can be agreement on the general purpose of clustering but not on how to achieve this and how to define clusters. Also, if the characteristics are known for the required clustering, these characteristics can be defined in more than one ways. Also, for a given clustering definition, there can be many ways to mathematically achieve it. For instance, if homogeneous patterns are to be sought out from the data, then some notion of homogeneous clusters is required to perform clustering. There is no universally acceptable definition of homogeneity through which clusters are defined. Not all clustering methods can reveal all kinds of clusters in data. Some methods can be good

in uncovering spherical clusters of equal size, for instance k -means, or some can find clusters with unequal sizes and orientation like Gaussian mixtures. Techniques like spectral clustering can identify various shapes for instances rings, but none of them can find out all types of clusters.

Usually it is expected that the user should not only have the knowledge of clustering techniques and related issues, but also the knowledge of the application domain. For instance in what situation these techniques perform best, what kinds of clusters each of these techniques are good in finding, and what are their limitations, also as well as what clustering characteristics are sensible to apply for a data application. It is crucial for users to identify what is the purpose of clustering and what types of clusters they are aiming for.

While some experts have guided users to think about their clustering needs and why they want to cluster data and what they want to achieve from the results afterwards, for instance see [Von Luxburg et al. \(2012\)](#), where they argue that it is meaningless to view clustering as a domain independent mathematical task. Also, in reality there is no universally acceptable definition of true clusters, because cluster analysis has been applied with very different aims in various domains. Since every application is unique, [Hennig \(2015b\)](#) argues that the definition of a good clustering depends heavily on the context and intent for clustering.

On the other hand it was vital to bring some clarity and system to identify homogeneities between clustering approaches and systematically design generic concepts to select a suitable algorithm among diverse approaches to help the community using these techniques. The development of the axiomatic theory for clustering is vital because it can provide directions for the selection of clustering algorithms in practical applications. Once a user has decided what properties they are looking for to solve a clustering problem, they can match these requirements with various clustering methods based on the properties developed through axioms for clustering methods. These properties defined by a set of axioms also allow to compare the performance of the clustering methods or their quality and to speak about the unique advantages that come with each of them.

While it is also debatable which of these axioms are appropriate to force on a clustering method and useful for the practical applications, these axioms can at least provide some guide for at least some clustering methods, if not for all, by classifying them into categories in a systematic way. Another concern in this regard is that it is not clear whether a clustering method which fulfils these axioms theoretically will also perform well for a variety of real life applications as well. For instance the single linkage clustering method has been shown to satisfy various axioms (see [Jardine and Sibson \(1968\)](#), [Zadeh and Ben-David \(2009\)](#) and [Carlsson and Mémoli \(2010\)](#)) but this method often fails in practice for a variety of real life applications.

The task of developing a general theory for clustering is not easy as many clustering

methods are very different in nature, whereas for those who have the same motivation, there is no guarantee that they will behave in the same manner for every problem. The aim of developing a general theory for clustering is not new and various approaches have been introduced in the literature. We will discuss some of these in the next section.

Usually the work in this direction is began by defining some reasonable requirements/rules/properties/axioms that every clustering methods should follow, for instance, the set of admissibility criteria by [Fisher and Ness \(1971\)](#), and then classifying the clustering methods according to these rules. In the following section we have reviewed some of these major approaches, and then focus on the work of [Kleinberg \(2003\)](#) and [Ben-David and Ackerman \(2009\)](#), which is more closely related to our work.

5.2 Existing literature

Researchers are working in different directions for the development of theoretical clustering, for instance, developing axioms for clustering functions/algorithms, developing axioms for clustering quality measures, developing the axiomatic framework by designing clustering problems in weighted setting, developing notions and measures of clusterability. Most often these depend upon the ideas of separation, compactness, stability, consistency and robustness. We don't intend to review all of these topics as our work is the development of the the theory for clustering functions, algorithms and quality measures. These sets of axioms are usually proposed for hierarchical and non-hierarchical (partitional) clustering setups.

Among the earliest attempts in this direction is [Rubin \(1967\)](#). He classified the clustering criteria by taking into account several foundational points including purpose of clustering, types of clusters, types of clustering functions and types of distance measure. He took into account the similarities and difference between different clustering tasks. First of all he required that there should be a well defined mathematical function to find a clustering. Rubin called it splitting function instead of clustering function. The splitting function evaluates many possible clusterings on a set of objects and defined on the base of its optimal value which clustering among them is the best. Rubin then took into account the purpose of clustering which can be different in various domains. He only considered those splitting functions which give non-overlapping clusters. He then classified the splitting functions into two broad classes, one based on geometric measure and other on statistical measure. A statistical measure is based on a probability model of data and a geometric measure that depends upon measure of similarity or dissimilarity between pairs of points. For these he considered measure based on coefficient of similarity and stability. He then defined the set of elementary rules for each of these groups that are reasonable to demand from any clustering methods under these categories. These rules includes characteristics such as well-separated

clusters, homogeneous clusters, and strong clustering structures among others.

Later [Jardine and Sibson \(1968\)](#) outlined properties that any clustering function should satisfy based on dissimilarities among data points and showed many common clustering functions failed to fulfil these. See references therein for earlier work on a theoretical framework for clustering. Following this work, [Fisher and Ness \(1971\)](#) gave a set of nine properties for admissibility of clustering methods based on decision theory. By admissible properties they mean such properties which are reasonable for any clustering procedure to satisfy in general or in particular applications. By restricting clustering procedures with these properties, they hope that the chance of selection of a bad clustering algorithm will vanish. However, these properties are not sufficient alone to choose the best method. Some of these properties are more general. We present only those properties here that are related to this work. Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be the n observations, where each x_i is a column vector. Let $\mathcal{C}_k = \{C_1, \dots, C_k\}$ be a clustering on \mathcal{X} with k clusters.

Definition 5.2.1. Image admissibility: Suppose the points in \mathcal{X} are ordered as x'_1, \dots, x'_n such that

$$C_1 = \{x'_1, \dots, x'_{j_1}\}, C_2 = \{x'_{j_1+1}, \dots, x'_{j_1+2}\}, \dots, C_k = \{x'_{j_1+\dots+j_{k-1}+1}, \dots, x'_n\}.$$

Let y_1, \dots, y_n be any re-ordering of the points and define

$$C'_1 = \{y_1, \dots, y_{j_1}\}, C'_2 = \{y_{j_1+1}, \dots, y_{j_1+2}\}, \dots, C'_k = \{y_{j_1+\dots+j_{k-1}+1}, \dots, y_n\},$$

where C'_1, \dots, C'_k is an image of C_1, \dots, C_k . A clustering is called image admissible if it does not have an image which is uniformly better in the sense that

- $d(x'_i, x'_j) \geq d(y_i, y_j)$ when i^{th} and j^{th} points are in same cluster, and
- $d(x'_i, x'_j) \leq d(y_i, y_j)$ when i^{th} and j^{th} points are in different cluster,

where strict inequality holds for at least one pair of (i, j).

Definition 5.2.2. Well structured clustering: A clustering is well structured if all within-cluster distances are smaller than all between-cluster distances.

[Puzicha et al. \(2000\)](#) developed axioms for clustering methods based on optimization criteria for both non-hierarchical and hierarchical clustering methods based on both similarity and dissimilarity measures. Most of these axioms were based on clustering homogeneity and compactness concepts using invariance, perturbation and robustness properties.

Another line of work in this regard is to develop a set of axioms for weighted clustering settings by considering that each data object has an associated weight, see, for instance [Wright \(1973\)](#). A somewhat different approach was adopted by [Pollard et al.](#)

(1981) for k -means, where he developed the convergence for the clustering criterion as the sample size increases.

Recently Kleinberg (2003) defined three axioms for any reasonable clustering function to obey. The axioms are appealing and sensible to demand from clustering functions but yet he showed that no clustering method can fulfil all three rules. On contrary, Ben-David and Ackerman (2009) claim that Kleinberg's impossibility result is due to their specific formulation, and that these axioms can serve as a consistent set of axioms by redefining them for clustering quality measures instead of clustering functions.

Similar approaches to Kleinberg (2003) are Correa-Morris (2013) and Zadeh and Ben-David (2009), where they further extend this work in similar way. Correa-Morris (2013) has mentioned some key factors ignored in Kleinberg (2003) formulation. They made some adjustment to Kleinberg (2003) formulation by introducing three types of consistency. All of their axioms were strongly linked to robustness of the clustering functions. Zadeh and Ben-David (2009) also followed the notion of Kleinberg (2003) and introduced a relaxation to consistency axiom of the paper mentioned latter, to make the set of axioms consistent. Carlsson and Mémoli (2013) focused mainly on the hierarchical clustering setup, in particular on single linkage. They modified Kleinberg (2003)'s axioms to show that all of the three axioms are satisfied within their formalism.

Ben-David and Ackerman (2009) proposed a consistent set of axioms, for clustering quality measures, namely scale-invariance, consistency and richness. In this work we have followed their work and have shown that ASW satisfies this set of axioms. We now define notations and review in detail Kleinberg (2003) and Ben-David and Ackerman (2009) before proving the axioms proposed by the latter authors for the ASW index.

5.3 Preliminaries

Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be the data set with n observation taken over p variables of interest.

Definition 5.3.1. A **distance function** d is defined over \mathcal{X} as a mapping of each pair in \mathcal{X} to the positive real domain \mathbb{R}^+ i.e., $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$ satisfying the symmetry and reflexivity properties $\forall x_i, x_j \in \mathcal{X}$.

Definition 5.3.2. For a distance function d over \mathcal{X} and a positive real η , the scalar multiplication of d with η is defined for every pair $x_i, x_j \in \mathcal{X}$, as $(\eta \cdot d)(x_i, x_j) = \eta \cdot d(x_i, x_j)$.

We call a clustering function a k -free clustering function when the number of clusters is not needed to be fixed a prior to clustering function.

Definition 5.3.3. A **k -free clustering function** f takes a pair (\mathcal{X}, d) as an input and returns a possible partitioning $\mathcal{C} \in \mathcal{S}(\mathcal{X})$ of \mathcal{X} , where $\mathcal{S}(\mathcal{X})$ denotes a set of all possible partitions of \mathcal{X} .

Definition 5.3.4. A k -clustering function f takes a triplet (\mathcal{X}, d, k) where $1 \leq k \leq |\mathcal{X}|$, and outputs a clustering \mathcal{C}_k having k clusters of \mathcal{X} .

A k -free clustering function does not need k in advance to be provided to perform clustering, and it can return any number of clusters, whereas a k -clustering function will need a predefined k a priori to pass to the function to return a clustering for that chosen number of clusters.

A k -clustering is denoted as $\mathcal{C}_k = \{C_1, \dots, C_k\}$, where $C_r, r \in \mathbb{N}_k$ denotes the clusters in \mathcal{C}_k . Since the number of clusters do not need to be fixed in advance for these axioms, we will work with a clustering say \mathcal{C} which can have any number of clusters. Let $x_i \sim_{\mathcal{C}} x_j$, if observation x_i and x_j for $i \neq j \in \mathbb{N}_n$ belong to the same cluster in a partition \mathcal{C} and $x_i \not\sim_{\mathcal{C}} x_j$, otherwise.

Let d and d' be two distance functions on a partition \mathcal{C} of \mathcal{X} .

Definition 5.3.5. A distance function d' is a \mathcal{C} -transformation of d , if $d'(x_i, x_j) \leq d(x_i, x_j)$ for all $x_i \sim_{\mathcal{C}} x_j$ and $d'(x_i, y_j) \geq d(x_i, x_j)$ for all $x_i \not\sim_{\mathcal{C}} x_j$ for all $i, j \in \mathbb{N}_n$.

d' is defined by increasing the between cluster distances and by decreasing the within cluster distances. One illustration of that is given in Figure 5.1, where the right hand panel represents the \mathcal{C} -transformation on the data shown on left hand panel.

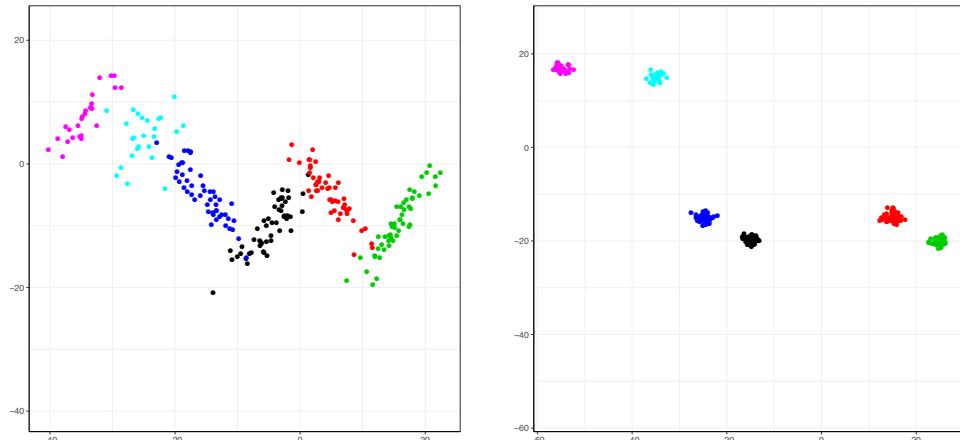


Figure 5.1 An example of \mathcal{C} -transformation on a data set obtained by shrinking the within cluster distances and increasing the distances between cluster centres on the data set shown in the left panel.

Kleinberg (2003) suggested three simple properties for any standard clustering function (CF) f . Let $\mathcal{S}(\mathcal{X})$ represent the collection of all possible partitions of \mathcal{X} . The

clustering function takes the data set in form of pair (\mathcal{X}, d) as input and returns the clustering \mathcal{C} as an output ($f(\mathcal{X}, d) = \mathcal{C} \in \mathcal{S}(\mathcal{X})$). The first property requires that the output of a clustering function should not be affected by the measurement units of (\mathcal{X}, d) .

CF Scale Invariance: For all $\eta > 0$, a function f is invariant to uniform scaling if $f(\mathcal{X}, d) = f(\mathcal{X}, \eta \cdot d)$.

The second property states that a clustering function should be capable of yielding any partition \mathcal{C} of \mathcal{X} from $\mathcal{S}(\mathcal{X})$ by constructing a distance function d on \mathcal{X} .

CF Richness: A CF is rich if for every possible partition $\mathcal{C} \in \mathcal{S}(\mathcal{X})$, there exists a corresponding d on \mathcal{X} such that $f(\mathcal{X}, d) = \mathcal{C}$.

Let d be a distance function on \mathcal{X} and d' be another distance function on \mathcal{X} defined by shrinking distances within clusters and expanding distances between clusters. A CF is consistent if the clusterings it produces on \mathcal{X} are the same using d and d' .

CF Consistency: A function f is consistent if $f(\mathcal{X}, d) = f(\mathcal{X}, d')$, where d' is a \mathcal{C} -transformation of d .

The aim of developing natural properties/axioms of Kleinberg (2003) for clustering is no doubt useful and relevant for developing understanding to general purpose of clustering. These axioms can distinguish the clustering algorithms from each other and leads to more informed decisions for the selection of the clustering algorithms for a given data application. Scale invariance is useful because it is reasonable to demand that the output of the a clustering function should not change due to the change of measurement scale. Consistency is a rather intuitive requirement based on a view that one wants to have clusters that are at the same time homogeneous (low distances) and separated (large distances to other clusters). Consistency is about decreasing within-cluster distances and increasing between-cluster distances. This states that the consistent changes to distance does not change the clustering output. If a method doesn't fulfill this, a practitioner interested in this kind of clustering may not want to use that method. Richness is relevant in practice insofar that if this is not fulfilled, certain clusterings are impossible to achieve, and the practitioner needs to keep in mind that these clusterings were not ruled out properly by the data but were not possible for any data to achieve in the first place. So the data was not the reason why such a clustering wasn't found.

Kleinberg states *the impossibility theorem* refering that there is no clustering function that satisfies all of the above three properties. Ben-David and Ackerman (2009) identify that the three desired properties can be achieved if we modify them for clustering quality measures instead of clustering functions. They have discussed that the Kleinberg (2003)'s impossibility result occurs mainly because of the consistency property on clustering functions, which requires that the original clustering remains the

same after consistent changes to the distances. The consistency property basically states that if consistent changes (i.e., \mathcal{C} -transformation) are made to the distances, the clustering function should not nominate some other clustering as the best clustering than it gave before. However through \mathcal{C} -transformation there could be many possibilities to get some other clustering whose quality is better than the original clustering while also maintain the quality of the original clustering. In fact \mathcal{C} -transformation allows much flexibility to redefine the within and between clusters distances in such a way that some other clustering can be an even better contestant than the original clustering. For instance one can introduce bigger between cluster gaps for only few clusters using \mathcal{C} -transformation to create some better clustering instead of introducing same between cluster gaps for all clusters. Once the restriction of getting the same clustering quality after \mathcal{C} -transformation is replaced with the same and better quality for the CQMs rather than CF the impossibility theorem no longer exists.

The process of clustering quality assessment tells us about the goodness and usefulness of the clustering structure obtained from any algorithm. A clustering quality measure (CQM) Π takes the pair (\mathcal{X}, d) and a clustering \mathcal{C} over (\mathcal{X}, d) and returns a non-negative real number. In addition, a CQM can also satisfy additional properties. We now give [Ben-David and Ackerman \(2009\)](#)'s three requirements for CQMs:

CQM Scale Invariance: *A CQM Π is scale invariant if for all $\eta > 0$, and every \mathcal{C} of (\mathcal{X}, d) , $\Pi(\mathcal{C}, (\mathcal{X}, d)) = \Pi(\mathcal{C}, (\mathcal{X}, \eta \cdot d))$.*

CQM Consistency: *A CQM Π is consistent measure if for every clustering \mathcal{C} over (\mathcal{X}, d) , $\Pi(\mathcal{C}, (\mathcal{X}, d')) \geq \Pi(\mathcal{C}, (\mathcal{X}, d))$ holds, provided that d' is a \mathcal{C} -transformation of d .*

CQM Richness: *A CQM Π is rich for every possible non-trivial clustering $\mathcal{C} \in \mathcal{S}(\mathcal{X})$ of \mathcal{X} there exist a distance function d over \mathcal{X} such that $\mathcal{C} = \arg \max_{\mathcal{C}} \Pi(\mathcal{C}, (\mathcal{X}, d))$.*

Richness is defined only for non-trivial clusterings. There are two cases which are considered trivial. This is when every observation forms a cluster such that there are n singleton clusters for a data set of size n . The other trivial clustering case is when all the observations are in one cluster.

Definition 5.3.6. *Consistent set of axioms:* Every object from the class of objects (for instance clustering functions) for which the set of axioms have been defined should follow all axioms individually.

See [Ben-David and Ackerman \(2009\)](#) for a detailed discussion on the consistency for the set of axioms in the Section 4.2 of their paper.

Theorem 1 ([Ben-David and Ackerman \(2009\)](#)). *Scale-invariance, richness and consistency for clustering-quality measures form a consistent set of axioms.*

5.4 Characterization of the ASW

Here we will explore which of the three requirements given in [Ben-David and Ackerman \(2009\)](#) are satisfied by the ASW. In order to make it easy for the readers to follow the proofs, intuition for the proofs before the start of the proof is also provided.

Definition 5.4.1. Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be the data set of n objects and d be a distance function over \mathcal{X} and \mathcal{C} be some clustering characterized on \mathcal{X} . Let the clustering labels be $l(1), \dots, l(n) \in \mathbb{N}_k$ determined by $l(i) = r$, $i \in \mathbb{N}_n$ and cluster sizes are determined by $n_r = \sum_{i=1}^n \mathbf{1}(l(i) = r)$, $r \in \mathbb{N}_k$. The silhouette width for a data index $i \in \mathbb{N}_n$ is

$$S_i(\mathcal{C}, d) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad (5.1)$$

where

$$a(i) = \frac{1}{n_{l(i)} - 1} \sum_{\substack{l(i)=l(j) \\ i \neq j}} d(x_i, x_j) \quad \text{and} \quad b(i) = \min_{r \neq l(i)} \frac{1}{n_r} \sum_{l(j)=r} d(x_i, x_j).$$

Definition 5.4.2. The **Average Silhouette Width** (ASW) of a clustering \mathcal{C} is defined as

$$\bar{S}(\mathcal{C}, d) = \frac{1}{n} \sum_{i=1}^n S_i(\mathcal{C}, d).$$

Theorem 2. *The ASW is a Scale Invariant CQM.*

Proof. For any $\eta > 0$ and any distance function d on \mathcal{X} , let $\eta \cdot d = d'$. Let $a'(i), b'(i)$, $S'_i(\mathcal{C}, d)$ and $\bar{S}'(\mathcal{C}, d)$ be based on d' . Therefore, $a'(i) = \eta \cdot a(i) = \frac{\eta}{n_{l(i)} - 1} \sum_{\substack{l(i)=l(j) \\ i \neq j}} d(x_i, x_j)$

and $b'(i) = \eta \cdot b(i) = \eta \cdot \min_{l(i) \neq r} \frac{1}{n_r} \sum_{l(j)=r} d(x_i, x_j)$, by Definition 5.3.2.

Now $S'_i(\mathcal{C}, d) = S_i(\mathcal{C}, \eta \cdot d) = \frac{b'(i) - a'(i)}{\max\{a'(i), b'(i)\}} = \frac{\eta \cdot b(i) - \eta \cdot a(i)}{\eta \cdot \max\{a(i), b(i)\}} = S_i(\mathcal{C}, d)$.

hence, $\bar{S}(\mathcal{C}, \eta \cdot d) = \bar{S}(\mathcal{C}, d)$ is always true. Thus for any $\eta > 0$ and any clustering \mathcal{C} of (\mathcal{X}, d) , we have, $\Pi(\mathcal{C}, (\mathcal{X}, \eta \cdot d)) = \Pi(\mathcal{C}, (\mathcal{X}, d))$ \square

Consistency is about getting the same or higher clustering quality after decreasing within-cluster distances and increasing between-cluster distances. For every individual point the $b(i)$ becomes bigger and the $a(i)$ smaller, so all the numerators of the silhouette width will improve. However it may happen that denominators also become bigger, and therefore one has to go through these cases looking at whether $a(i)$ or $b(i)$ in the denominator (which may change when changing the distances) to show that in fact the silhouette width always becomes better or at least not worse. In order to prove consistency for an index one has to observe what change in the value (or the individual expressions and how they all add up) of the index will happen with a consistent change in distances.

Theorem 3. *The ASW is a consistent CQM.*

Proof. Let d' be a \mathcal{C} -transformed distance function of d and $a'(i)$, $b'(i)$, $S'_i(\mathcal{C}, d')$, $\bar{S}'(\mathcal{C}, d')$ be based on d' . The following two inequalities hold by Definition 5.3.5:
 $d'(x_i, x_j) \leq d(x_i, x_j)$ for all $x_i \sim_{\mathcal{C}} x_j$ and $\min_{x_i \sim_{\mathcal{C}} x_j} d'(x_i, x_j) \geq \min_{x_i \sim_{\mathcal{C}} x_j} d(x_i, x_j)$.
This implies that

$$a'(i) \leq a(i), \quad (5.2)$$

and

$$b'(i) \geq b(i). \quad (5.3)$$

For consistency we need to prove,

$$\begin{aligned} S'_i(\mathcal{C}, d) &\geq S_i(\mathcal{C}, d), \\ \Leftrightarrow \frac{b'(i) - a'(i)}{\max\{a'(i), b'(i)\}} - \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} &\geq 0. \end{aligned} \quad (5.4)$$

There can be four possible cases:

$$\text{Case I: } \max\{a(i), b(i)\} = a(i), \quad \max\{a'(i), b'(i)\} = a'(i). \quad (5.5)$$

$$\text{Case II: } \max\{a(i), b(i)\} = a(i), \quad \max\{a'(i), b'(i)\} = b'(i). \quad (5.6)$$

$$\text{Case III: } \max\{a(i), b(i)\} = b(i), \quad \max\{a'(i), b'(i)\} = a'(i). \quad (5.7)$$

$$\text{Case IV: } \max\{a(i), b(i)\} = b(i), \quad \max\{a'(i), b'(i)\} = b'(i). \quad (5.8)$$

We will now check whether the inequality given in (5.4) will hold for each of these cases.

Case I: We have $a(i) \geq b(i)$ and $a'(i) \geq b'(i)$. Combining these two conditions with (5.2), (5.3) we can draw the following scale to understand the relationship between these four quantities.



Using (5.5) in (5.4) we need to show,

$$\begin{aligned} \frac{b'(i) - a'(i)}{a'(i)} - \frac{b(i) - a(i)}{a(i)} &\geq 0, \\ \frac{b'(i)}{a'(i)} - \frac{b(i)}{a(i)} &\geq 0, \\ L' - L &\geq 0, \end{aligned} \quad (5.9)$$

where $L' = \frac{b'(i)}{a'(i)}$ and $L = \frac{b(i)}{a(i)}$. Note that (5.9) will be always true if L is smaller than L' . Since $b'(i) > b(i)$, the numerator of L' is bigger than the numerator of L . Also, the

denominator of L is smaller than the denominator of L' because $a'(i) > a(i)$. Thus (5.9) will always hold¹.

Case II: We have $a(i) \geq b(i)$ and $b'(i) \geq a'(i)$. Using (5.6) in (5.4) we need to show,

$$\frac{a'(i)}{b'(i)} + \frac{b(i)}{a(i)} \leq 2. \quad (5.10)$$

(5.10) will always hold due to (5.6), which will keep both ratios on left hand side of (5.10) less than one.

Case III: We have $b(i) \geq a(i)$ and $a'(i) \geq b'(i)$. Then $a'(i) \geq b'(i) \geq b(i) \geq a(i)$, which is a contradiction to (5.2), hence this case will never exist.

Case IV: We have $b(i) \geq a(i)$ and $b'(i) \geq a'(i)$. Combining these two conditions with (5.2), (5.3) we can draw the following scale to understand the relationship between these four quantities.



Using (5.8) in (5.4) we need to show,

$$\begin{aligned} \frac{b'(i) - a'(i)}{b'(i)} - \frac{b(i) - a(i)}{b(i)} &\geq 0 \Leftrightarrow \\ \frac{a(i)}{b(i)} - \frac{a'(i)}{b'(i)} &\geq 0 \Leftrightarrow \\ L - L' &\geq 0, \end{aligned} \quad (5.11)$$

where $L = \frac{a(i)}{b(i)}$ and $L' = \frac{a'(i)}{b'(i)}$. For (5.11) to be true, $L' \leq L$ should hold always. Now the numerator of L is greater than numerator of L' and denominator of L is less than denominator of L' . The relationships between these quantities are of such kind that (5.11) will always hold².

Recall that ASW is the average of $S_i(\mathcal{C}, d)$ over all $i \in \mathbb{N}_n$, from above results it follows that $\bar{S}(\mathcal{C}, d) \leq \bar{S}'(\mathcal{C}, d)$ always true. Thus, $\Pi(\mathcal{C}, (\mathcal{X}, d')) \geq \Pi(\mathcal{C}, (\mathcal{X}, d))$ always holds. \square

Richness involves optimisation ($\arg \max$) and state that it is not possible to get a

¹both L' and L are less than one in the entire proof.

²The smallest ($a'(i)$) in these four quantities is divided by the largest ($b'(i)$). Which will insure L' is never greater than L . When the values of these four quantities are far from each other this is quite obvious, whereas if the values of these quantities lies close to each other $L - L'$ will also be close to zero.

better clustering than \mathcal{C} with a given distance definition of \mathcal{X} . The distance can be defined in any way because richness requires that it should be possible to construct the distance for any desired partition \mathcal{C} in order to make this partition the best partition of the \mathcal{X} for that distance [Kleinberg \(2003\)](#). This is done in literature by defining distance that has all within-cluster distances (of \mathcal{C}) small and all between-cluster distances (of \mathcal{C}) large ([Kleinberg \(2003\)](#), [Zadeh and Ben-David \(2009\)](#), [Ben-David and Ackerman \(2009\)](#)). In order to prove richness holds for an index one has to consider all other possible clusterings than \mathcal{C} and show that none of them give a better value of index for that given distance definition.

There are some indices like Gamma (see [Ackerman \(2012\)](#), chap 3) for which it is possible to set the distance definition in such a way that the maximum value of the index can be achieved, such that for any other clustering, the value of the index can only decrease. For ASW it is not possible to achieve the exact maximum value of the index. For instance, for $a(i) = 1, b(i) = 2, \bar{s} = 0.5$, and for $a(i) = 1, b(i) = 1000, \bar{s} = 0.999$. The purpose here is not to achieve the maximum value of the index but in fact to show what ever value of index is achieved, no other clustering can give higher value than this value for this given distance. We only need to consider the within-cluster distance to be some real value say r_1 and between-cluster distance, say, r_2 , such that $r_1 < r_2$.

Therefore, the strategy for proving richness for ASW is first construct a distance function and calculate the ASW value with it. Next, one needs to then show that no other clustering can give better value than what one got already. Surely putting points together that are not together in \mathcal{C} , one can generate a large within-cluster distance, which is bad (for the $a(i)$ of the silhouette width of these points). If one split up clusters of \mathcal{C} , this will generate small between-cluster distances, which is bad for the $b(i)$ of the silhouette width of these points. Therefore, one needs to show that in these cases in fact the ASW becomes worse. There needs to be a separate treatment for one-point clusters (in \mathcal{C} and in the clustering to which \mathcal{C} is compared), because these have no within-cluster distances and are handled by separate definition.

All the possible cases to proof richness are presented in Figure (5.2). In order to proof the theorem one has to consider all possible unique cases for \mathcal{C} . There are two cases possible for \mathcal{C} for ASW because of the separate definition for the ASW for singleton clusters. These are named as “Case 1” and “Case 2” in the figure. Next for these cases one has to consider all possible clusterings say \mathcal{C}' other than \mathcal{C} . For \mathcal{C}' one has to again consider singletons separately. The proof then considers all of these cases separately for the evaluation of the ASW value. An example in the Appendix D provides the intuition on how to calculate the ASW values for \mathcal{C}' using the distance defined for \mathcal{C} .

Theorem 4. *The ASW is a Rich CQM.*

We first sketch proof of the theorem 4. In order to prove ASW is a rich CQM we need to consider every possible non-trivial clustering \mathcal{C} and construct a distance function d

for it such that no other clustering \mathcal{C}' is a better opponent i.e., no other clustering can give an improved ASW value beyond \mathcal{C} .

There exist more than one possibility for \mathcal{C} and \mathcal{C}' . Each of these possibilities is drawn in Figure (5.2).

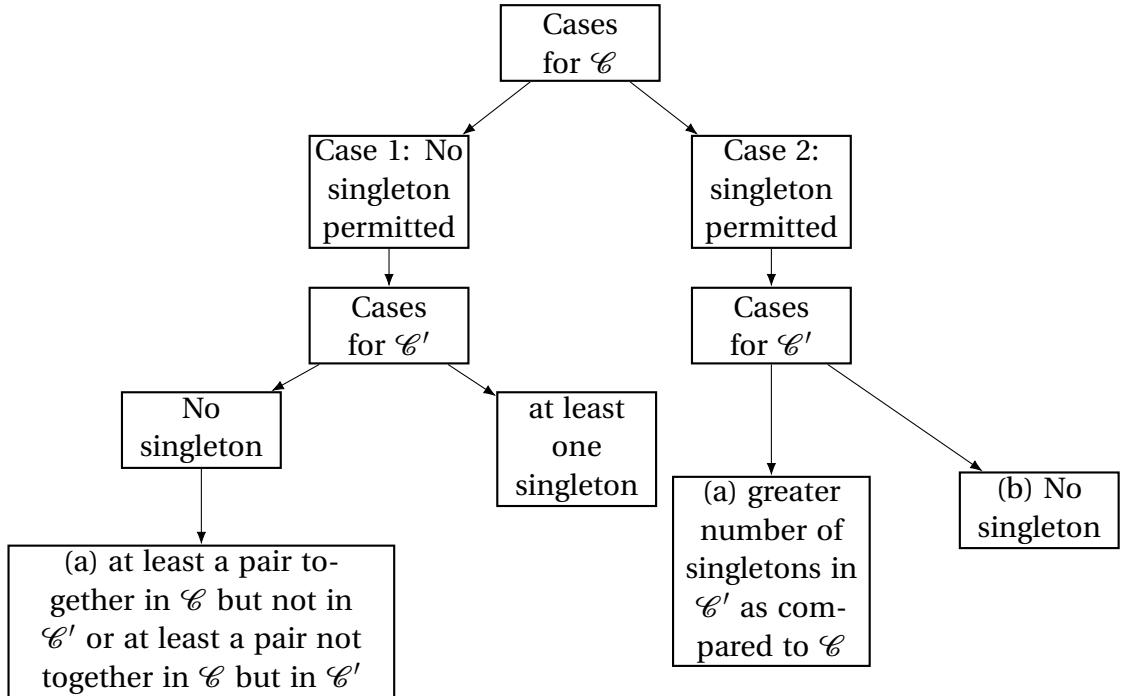


Figure 5.2 All possible cases to consider for the proof of richness theorem for ASW.

There can be two other cases under Case 2 which is smaller or equal number of singletons in \mathcal{C}' than \mathcal{C} but since the same arguments applies as in Case 2(b) therefore separate case distinctions have not been made for these cases. Some of these cases have sub-case distinctions. They are presented in the proof. A simple case of Case 1(a) presented in Figure 5.2 has been stated and proved first separately in Lemma 1. In the proof of theorem itself the generalization of this has been discussed. The argument for the special case and general case is not same but the argument for the latter case is followed from lemma, therefore proved first.

Lemma 1. *Let \mathcal{C} be a clustering of \mathcal{X} with no singletons, obtained by some clustering function f using the distance function d as: $d(x_i, x_i) = 0$ and $d(x_i, x_j) = 1$ if $x_i \sim_{\mathcal{C}} x_j$ and $d(x_i, x_j) = 2$ if $x_i \not\sim_{\mathcal{C}} x_j$ for all $i, j \in \mathcal{X}_n$. If \mathcal{C}' is a clustering such that it is similar to \mathcal{C} in all respects except that there is only one pair of points that is together in \mathcal{C}_k but not in \mathcal{C}' , then $\bar{S}(\mathcal{C}', d) < \bar{S}(\mathcal{C}, d)$.*

Proof of Lemma 1. Due to definition of d for all points $i \in \mathcal{X}$, $a(i) = \frac{(n_{r^*}-1) \times 1}{(n_{r^*}-1)} = 1$ and $b(i) = \frac{(n_{r^\dagger}) \times 2}{n_{r^\dagger}} = 2$, where n_{r^*} and n_{r^\dagger} are the numbers of objects in some clusters C_{r^*} and C_{r^\dagger} of \mathcal{C} . This gives $S_i(\mathcal{C}, d) = 0.5$, for all $i \in \mathcal{X}$ such that $\bar{S}(\mathcal{C}, d) = 0.5$.

Let C_{r^*} and C_{r^\dagger} be two clusters in \mathcal{C} , and \mathcal{C}' be a clustering such that the point h in cluster $C_{r^*} \in \mathcal{C}$ is now in cluster $C_{r^\dagger} \in \mathcal{C}'$. Let the corresponding clusters to C_{r^*} and C_{r^\dagger} in \mathcal{C} be denoted by $C'_{r^*} \in \mathcal{C}'$ and $C'_{r^\dagger} \in \mathcal{C}'$ respectively. For the point $h \in C'_{r^\dagger}$, $S_h(\mathcal{C}', d) = \frac{1-2}{2} = -0.5$ opposite of what it had earlier due to the definition of d . The $S_i(\mathcal{C}', d)$ for all the points in the cluster $C'_{r^*} \in \mathcal{C}'$ and $S_i(\mathcal{C}', d)$ for all the points in $C'_{r^\dagger} \in \mathcal{C}'$ will reduce from 0.5.

The reason for the reduction in $S_i(\mathcal{C}', d) \in C'_{r^*}$ for all i 's in C'_{r^*} is the reduction in $b'(i)$ values from 2 whereas there will be no change in $a'(i)$ as compared to the $a(i)$ in $C_{r^*} \in \mathcal{C}$. This reason of reduction in $b'(i)$ is due to the fact that one point that was previously the member of this cluster has moved out to other cluster which has provided a new lower value than 2 of between cluster distances for the points in this cluster. Since now one of the $d(x_i, x_j)$ is 1 for the case $x_i \sim_{\mathcal{C}} x_j$ which were all 2 previously, will reduce the value of $b'(i)$. Note that, as a result to this $b'(i) - a'(i) < b(i) - a(i)$ will always hold for all i 's in C'_{r^*} . Next we have to look for the minimum value $b'(i)$ can take to determine $\max\{b'(i), a'(i)\}$. Let $n_{C'_{r^*}}$ be the number of objects in cluster C'_{r^*} , then $b'(i) = \min_{r \neq r^*} \frac{1}{n_{C'_r}} \sum_{j=1, i \neq j}^{n_{C'_r}} d(x_i, x_j)$. Since $d(x_i, x_j)$ is either 1 or 2 and in addition we know that one of the $d(x_i, x_j) = 1$ for $x_i \sim x_j$ which were all two previously therefore, $1 \leq b'(i) < 2$. Since $a'(i) = 1$ and $b'(i)$ can not be equal to 2 any more but remains less than it and can decreased to at most 1 therefore $\max\{a'(i), b'(i)\} = b'(i)$ except when $b'(i) = 1$ where we no longer need to care about max. Also, note that $b'(i) - a'(i)$ can at most be as small as 0. Since for all i 's in C'_{r^*} , $1 = a'(i) \leq b'(i) < 2$ therefore, this will ensure $S_i(\mathcal{C}', d) < S_i(\mathcal{C}, d)$. In general, if there are $n_{C_{r^*}}$ points in cluster C_{r^*} there will be $n_{C_{r^*}}$ values of $S_i(\mathcal{C}', d)$ less than $S_i(\mathcal{C}, d)$.

The reason for the reduction in $S_i(\mathcal{C}', d) \in C'_{r^\dagger}$ is the increase in $a'(i)$ whereas $b'(i)$ for $i \in C'_{r^\dagger}$ in \mathcal{C}' will not change as compared to $i \in C_{r^\dagger}$ in \mathcal{C} . The $a'(i)$ will increase from 1 because of the object h in cluster C_{r^\dagger} . Since one of the $d(x_i, x_j) = 2$ for $x_i \sim_C x_j$ which were all 1 previously, resulting in all $a'(i) > a(i)$ for all $i \in C_{r^\dagger}$. Note that for all $i \in C_{r^\dagger}$, $a'(i) > a(i)$ resulting in $b'(i) - a'(i) < b(i) - a(i)$. Also, this is possible to derive the maximum value which $a'(i)$ can achieve. Let $n_{C'_{r^\dagger}}$ be the number of points in the cluster C'_{r^\dagger} , then $a'(i) = \frac{1}{(n_{C'_{r^\dagger}}-1)} \sum_{j=1, i \neq j}^{(n_{C'_{r^\dagger}}-1)} d(x_i, x_j)$. Since $d(x_i, x_j)$ is either 1 or 2 and in addition we know one of the $d(x_i, x_j) = 2$ for $x_i \sim_C x_j$, therefore, $1 < a'(i) \leq 2$. The highest $a'(i)$ can achieve is 2 such that $\max\{a'(i), b'(i)\} = b'(i)$. Also note that $b'(i) - a'(i) = 0$ in this case as well. Since $1 < a'(i) \leq 2$ and $b'(i) = 2$, therefore $S_i(\mathcal{C}', d) \in C'_{r^\dagger}$ for all $i \in C'_{r^\dagger}$. In general, if there are $n_{C_{r^\dagger}}$ number of observations in C_{r^\dagger} there will be

$n_{C_r^{\dagger}}$ number of values of $S_i(\mathcal{C}', d)$ less than $S_i(\mathcal{C}, d)$.

Also note that the $S_i(\mathcal{C}', d)$ will remain unchanged for the points in the clusters that are same in both clusterings \mathcal{C} and \mathcal{C}' . Thus there are atleast $(n_{C_r^*} + n_{C_r^{\dagger}})$ in total of $S_i(\mathcal{C}', d)$ less than $S_i(\mathcal{C}, d)$ which will make $\bar{S}(\mathcal{C}', d)$ strictly less than 0.5. To recapitulate, for some of the points $S_i(\mathcal{C}', d)$ will remain same as for $S_i(\mathcal{C}, d)$, for the point that changed cluster membership it's -0.5 and for some other points $S_i(\mathcal{C}, d) < 0.5$, thus clearly $\bar{S}(\mathcal{C}', d) < \bar{S}(\mathcal{C}, d)$. \square

We have learnt few things from Lemma 1 which are given below as remarks while keep in mind that all the set up remain same as in lemma. These remarks will provide more insight for the Case 1 (a) presented in the Figure 5.2 for the Theorem 4.

Remark 1. *It is obvious that if one of the $S_i(\mathcal{C}', d)$'s will reduce from 0.5 and all other remains 0.5 the ASW will reduce from 0.5. We have seen that as one point changes its cluster membership this will adversely affect $S_i(\mathcal{C}', d)$'s of all i's in these two clusters involved. Since \mathcal{C} and \mathcal{C}' are two unique clusterings, to prove \mathcal{C} is a rich CQM it is enough to show only for one $i \in \mathcal{C}'$ that $S_i(\mathcal{C}', d) < 0.5$ while others remain at 0.5. This can be attained by only considering one pair of points that is together in \mathcal{C} but not in \mathcal{C}' . As the number of such pairs will increase, this will cause bigger reduction in $\bar{S}(\mathcal{C}', d)$ as compared to just one pair. Since \mathcal{C}' is a different clustering than \mathcal{C} , in fact there can be me more such pair of points that were in some clusters in \mathcal{C} but are in some other clusters in \mathcal{C}' . Of course the situation in Lemma 1 can be generalized to many pair of points that are together in \mathcal{C} but not in \mathcal{C}' which is done in the proof of theorem 4.*

Remark 2. *Suppose that \mathcal{C} is a clustering such that some point i belongs to some cluster C_r of it. Suppose that this clusterings is based on distance d such that all within cluster distances are 1 and all between cluster distances are 2. Since all the clusters are equally distance from each other by definition of d therefore, all clusters other than C_r are closest neighbours of C_r and any one can provide $b(i)$. But if a point moves out of a cluster C_r to some cluster $C_{r^*} \in \mathcal{C}$ then a) C_{r^*} will provide the $b(i)$ value for all the points in C_r i.e., C_{r^*} will become the closest cluster to cluster C_r . b) C_r will provide the $b(i)$ value for the point i that moved out of C_r .*

Remark 3. *Suppose that \mathcal{C} is a clustering such that the point i belongs to some cluster C_{r^*} of it. Suppose that this clusterings is based on distance d such that all within cluster distances are 1 and all between cluster distances are 2. Suppose point $i \in C_{r^*}$ now changes its cluster membership and $i \notin C_{r^*}$ but $i \in C_{r^{\dagger}}$. As a result to the point i changing its cluster membership the following three things will happen*

- (i) *The $b(j)$ for all $j \in C_{r^*}$ will change (decrease) and $a(j)$ will remain unaffected*
- (ii) *The $a(j)$ for all $j \in C_{r^{\dagger}}$ will change (increase) and $b(j)$ will remain unaffected*

- (iii) For the point i , $a(i)$ and $b(i)$ both will change, in fact their values will be interchanged.

For the generalization of the case presented in Lemma 1 we have to think that there can be more pairs of points that are not together in \mathcal{C}' but were together in \mathcal{C} . If this will happen, the values of $a'(i)$ and $b'(i)$ can change together unlike in Lemma 1. This is because various pair of points in various clusters \mathcal{C}' are not together now affecting both $a'(i)$ and $b'(i)$ simultaneously. We now give the proof of the Theorem 4.

Proof of Theorem 4. In order to prove that the ASW is a rich CQM, we need to consider every possible non-trivial clustering \mathcal{C} and construct a distance function d for it such that no other clustering \mathcal{C}' is a better opponent i.e., no other clustering can give improved ASW value beyond \mathcal{C} . There exist two possibilities for the clustering \mathcal{C} to consider in this proof. To prove the theorem for each of these possibilities we will divide the proof in two cases. We define the cases now. Case 1: \mathcal{C} is a non-trivial clustering where all clusters have more than one object, Case 2: \mathcal{C} is a clustering where there is at least one one-point cluster.

Case 1: Given \mathcal{C} , construct a distance function d such that $d(x_i, x_i) = 0$, $d(x_i, x_j) = 1$ if $x_i \sim_{\mathcal{C}} x_j$, and $i \neq j$, and $d(x_i, x_j) = 2$ if $x_i \not\sim_{\mathcal{C}} x_j$ for all $i, j \in \mathcal{X}_n$. Note that for all points $x_i \in \mathcal{X}$, $a(i) = \frac{(n_{r^*}-1) \times 1}{(n_{r^*}-1)} = 1$ and $b(i) = \frac{(n_{r^+}) \times 2}{n_{r^+}} = 2$, where n_{r^*} and n_{r^+} is the number of objects in some clusters C_{r^*} and C_{r^+} of \mathcal{C} . This gives $S_i(\mathcal{C}, d) = 0.5$ for all $i \in \mathcal{X}$, such that $\bar{S}(\mathcal{C}, d) = 0.5$. We claim that this is the only optimal clustering of \mathcal{X} and for any other clustering \mathcal{C}' of \mathcal{X} , $\bar{S}(\mathcal{C}', d)$ will be smaller than $\bar{S}(\mathcal{C}, d)$, which we show now.

There is more than one possibility for clusterings \mathcal{C}' . First assume that \mathcal{C}' is a clustering in which there is no single point cluster. Note that since \mathcal{C}' is some clustering other than \mathcal{C} therefore some points in some clusters of \mathcal{C} will now be in some other clusters in \mathcal{C}' . Because of the points that are now in different clusters the $\bar{S}(\mathcal{C}', d)$ will reduce from 0.5 due to the change in the $S_i(\mathcal{C}', d)$, which we will show now.

Since \mathcal{C}' is some other clustering than \mathcal{C} , one of the following two possibilities must hold: (a) there is at least one pair of points that are together in \mathcal{C} but not in \mathcal{C}' or (b) there is at least one pair of points that are not together in \mathcal{C} but in \mathcal{C}' . First, due to the definition of d , in \mathcal{C} the distances can be either 1 or 2, which implies that $1 \leq a'(i) \leq 2$ and $1 \leq b'(i) \leq 2$ for any $i \in \mathcal{C}'$. This implies $S_i(\mathcal{C}', d) \leq 0.5$ for any $i \in \mathcal{C}'$. This will hold for both (a) and (b) as shown now. Consider (a) now. There are two possibilities for $S_i(\mathcal{C}, d)$ for all $i \in \mathcal{C}'$ which are (a.1) for $i \in \mathcal{C}'$, $S_i(\mathcal{C}, d)$ can be either less or (a.2) equal to 0.5. While these two conditions may or may not occur together or just (a.1) can occur for all i 's but note that only (a.2) can't occur for all $i \in \mathcal{C}'$. This is because of (a). Now under (a.1) there will be at least one i in \mathcal{C}' for which $d(x_i, x_j) = 2$, for $x_i \sim_{\mathcal{C}} x_j$ because there is at least one such pair that is not together in \mathcal{C}' but was in \mathcal{C} . Now since all the within cluster distances are either 1 or 2 in \mathcal{C}' , $1 < a'(i) \leq 2$. Next we look at $b'(i)$ for these $i \in \mathcal{C}'$. Note that there is at least one i for which $d(x_i, x_j) = 1$,

for $x_i \sim_C x_j$, which implies $1 \leq b'(i) < 2$. Note that $\max\{a'(i), b'(i)\}$ can be either $a'(i)$ or $b'(i)$ but will be only from $[1, 2]$. Since $b'(i) \neq 2$ but less than it and $a'(i) \neq 1$ but greater than 1 for at least one i in \mathcal{C}' therefore, $b'(i) < b(i)$ and $a'(i) > a(i)$. Under (a.2) which is to include the possibility some clusters remain same in both clusterings \mathcal{C} and \mathcal{C}' , for such i 's $S_i(\mathcal{C}', d) = S_i(\mathcal{C}, d)$. Hence for at least one i , $S_i(\mathcal{C}', d) < S_i(\mathcal{C}, d)$ and for no $i \in \mathcal{C}'$, $S_i(\mathcal{C}', d)$ can be greater than 0.5 which implies that $\bar{S}(\mathcal{C}', d) < \bar{S}(\mathcal{C}, d)$.

Next assume that \mathcal{C}' is a clustering such that there is at least one single point cluster in it. Let the member of a one-point cluster be denoted by i , such that $S_i(\mathcal{C}', d) = 0$ by definition. Also for the remaining $(n - 1)$ points, $S_j(\mathcal{C}', d) \leq 0.5$. Denote $S_j(\mathcal{C}', d) = v_j$ where $-1 \leq v_j \leq 0.5$. This will lead to $\bar{S}(\mathcal{C}', d) = \frac{\sum_{j=1}^{(n-1)} v_j}{n} < 0.5$.

Case 2: There is at least single one-point cluster in the clustering \mathcal{C} . Construct a distance function d such that $d(x_i, x_i) = 0$, $d(x_i, x_j) = 1$, if $x_i \sim_{\mathcal{C}} x_j$ and $d(x_i, x_j) = 2$, if $x_i \not\sim_{\mathcal{C}} x_j$. Let there be $t \in \mathbb{N}$, ($1 \leq t < n$) one-point clusters. Since there are t clusters which contain just one point, the ASW for this clustering will be $\bar{S}(\mathcal{C}, d) = \frac{\sum_{i=1}^n S_i(\mathcal{C}, d)}{n} = \frac{\sum_{i=1}^t S_i(\mathcal{C}, d) + \sum_{i=(t+1)}^{(n-t)} S_i(\mathcal{C}, d)}{n} = \frac{t \times 0 + (n-t) \times 0.5}{n} = \frac{(n-t) \times 0.5}{n}$. We will now consider all the possible non-trivial clustering \mathcal{C}' and show that they will not give better value of ASW than $(n - t) \times 0.5/n$.

There are two possibilities to consider for \mathcal{C}' . As a first possibility assume that \mathcal{C}' is such a clustering that there is no one-point cluster in it i.e., all clusters have more than one point. In such a situation t one-point clusters have merged into other clusters. The $S_i(\mathcal{C}', d)$ for the points that were forming one-point cluster will remain 0 even if they now move to other clusters. This is because now $b'(i) = a'(i)$ for these points. For these points $b'(i) = b(i) = 2$ but now $a'(i) = 2$ instead of 1. In the other hand the clusters which got the points that were previously one-point clusters the $a'(i)$'s for the remaining values in these clusters will increase. This is because these clusters have now at least one such pair of points that has $d(x_i, x_j) = 2$, if $x_i \sim_{\mathcal{C}} x_j$ which were all previously 1. Hence $b'(i) - a'(i) < b(i) - a(i)$ where i represents the index for the points that are in those clusters that are merged with one-point clusters and $S_i(\mathcal{C}', d)$ cannot become better for any point. Therefore, it is clear that $\bar{S}(\mathcal{C}', d) < \bar{S}(\mathcal{C}, d)$.

Note that \mathcal{C}' can be also a clustering such that there is at least single one-point cluster in it, but the number of single point clusters in \mathcal{C}' is smaller than the number of single point clusters in \mathcal{C} . In such a situation there will be $t^* < t$ clusters in \mathcal{C}' that have been merged into other clusters. For such a case the same logic given in previous paragraph holds.

As a second possibility assume that the number of one-point clusters in \mathcal{C}' is greater than the number of one-point clusters in \mathcal{C} . Let there are $t \in \mathbb{N}$, for $t < n$ one-point clusters in \mathcal{C} and $t^\dagger \in \mathbb{N}$, for $(t^\dagger < n, t^\dagger > t)$. In such a situation there will be t^\dagger points in \mathcal{C}' for which $S_i(\mathcal{C}', d) = 0$ such that $\bar{S}(\mathcal{C}', d) = \frac{\sum_{j=1}^{(n-t^\dagger)} v_j}{n}$, which will be always less than

$\bar{S}(\mathcal{C}, d) = \frac{(n-t) \times 0.5}{n}$ due to $t^\dagger > t$ and no $S_i(\mathcal{C}', d)$ can become better than 0.5. ³

□

³In addition note that there can be more data points $i \in \mathcal{C}'$ for which $S_i(\mathcal{C}', d) = 0$, this is because even if the single point clusters of \mathcal{C} are amalgamate with other non-singleton clusters in \mathcal{C}' their $S_i(\mathcal{C}', d) = 0$ because for such i 's $a'(i) = b'(i) = 2$.

Chapter 6

Future aspects

The objective of the current thesis was to develop the clustering methods based on the optimization of the ASW index. We have developed the theory, methodology and algorithms for the proposed methods. We began with proposing a clustering algorithm named HOSil in hierarchical setting in Chapter 3 and have thoroughly investigated it against various clustering scenarios for (a) measuring clustering quality, (b) validated this clustering using external validation index—ARI, and (c) for the estimation of number of clusters. Alongside we have investigated the performance of ASW index for measuring the clustering quality and for the estimation of number of clusters. This comparison was conducted against many internal indices and clustering methods. The proposed algorithm turns out to be computationally expensive. We have proposed a fast version of this by designing a methodology that make use of both partitional and hierarchical schemes.

We have developed a second coherent clustering method by proposing an algorithm OSil for the optimization of the ASW index in non-hierarchical setting in Chapter 4. This algorithm needs an initial clustering to start the optimization process. The algorithm's performance was learned by initializing it using several clustering algorithms. We then proposed a final version of the algorithm and worked our way through the development of the fast version FOSil for optimum ASW clustering.

The proposed versions has been applied to the real life applications. We have also provided a small study to give an insight about the effect of various distance metric on the methods used in the study.

Apart from validating the quality delivered by ASW index empirically, we took the approach of the axiomatic theory developed and proved that the index satisfies the three properties namely, scale-invariance, consistency and richness proposed in Kleinberg (2003).

The work presented here can be seen as first step towards developing OASW based clustering methods. There are many ways in which these algorithms can be improved, modified for different applications or extended to other domains. Improvement of the

clustering algorithms particularly to make them scalable to large dataset with the best approximation possible has always been of interest for clustering analysis community. In the following subsections a few suggestions are made to improve the work presented in Chapter 3, 4, and 5.

6.1 An alternative HOSil algorithm suggestion

In this section we will suggest an algorithm for the fast calculation of agglomerative hierarchical clustering to optimize the ASW. In the clustering literature one practice to scale algorithms to large data sets is to take several samples of the data and to optimize the objective function based on the sampled observations only. The algorithm proposed here also makes use of this approach.

Consider a data set having n observations. Take a random sample from data set of size $v/n * 100$. Selection of v depends upon factors such as the computational power of the processors available, the size of the data and the number of clusters. We recommend to use v between 150 and 300 based on our experience from HOSil. Use the HOSil clustering algorithm presented in Section 3.2 to cluster the sample of data. The difference is that at each hierarchy level first a proportion of data is clustered only, then assign the remaining data points to these clusters by optimizing the SW of clustering for each observation. For instance after the clustering is obtained for a proportion of data, take an observation from the remaining data and try putting it in every cluster and calculate resulting ASW. Assign the observation to the cluster which gave maximum ASW for all. Complete the assignment of all observations to get a clustering for the whole data at this hierarchy level. Now for the next hierarchy level take a sample from each cluster such that the overall sample size is $v/n * 100$. Any sampling scheme can be used that gives a representative sample, for instance simple proportional allocation sampling scheme or probability proportional to size etc. We don't recommend to use the clustering here which was obtained from the sample from the previous hierarchy level before finding the complete clustering. Taking a sample from the clustering obtain from entire data set at previous hierarchy will give more accurate clustering as compared to the clustering on the sample data only. Again decide which two (or more) clusters should be combined at this hierarchy level based on HOSil for this sample. Obtain the clustering for the whole data at this hierarchy level as described earlier. Repeatedly complete all the hierarchy level from bottom to top.

6.2 OSil further improvement and extensions

Approximation algorithms to reduce computational burden The OSil₁ algorithms proposed in Chapter 3 was computationally expensive therefore, its fast version was

proposed which has reduced the computational cost of OSil₁. However, other ways of implementation of the OSil₁ algorithm to make it computational faster can be explored. One possible direction in this line of work could be to optimize the ASW for clustering using simulating annealing algorithms. As mentioned already clustering is a combinatorial optimization problem that finds the optima of functions against discrete variables. Broadly speaking the combinatorial optimization problem can be solved using optimization algortihms or approximation algorithms Merendino and Celebi (2013). The optimization algorithms are good option in theory because they return the best solution but they can be hard to implement and computational slow or prohibitive in practice. An alternative way to solve the combinatorial optimization problem is the approximation algorithms also known as heuristics. One popular choice among these is the simulating annealing algorithms (Metropolis et al. (1953)). Adaptation of these algorithms for OASW clustering can decrease the computational time and will lead to algorithm(s) applicable to big data sets. The computational complexity burden of the optimization algorithm proposed in this work can be reduced by taking the simulated annealing approach.

Another appraoch is to go for the stochastic optimization schemes. For instance, the evolutionary algorithms (Hruschka et al. (2009)) or otherwise known as the genetic algorithms (Lucasius et al. (1993), Davis (1991), Maulik and Bandyopadhyay (2000)) can be also used to get fast approximation but they are not primarily known for this. They are good in avoiding local optima and also give the global or near global optimum. Although the algorithm proposed in this work were able to get higher ASW quality for clustering for most of the clustering structures than their competitors but this quality was merely a local optimum. Development of the genetic algorithm for OASW clustering will further improve the clustering quality and can offer a global optimum solution. These algorithms are slow but there has been work in literature to speed them up, for instance Sheng and Liu (2006) proposed a genetic k -medoids algorithm (see also Lucasius et al. (1993)).

How will the results of OSil/HOSil clustering change for various distance metrics? This questions has been touched very briefly in this work. As concluded in Section 4.16 the distance metric performance is dependent on the clustering structures. More research is needed in order to explore the behaviour for other clustering structures and other distance metrics. This could also be then expended to other data types that are categorical or mix type (Huang (1998)), and data coming from spaces other than Euclidean.

Effect of outliers and noise on OASW clustering The performance of the algorithms proposed in this work will be greatly affected by the presence of noise or outliers in the data. Depending upon the type of noise or outliers in the data, the clustering results can completely change. For instance for k -means and many other algorithm several heterogeneous clusters can be combined and isolated outliers can form separate clus-

ters. The presence of outliers can also affect greatly the estimation of the number of clusters performance of the algorithms. The algorithms proposed in this dissertation are not robust to outliers. For instance, Figure 6.1 shows the clustering results of the OSil₁ algorithm for three different types of outliers (single outlier, multiple clustered outliers, uniform noise) introduced in Model 1 defined in Chapter 4. This data has two Gaussian clusters. Panel (a) in the figure is an OSil₁ clustering result with estimated k when a single outlier (shown in +) is added to the data. OSil₁ has estimated 3 clusters here and formed a third cluster with just one point, which is the outlier. Panel (b) shows the clustering result with fixed known k for this data set. Panels (c) represent data clustering when 5 outliers are added to the data (shown in +) with estimated number of clusters from OSil₁. Panel (d) represent the results for the same data for the known number of clusters from OSil₁. The distance of the outliers/noise from the clusters also affects the clustering results. Panels (e) and (f) show the one outlier and multiple outliers added to the data further away from the clusters as compared to panel (a) and (c), respectively. The clustering output of OSil₁ has changed. It has now estimated 2 number of clusters instead of 3. Lastly, panels (g) and (h) represent the clustering results for estimated and known number of clusters when noise generated from Uniform distribution is added to the data. Exploring the ways to make the algorithm proposed in this work robust is a worthwhile project to deal real life data sets that has noise. For instance the scanned brain images taken from fMRI machines. In this respect there is a good amount of literature available on the robust cluster analysis, for instance see [Coretto and Hennig \(2016\)](#) and [de Amorim and Hennig \(2015\)](#).

Missing data handling In real life applications missing data also known as incomplete data is a very common problem. There can be situations where there is some data missing for instance, it could be that some variables have no observations for a few objects. Missing data might occur due to various reasons, for instance, for DNA microarray this might be due to scratches on the slide. The clustering methods need a full data matrix to cluster. A common way to deal with missing data is to use imputation methods. For cluster analysis this can be done as a pre-processing step. However, the analysis based on simply replacing the missing values with row mean is not reliable. Therefore, more sophisticated solutions has been developed, for instance kNN ([Troyanskaya et al. \(2001\)](#)), or model-based ([Ghahramani and Jordan \(1995\)](#)) approaches for imputation. Although the imputation methods will create new data values to fill the missing values, however analysis based on the newly generated estimated data will suffer from the reliability issue as these data values will not exist in real (see [Troyanskaya et al. \(2001\)](#)). An alternative clustering method that does not create new data values was proposed in [Wagstaff \(2004\)](#). They proposed a k-means clustering method for missing values for multiple feature that do not make use of imputation methods. It is of real interest to research further in this regard and develop missing data handling solutions for the methods developed in this thesis.

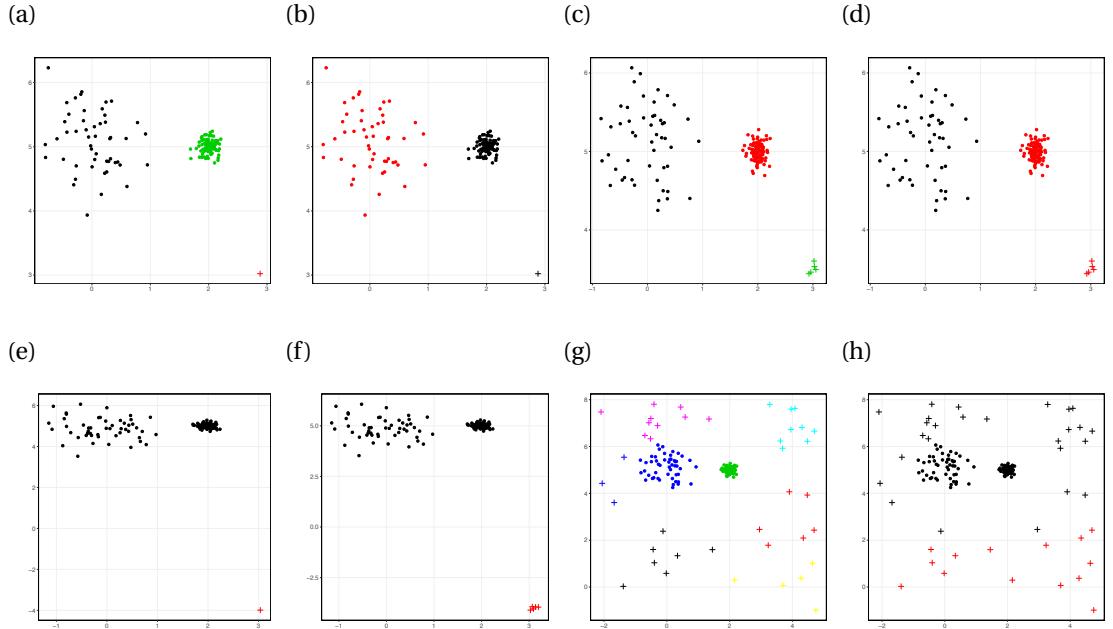


Figure 6.1 OSil₁ clustering results for various kinds of outlier/noise in the data. The plus symbol represents the outliers and colours represent clusters. For a complete description of the figure see Section 6.2 last paragraph.

6.3 Future theory development

There is a huge scope of work in the direction of theory development in line with what we have proposed in Chapter 5. In this work we have proved the set of axioms for the ASW as a clustering quality measure and have not investigated the clustering functions and algorithms based on optimization of this index.

The first task in this regard is to investigate which of the set of axioms the clustering functions and algorithms proposed in this work will satisfy i.e., the taxonomy of the clustering functions and algorithms based on OASW for hierarchical and non-hierarchical versions can be developed. In Chapter 3 we have developed a new linkage criterion for clustering based on the ASW index and an algorithm to implement this. Ackerman et al. (2010a) have proposed a characterization of linkage-based hierarchical clustering methods following their work in Ben-David and Ackerman (2009). Following their strategy, one could characterize the linkage based clustering function proposed in this work.

Ackerman et al. (2010b) have proposed a set of properties for general clustering functions by extending their work in Ben-David and Ackerman (2009). The properties proposed there are isomorphic invariance, scale invariance, consistency, and richness.

They have proposed other properties related to scale invariance, consistency, and richness. For instance, the inner and outer consistent properties for clustering functions by introducing relaxation on the consistency axiom. If a function satisfies consistent it is both inner and outer consistent, but if not one can check separately which property from inner and outer consistency the function fulfils. It seems that the clustering functions based on ASW will also fulfil these properties because ASW fulfils them. One property proposed there is locality, which loosely means that a clustering function depends upon within cluster distances only. ASW on the other hand also takes into account between cluster distance (in the definition of $b(i)$), and therefore it might not satisfy this property.

Another useful extension to this work could be the generalization of OASW clustering methods to weighted data settings. Among the nine admissibility properties of the [Fisher and Ness \(1971\)](#) there is the following property which is related to weighted clustering settings:

Definition 6.3.1. Point Proportion Admissibility: A clustering procedure is point proportion admissible if after duplicating one or more points the clusters' boundaries do not change.

[Margareta Ackerman and Loker \(2012\)](#) have build on these and have introduced three notions in weighted settings i.e., weight sensitivity, weight robustness and weight considering. Loosely speaking, weight robust clustering methods are those that are not affected by weights on the data object and for all kind of weights they yield the same clustering output. The weight sensitive clustering algorithms are those that can yield different clusterings on data if different weights for the objects are applied. This property is opposite to the point proportional admissible property of [Fisher and Ness \(1971\)](#). Finally, the weight considering clustering algorithms are those for which the different weights may or may not affect their output, i.e., there will be some clusterings by algorithms that will be weight sensitive whereas some others will not be weight sensitive. We now give formal definitions analogous to CQMS in weighted settings.

Definition 6.3.2. Weight responsiveness: A CQM Π is weight responsive on a clustering \mathcal{C} of (\mathcal{X}, d) if

- (i) there exists a weight function w so that $\Pi(\mathcal{C}, (w[\mathcal{X}], d)) = \Pi(\mathcal{C}, (\mathcal{X}, d))$, and
- (ii) there exists a weight function w' so that $\Pi(\mathcal{C}, (w'[\mathcal{X}], d)) \neq \Pi(\mathcal{C}, (\mathcal{X}, d))$

Definition 6.3.3. Weight sensitivity: A CQM Π is weight sensitive if for all (\mathcal{X}, d) and all $\mathcal{C} \in \mathcal{S}(\mathcal{X})$, Π is weight responsive on \mathcal{C} .

Definition 6.3.4. Weight robustness: A CQM Π is weight robust if for all (\mathcal{X}, d) and all $\mathcal{C} \in \mathcal{S}(\mathcal{X})$, Π is not weight responsive on \mathcal{C} .

Definition 6.3.5. Weight considering: A CQM Π is weight considering if

- (i) There exist a (\mathcal{X}, d) and a clustering \mathcal{C} of (\mathcal{X}, d) so that Π is weight responsive on \mathcal{C}
- (ii) There exist a (\mathcal{X}, d) and $\mathcal{C} \in \mathcal{S}(\mathcal{X})$ so that Π is not weight responsive on \mathcal{C}

An interesting extension for the theory development for ASW and clustering functions defined by its optimization is to extend them to weighted clustering setting. [Margareta Ackerman and Loker \(2012\)](#) have shown that many partitional methods including k -means and PAM are weight sensitive. As ASW is not a robust index, our intuition is that clustering functions based on optimum ASW will not be weight robust and will be affected by some kind of weights in some sense and so does the clustering algorithms based on it. But under which of these weighted definition it will exactly fall can't be sensed without proper investigation. Probably ASW will not be weight sensitive but weight considering.

Another, extension in this regard would be the development of the general clustering axioms for robust clustering functions and testing them for ASW index and clustering functions and algorithms based on it.

Consider the following two properties from the set of admissibility properties of [Fisher and Ness \(1971\)](#).

Definition 6.3.6. Cluster Proportion - Admissibility: A clustering procedure is cluster proportion admissible if after replicating each point within the same cluster, the same number of times, the clusters' boundaries do not change.

Definition 6.3.7. Cluster Omission - Admissibility: A procedure is cluster omission admissible if in the case that all points in any one of the k - clusters say C_h , are removed from \mathcal{X} , the procedure gives the original clusters except for C_h when applied to the subset $\mathcal{X} - C_h$ to get $k - 1$ clusters.

Further in this line one can also try to prove the above two properties of [Fisher and Ness \(1971\)](#).

Yet another line of work very useful in this regard would be translating the [Margareta Ackerman and Loker \(2012\)](#) properties plus the above two properties to CQMs. These analogous properties for CQM can tell more about the behaviour of the ASW index and can be extended to other CQMs.

Appendix A

Statistical distributions for data generation

All the continuous probability distributions used to generate synthetic datasets are defined in this section

The multivariate Gaussian distribution

Let $X_j \in \mathbb{R}^p$ for $p \geq 1$ be i.i.d. random variates. Let $\psi_X(x; \mu, \Sigma)$ represent the probability density function (pdf) of these p variates where $\mu \in \mathbb{R}^p$ represents the mean vector and Σ be $p \times p$ covariance matrix for these variates. Let $x_i; i = 1, \dots, n$ be the i.i.d. sample realization for random variables X_i . The p -variate Gaussian distribution can be defined as:

$$\psi_X(x; \mu, \Sigma) = \frac{1}{2\pi^{p/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^t \Sigma^{-1} (x - \mu)\right), \quad x \in (-\infty, +\infty),$$

where $-\infty \leq \mu \leq +\infty$, $\sigma > 0$ and

$$\Sigma = \begin{bmatrix} Cov(X_1, X_1) & Cov(X_1, X_2) & \dots & Cov(X_1, X_p) \\ Cov(X_2, X_1) & Cov(X_2, X_2) & \dots & Cov(X_2, X_p) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_p, X_1) & Cov(X_p, X_2) & \dots & Cov(X_p, X_p) \end{bmatrix}.$$

The covariance between two variables can be further written as $Cov(X_j, X_{j'}) = \sigma_{jj'} = \rho_{jj'} \sigma_j \sigma_{j'}$, where $\rho_{jj'}$ denotes the correlation between two variables j and j' . We will use $X_p \sim N_p(\mu_p, \Sigma_{p \times p})$ to denote Gaussian random variates.

The skew Gaussian distribution

Let $\varphi(Z)$ be a continuous random variable with pdf on \mathbb{R}^p defined as follows:

$$f(z; \alpha) = 2\varphi(z)\Phi(\alpha z), \quad -\infty < z < \infty$$

where α is the shape parameter and $\varphi(z)$ is a standard normal pdf given as follows:

$$\varphi(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}, -\infty < z < \infty$$

and $\Phi(z)$ is the cumulative distribution function given as :

$$\Phi(z) = \int_{-\infty}^{\alpha z} \varphi(t) dt.$$

The variable $X = \zeta + \omega Z$, ($\zeta \in \mathcal{R}, \omega \in \mathbb{R}^+$), will be a skew-normal variate with the location parameter ζ , scale parameter ω and shape parameter α . In case of extended skew normal distribution an additional parameter τ is used to define the hidden mean of distribution. We will use $X \sim SN(\zeta, \omega, \alpha, \tau)$ to represent a variable following a skew Gaussian distribution.

The non-central Chi-squared distribution

Let $X = \sum_{\eta=1}^r Y_\eta$, where Y_η be the r independent Gaussian variables with mean μ_η and variances σ_η^2 . Let λ be the sum of means of these r variates, i.e., $\lambda = \sum_{\eta=1}^r \mu_\eta$. Let $\varphi_X(x; \lambda, r)$ denotes the pdf of non-central Chi-squared distribution where λ is called the non-centrality parameter and r is the degree of freedom. The pdf can be defined as

$$\varphi_X(x; \lambda, r) = \frac{1}{2} e^{-(x+\lambda)/2} \left(\frac{x}{\lambda} \right)^{r/4-1/2} I_{r/2-1}(\sqrt{\lambda x}), \quad x \in [0, +\infty)$$

where $I_r(y)$ is the Bessel function given by

$$I_r(y) = (y/2)^r \sum_{m=0}^{\infty} \frac{(y^2/4)^m}{m! \Gamma(r+m+1)}.$$

We will use $X \sim \chi_r^2(\lambda)$ to specify a Chi-squared distribution for a random variable.

The Student's t-distribution Let $Y \stackrel{i.i.d.}{\sim} N(\mu, \sigma)$, then $X = \frac{(\bar{Y}-\mu)}{S_Y/\sqrt{n}}$ is a t - distributed random variable with $v = n - 1$ degrees of freedom. The pdf can be written as

$$f_X(x; v) = \frac{\Gamma(\frac{v+1}{2})}{\sqrt{v\pi}\Gamma(\frac{v}{2})} \left(1 + \frac{x^2}{v} \right)^{-\frac{v+1}{2}}, \quad x \in (-\infty, +\infty).$$

We will use $X \sim t_v$ to represent a random variable following a t-distribution.

The non-central t-distribution

Let Z be a standard normal variate: $Z \sim N(0, 1)$ and V be chi squared variate with r degrees of freedom : $V \sim \chi_r^2$, then, $X = \frac{Z+v}{\sqrt{V/r}}$, is a non-central t- distributed random variable with r degrees of freedom and v as a non-centrality parameter. We will use $X \sim t_r(v)$ to represent a random variable following a non-central t-distribution.

The continuous Uniform distribution

The continuous uniform distribution gives the constant probability to select any number from the continuous interval between a and b . The pdf for a random variable $X \sim \mathbb{U}(a, b)$ can be written as

$$f(X; a, b) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{if } x < a \text{ or } x > b, \end{cases}$$

where $-\infty < a < b < +\infty$.

The Gamma distribution

Let X be a random variable following a Gamma distribution and α, β be the shape and rate parameters of the distribution. The pdf can be written as

$$f_X(x; \alpha, \beta) = \frac{\beta x^{\alpha-1} e^{-x\beta}}{\Gamma(\alpha)}; \quad x \in (0, +\infty), \alpha, \beta > 0,$$

where $\Gamma(\alpha)$ is the Gamma function. We will use $X \sim \text{Gam}(\alpha, \beta)$ to denote a random variable coming from Gamma distribution.

The Beta distribution

Let X be a random variable following a beta distribution, then the pdf can be written as

$$f_X(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{(\alpha-1)} (1-x)^{(\beta-1)}; \quad x \in (0, 1),$$

where $B(\alpha, \beta)$ is beta function defined as $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$. The real quantities $\alpha > 0$ and $\beta > 0$ are the two shape parameters of the distribution. We will denote $X \sim \text{Beta}(\alpha, \beta)$ to represent a random variable following Beta distribution.

The non-central Beta distribution

Let $X_1 \sim \chi_r^2(\lambda)$ and $X_2 \sim \chi_m^2$, then the variable $X = \frac{\chi_r^2(\lambda)}{\chi_r^2(\lambda) + \chi_m^2}$ follows the non-central Beta distribution of Type-1 with two real shape parameters $v_1, v_2 > 0$ and a real non-centrality parameter $\lambda \geq 0$. Here $v_1 = r/2$ and $v_2 = m/2$. We will use the notation $X \sim \text{NBeta}(v_1, v_2, \lambda)$ to represent a non-central Beta variate.

The Exponential distribution

Let X follows the exponential distribution with λ rate parameter. The pdf can be written as

$$f_X(x; \lambda) = \lambda e^{-\lambda x}; \quad x \in [0, \infty), \lambda > 0.$$

We will denote $X \sim \text{Exp}(\lambda)$ to represent a random variable following exponential distribution.

The non-central F-distribution

Let the random variables Y_1 follows the non-central Chi squared distribution with v_1 degrees of freedom and λ be mean and Y_2 following central Chi-squared distribution

with v_2 degrees of freedom, then the variable $X = \frac{Y_1/n_1}{Y_2/n_2}$ will follow non-central F distribution with (v_1, v_2) degrees of freedom. The pdf can be written as

$$g_X(x; v_1, v_2, \lambda) = \sum_{m=0}^{\infty} \frac{e^{-\lambda} \lambda^m}{m!} f_{v_1+2m, v_2}(x); \quad x \in (0, \infty), \quad v_1, v_2 \in (0, \infty), \quad \lambda \in [0, \infty),$$

where $f_{v_1+2m, v_2}(x)$ is the central F distribution with $(v_1 + 2m, v_2)$ degrees of freedom. For $x \in (0, \infty)$ the pdf for central F distribution is defined as

$$f_{v_1+2m, v_2}(x) = \frac{(v_1 + 2m)\Gamma((v_1 + 2m)/2 + v_2/2)}{v_2\Gamma((v_1 + 2m)/2)\Gamma(v_2/2)} \frac{\left(\frac{v_1+2m}{v_2}x\right)^{(v_1+2m)/2-1}}{\left(1 + \frac{v_1+2m}{v_2}x\right)^{(v_1+2m)/2+v_2/2}}.$$

v_1 and v_2 are the shape parameters of the distribution. λ is same as defined in Section A and is the non-centrality parameter of the distribution. We will denote $X \sim \mathbb{F}_{(v_1, v_2)}(\lambda)$ to represent a random variable following non-central F distribution.

The Weibull distribution

The pdf for a random variable following the Weibull distribution is defined as

$$f_X(x; \tau, \zeta) = \frac{\tau}{\zeta} \left(\frac{x}{\zeta}\right)^{\tau-1} e^{-(x/\zeta)^\tau}; \quad x \in [0, +\infty); \quad \zeta, \tau \in (0, +\infty).$$

τ is the shape and ζ is the scale parameter of the distribution. We will denote $X \sim \mathbb{W}(\tau, \zeta)$ to represent a random variable following Weibull distribution.

Appendix B

HOSil Algorithm results

This Appendix has two parts, each of which contains the results of Chapter 3. Appendix B.1 presents graphical clustering results for all the DGPs. For the DGPs used in simulation, clustering results for one iteration of the simulations are shown. Appendix B.2 reports the frequency counts for the estimation of number of clusters for the DGPs used in simulation for all the clustering methods and estimation methods used in the study.

B.1 HOSil clustering visualization

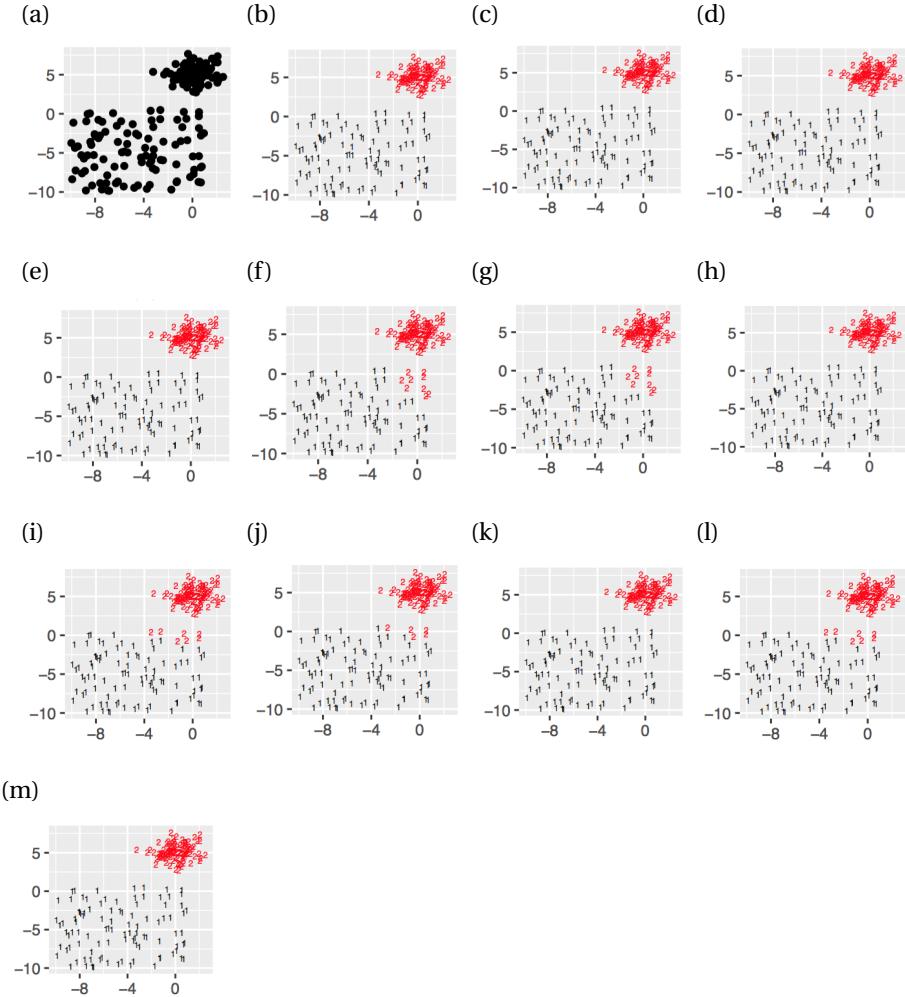


Figure B.1 Clustering results from clustering methods for a simulated dataset from Model 1, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering (i) PAM clustering (j) spectral clustering (k) model-based clustering (l) PAMSIL _{k} /PAMSIL _{\hat{k}} clustering, (m) HOSil _{k} /HOSil _{\hat{k}} clustering against true and estimated k .

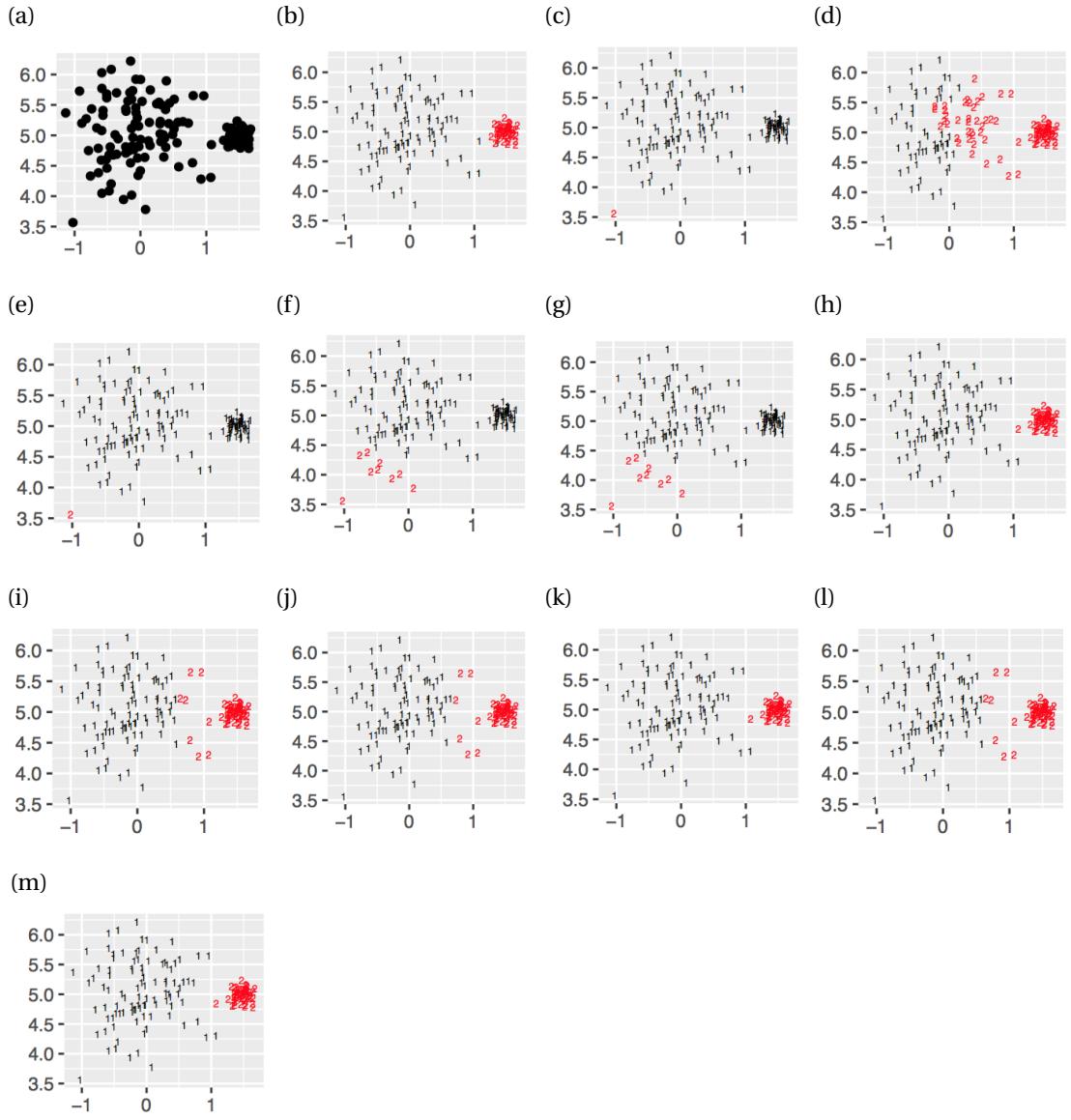


Figure B.2 Clustering results from clustering methods for a simulated dataset from Model 2, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering (i) PAM clustering (j) spectral clustering (k) model-based clustering (l) PAMSIL $_k$ /PAMSIL $_{\hat{k}}$ clustering, (m) HOSil $_k$ /HOSil $_{\hat{k}}$ clustering.

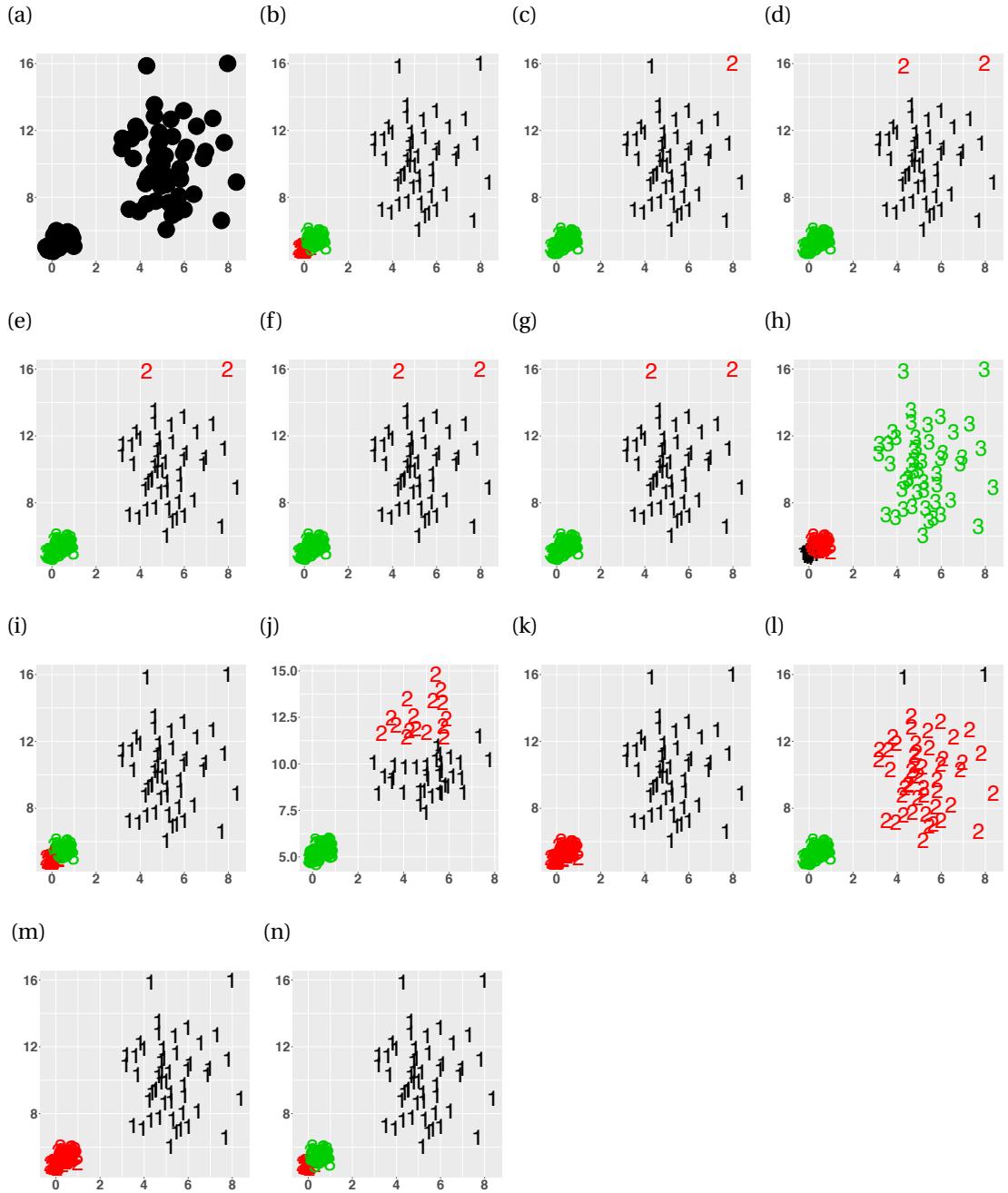


Figure B.3 Clustering results from clustering methods for a simulated dataset from Model 3, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering (i) PAM clustering (j) spectral clustering (k) model-based clustering (l) PAMSIL _{k} clustering, (m) HOSil _{k} clustering, (n) HOSil _{\hat{k}} and PAMSIL _{\hat{k}} clustering.

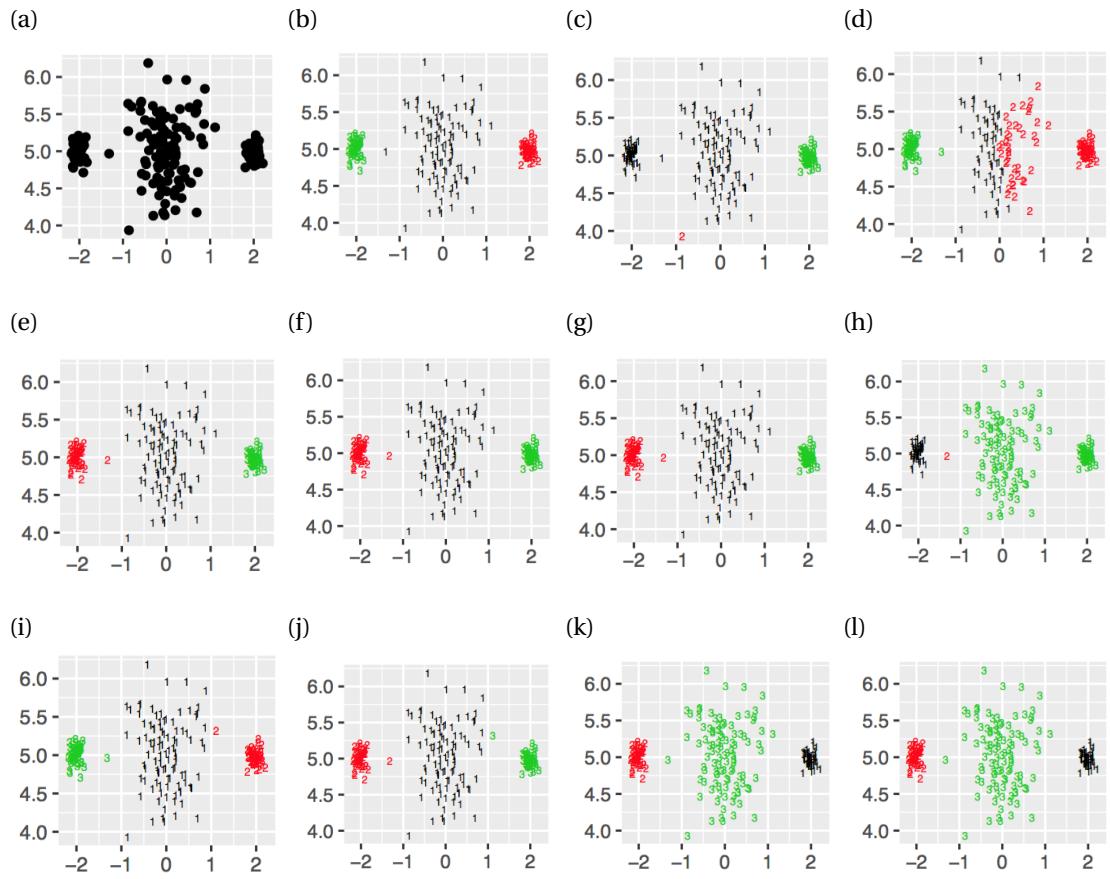


Figure B.4 Clustering results from clustering methods for a simulated dataset from Model 4, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering (i) PAM clustering (j) spectral clustering (k) model-based clustering (l) PAMSIL_k , $\text{PAMSIL}_{\hat{k}}$, HOSil_k , $\text{HOSil}_{\hat{k}}$ clusterings.

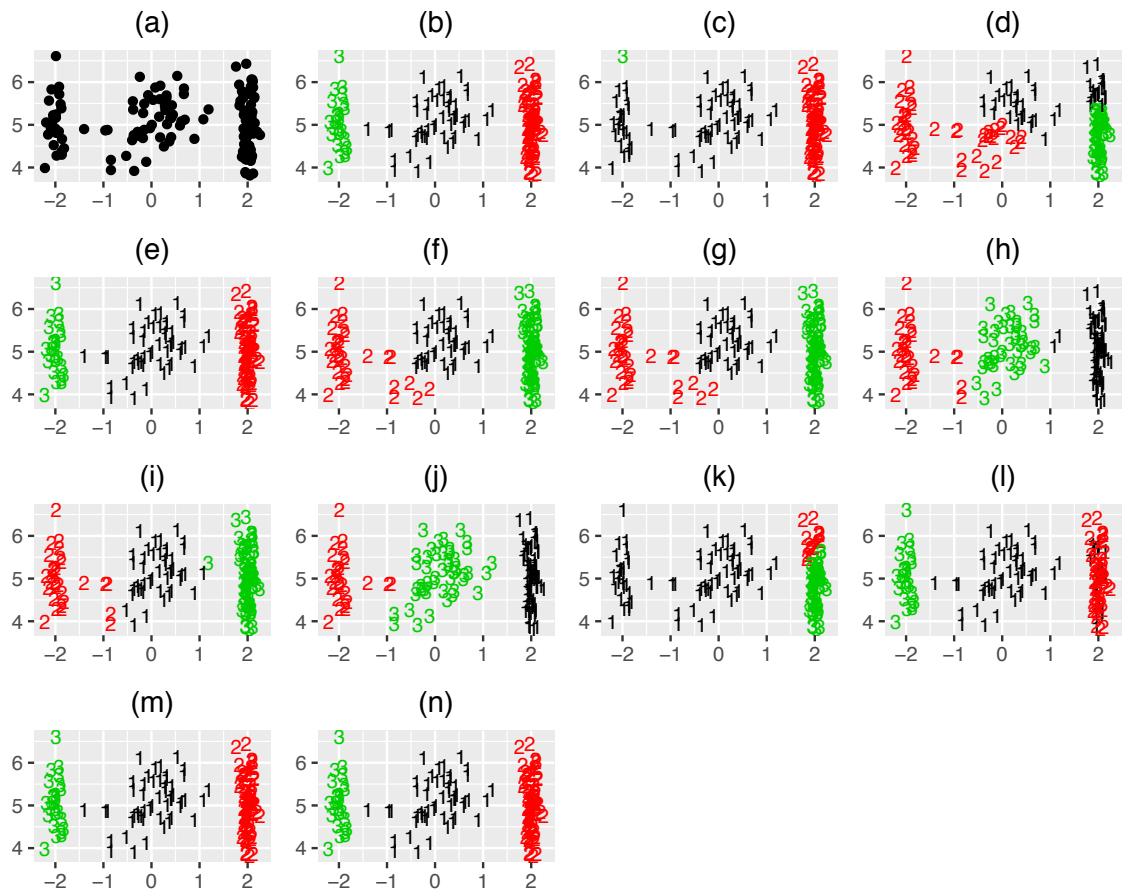


Figure B.5 Clustering results from clustering methods for a simulated dataset from Model 5, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means, (i) PAM, (j) spectral, (k) model-based, (l) PAMSIL $_k$ /PAMSIL \hat{k} , and (m) HOSil $_k$ /HOSil \hat{k} clusterings.

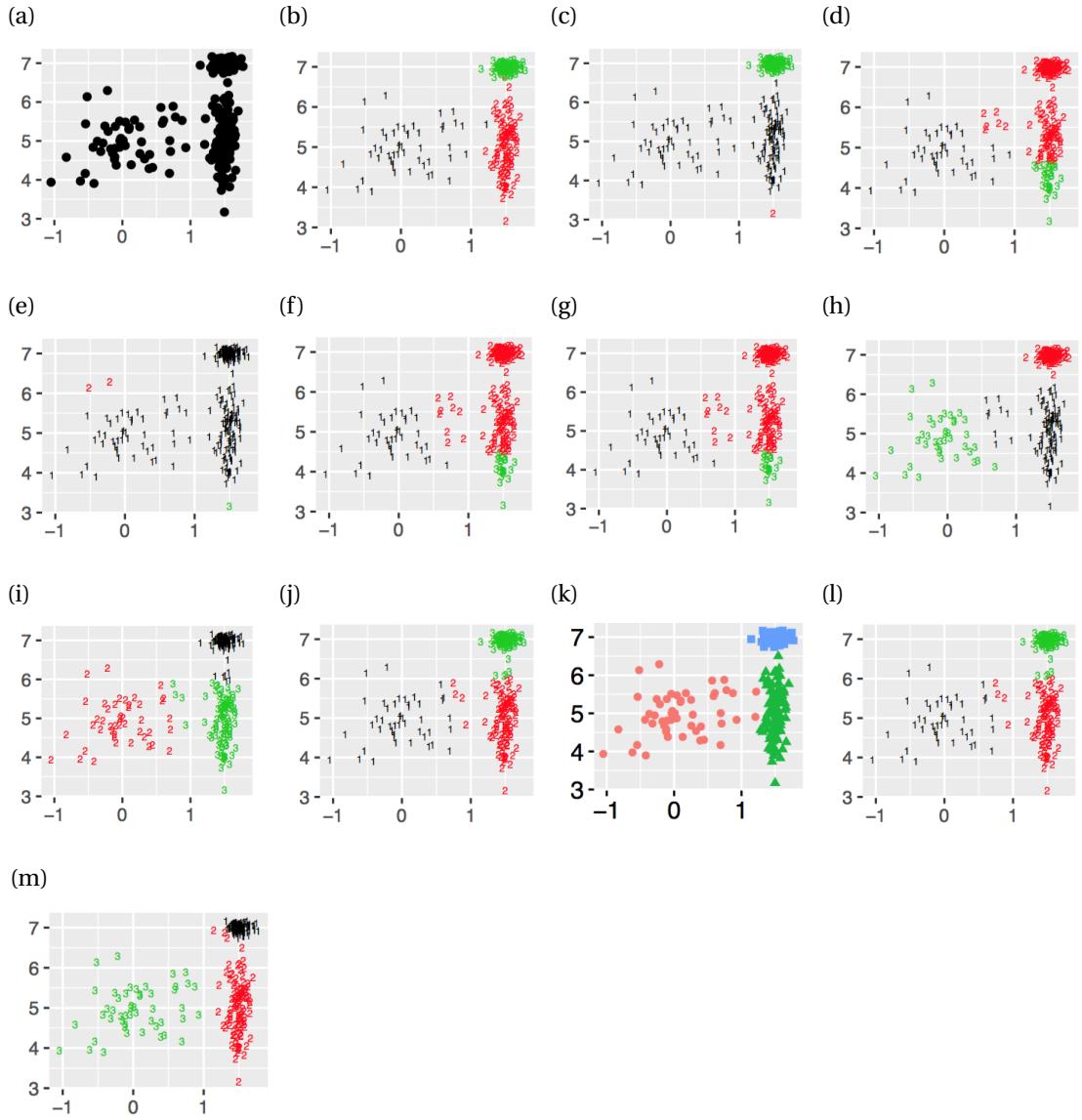


Figure B.6 Clustering results from clustering methods for a simulated dataset from Model 6, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering, (i) PAM clustering, (j) spectral clustering, (k) model-based clustering, (l) PAMSIL _{k} /PAMSIL _{\hat{k}} clustering, and (m) HOSil _{k} /HOSil _{\hat{k}} clustering.

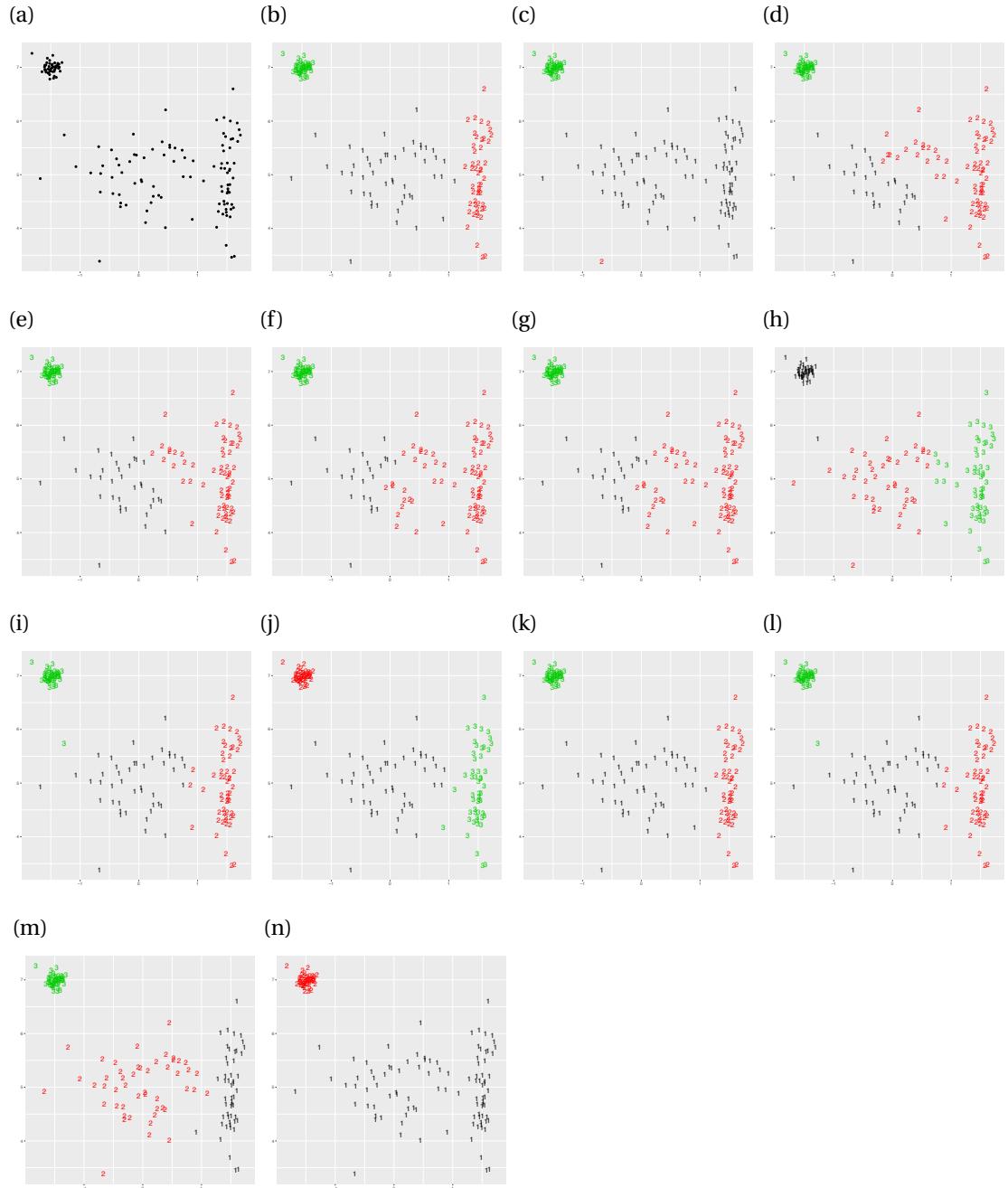


Figure B.7 Clustering results from clustering methods for a simulated dataset from Model 6.A, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering, (i) PAM clustering, (j) spectral clustering, (k) model-based clustering, (l) PAMSIL $_k$ clustering, (m) HOSil $_k$ clustering, (n) PAMSIL \hat{k} and HOSil \hat{k} clustering.

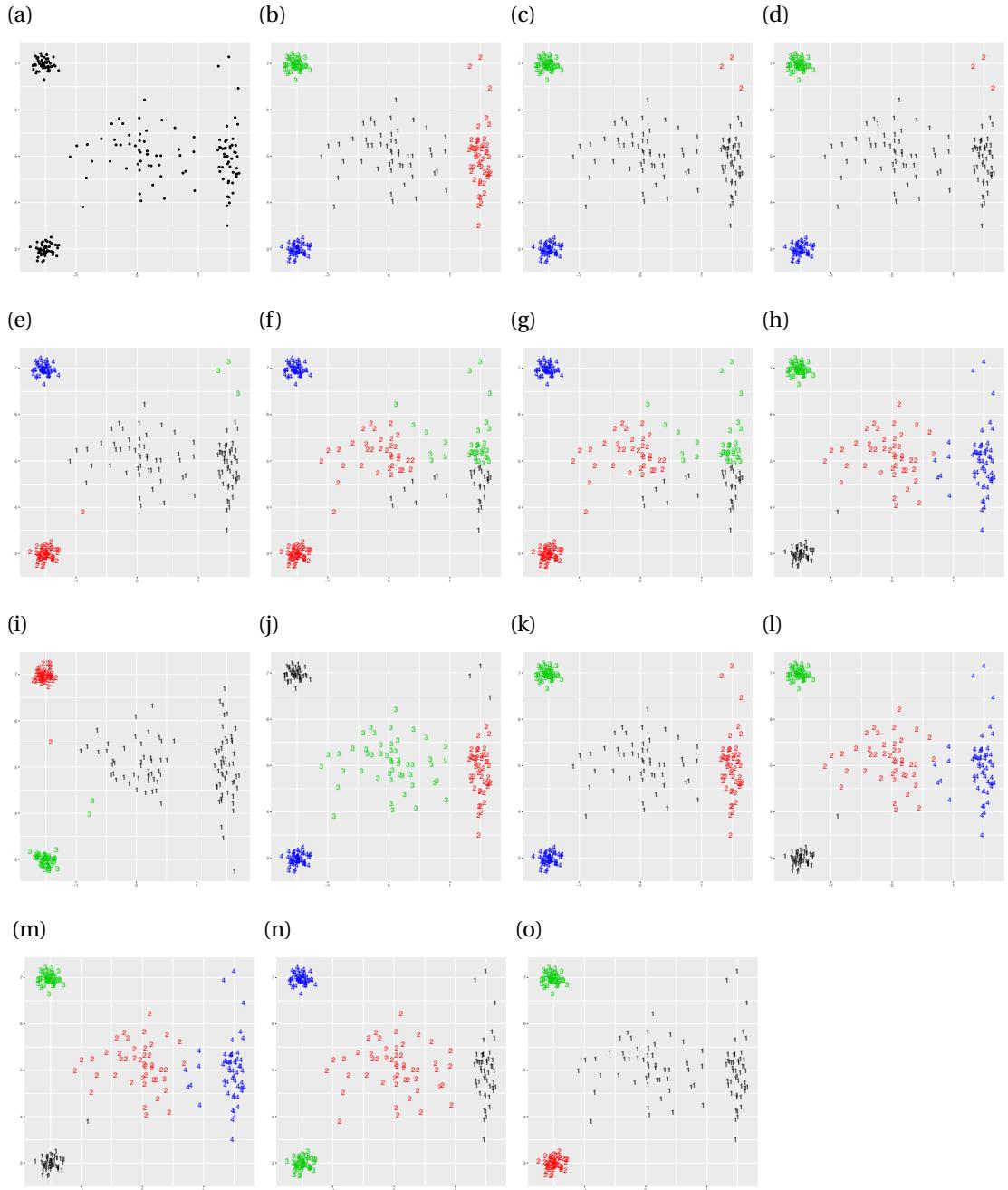


Figure B.8 Clustering results from clustering methods for a simulated dataset from Model 6.B, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering, (i) PAM clustering, (j) spectral clustering, (k) model-based clustering, (l) PAMSIL $_k$ clustering, (m) PAMSIL $_{\hat{k}}$ clustering, (n) HOSil $_k$, and (o) HOSil $_{\hat{k}}$.

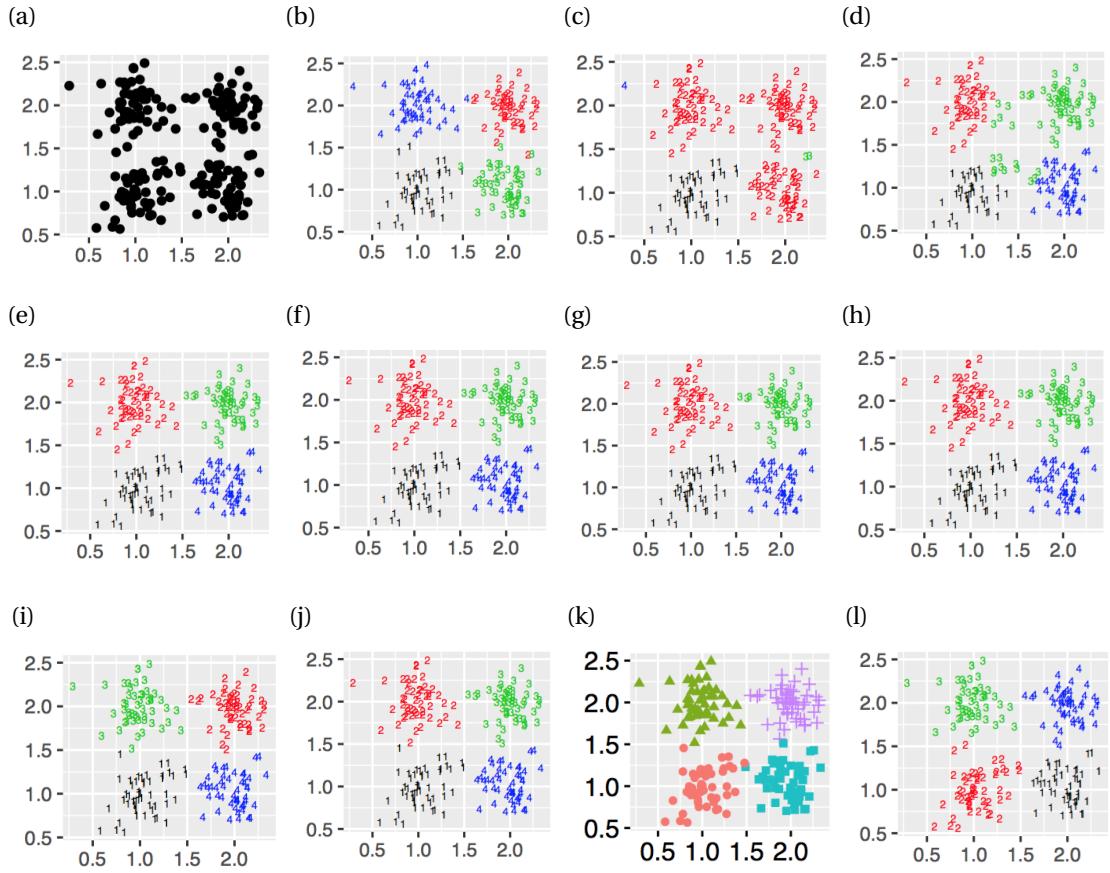


Figure B.9 Clustering results from clustering methods for a simulated dataset from Model 7, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering, (i) PAM clustering, (j) spectral clustering, (k) model-based clustering, (l) PAMSIL $_k$ /PAMSIL $_{\hat{k}}$ and HOSil $_k$ /HOSil $_{\hat{k}}$ clustering.

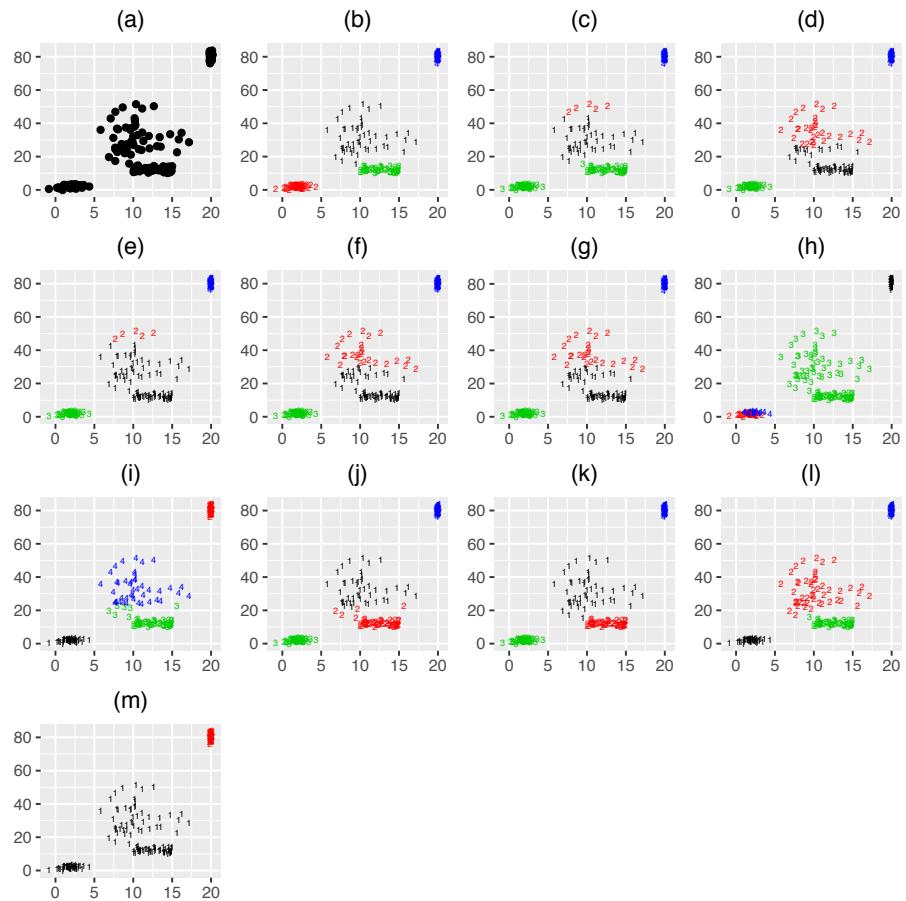


Figure B.10 Clustering results from clustering methods for a simulated dataset from Model 8, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) spectral clustering, (i) PAM and PAMSIL_k clustering, (j) k -means clustering, (k) model-based clustering, (l) HOSil_k clustering, (m) PAMSIL _{\hat{k}} and HOSil _{\hat{k}} clustering.

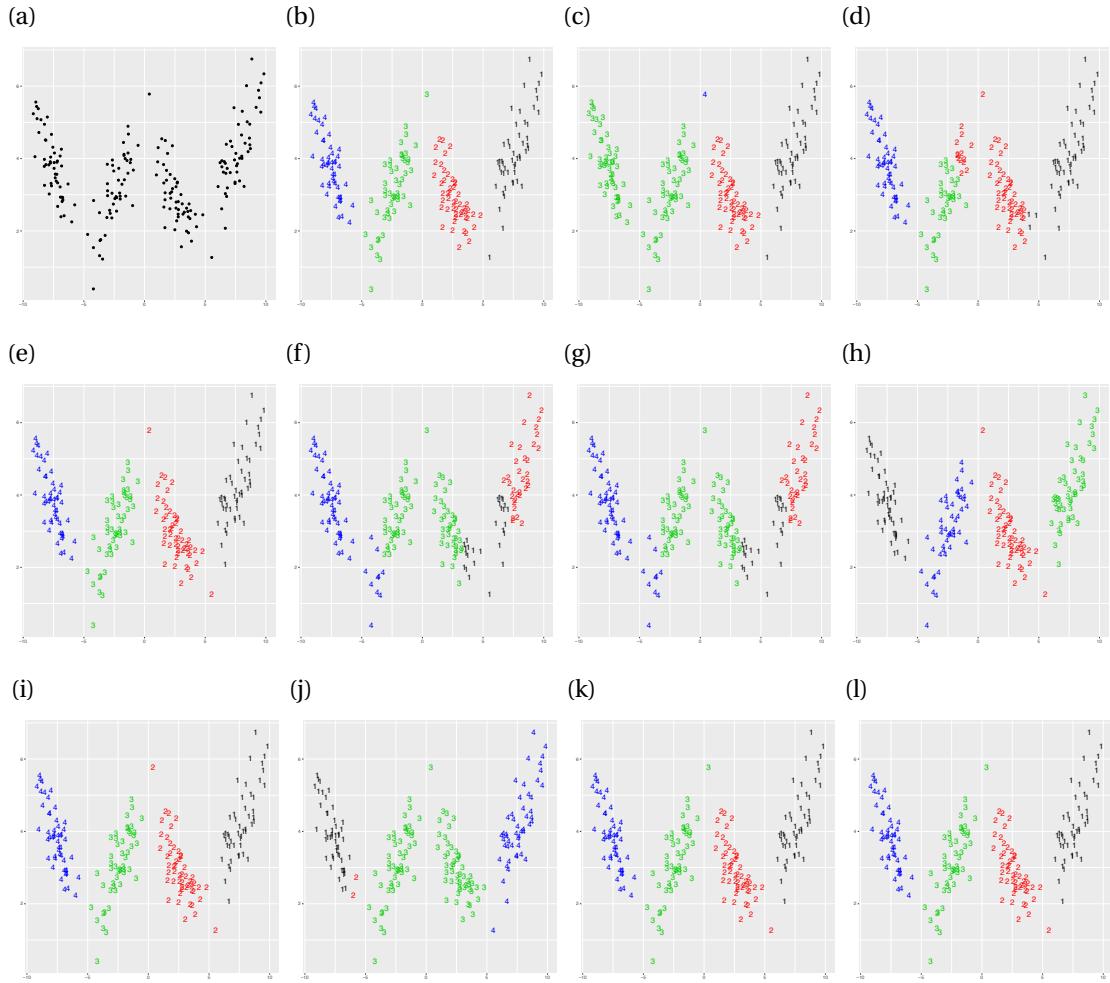


Figure B.11 Clustering results from clustering methods for a simulated dataset from Model 9, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering, (i) PAM clustering, (j) spectral clustering, (k) model-based clustering, (l) PAMSIL $_k$ /PAMSIL $_{\hat{k}}$ and HOSil $_k$ /HOSil $_{\hat{k}}$ clustering.

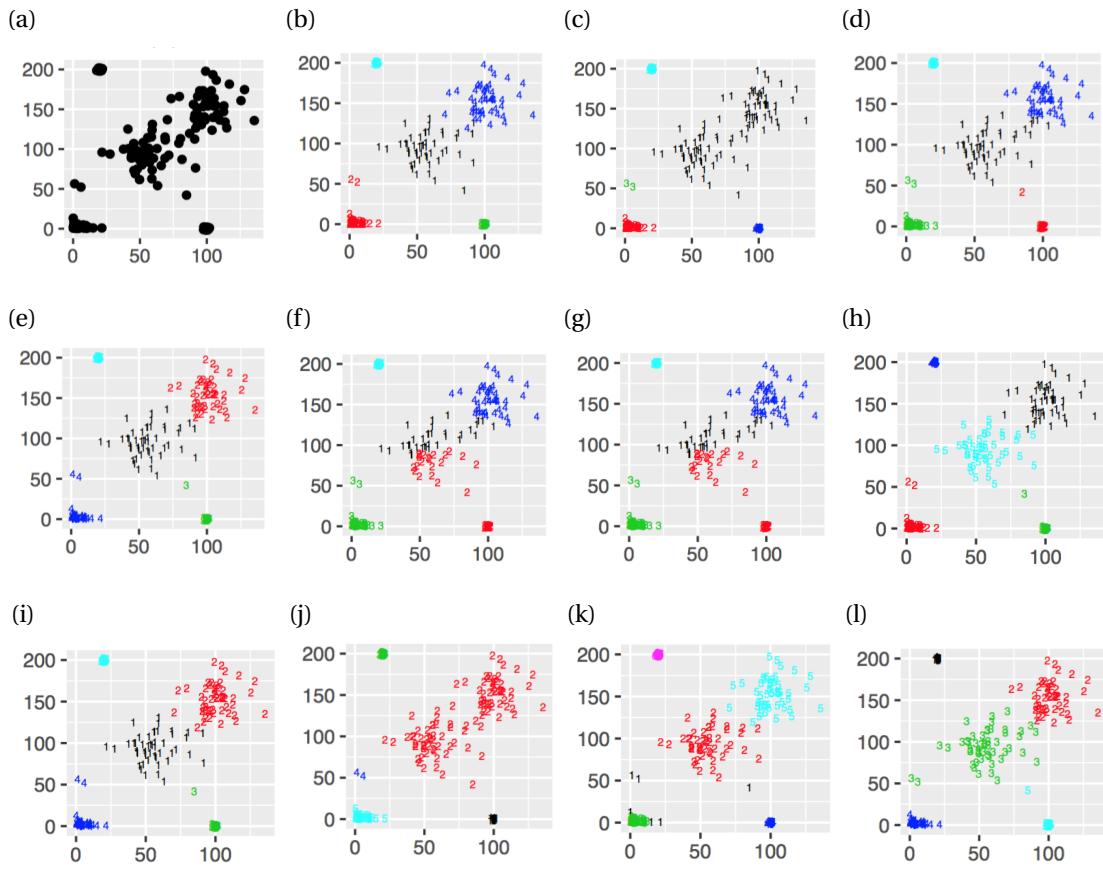


Figure B.12 Clustering results from clustering methods for a simulated dataset from Model 10, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering (i) PAM clustering (j) spectral clustering (k) model-based clustering (l) $\text{PAMSIL}_k/\text{PAMSIL}_{\hat{k}}$ and $\text{HOSil}_k/\text{HOSil}_{\hat{k}}$ clustering.

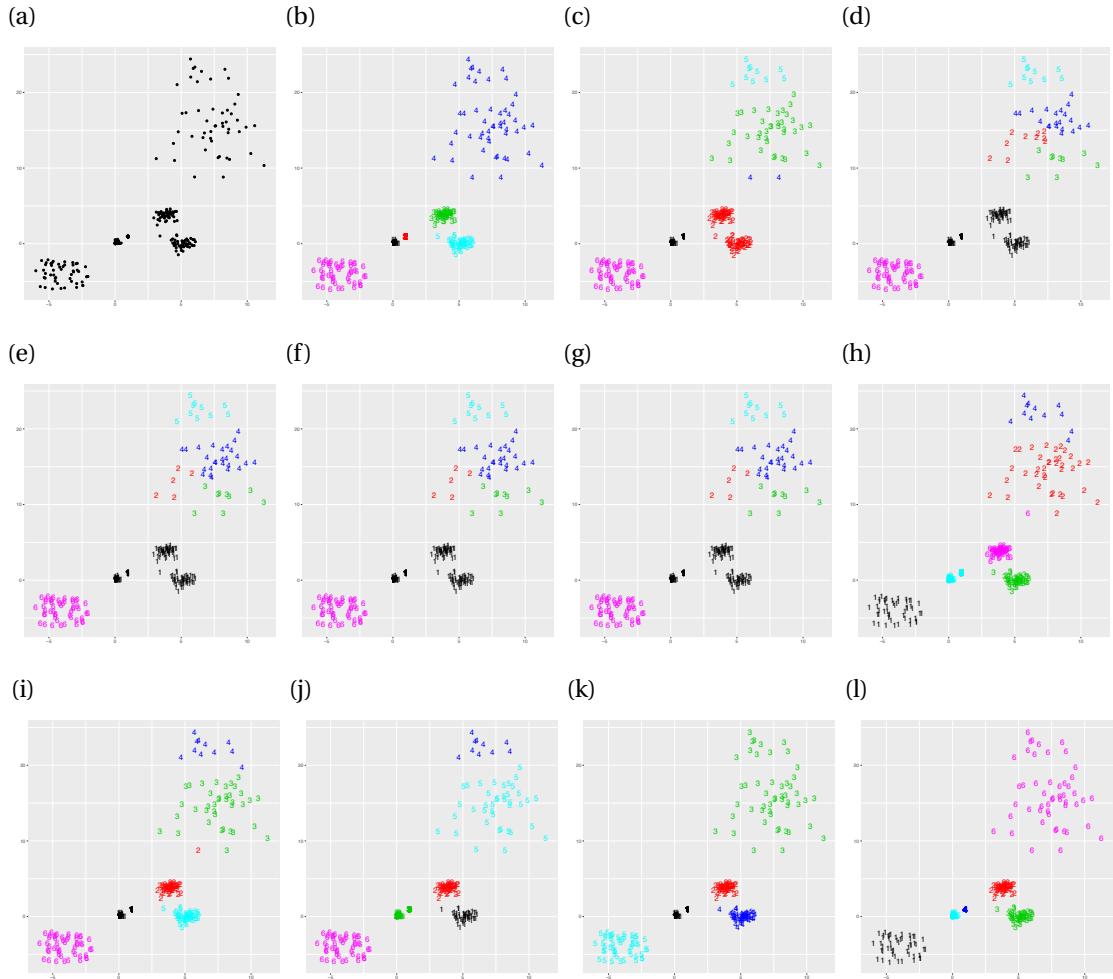


Figure B.13 Clustering results from clustering methods for a simulated dataset from Model 11, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering (i) PAM clustering (j) spectral clustering (k) model-based clustering (l) $\text{PAMSIL}_k/\text{PAMSIL}_{\hat{k}}$ and $\text{HOSil}_k/\text{HOSil}_{\hat{k}}$ clustering.

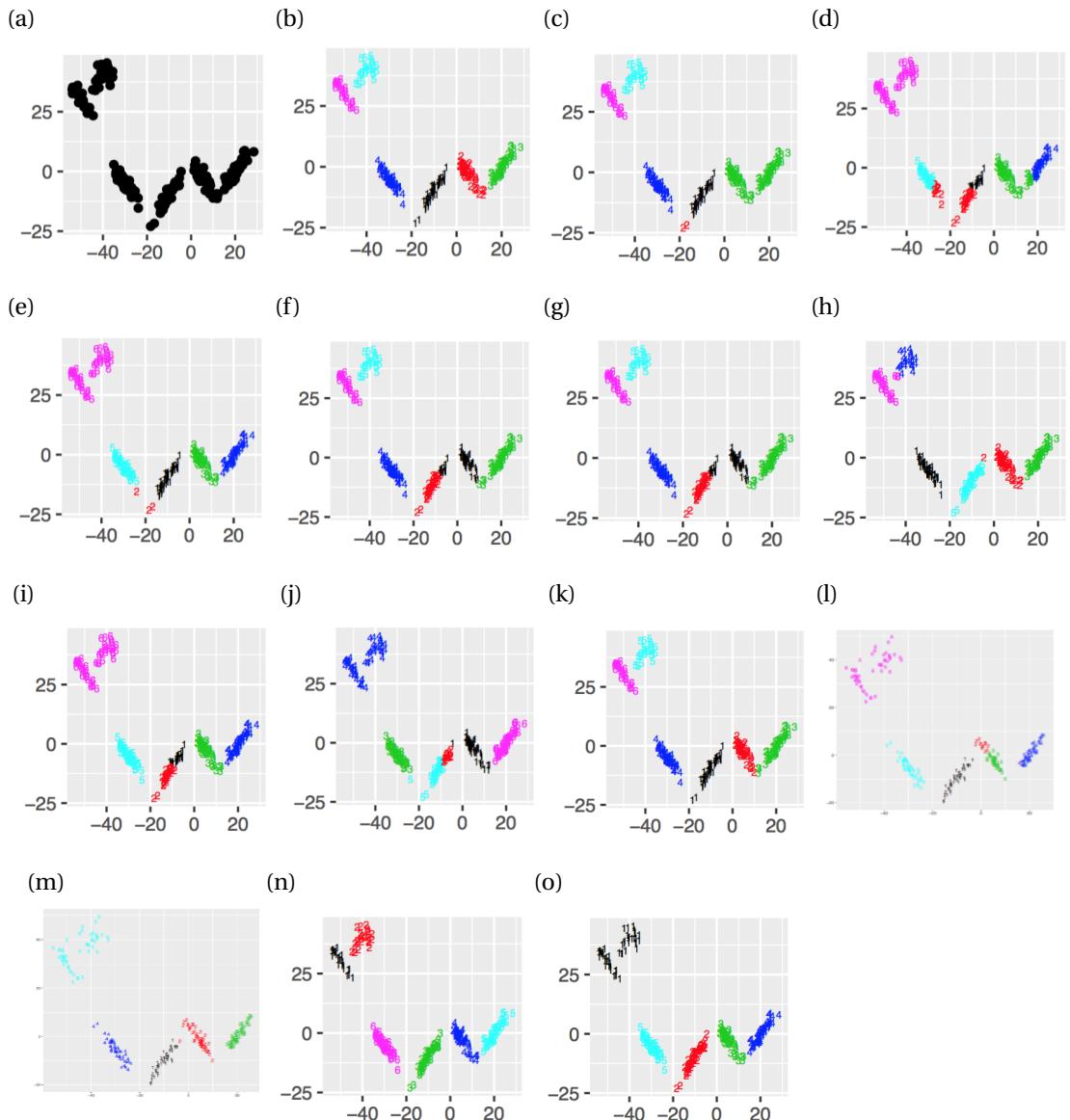


Figure B.14 Clustering results from clustering methods for a simulated dataset from Model 12, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering (i) PAM clustering (j) spectral clustering (k) model-based clustering (l) PAMSIL $_k$ clustering (m) PAMSIL $_{\hat{k}}$ (n) HOSil $_k$ clustering (o) HOSil $_{\hat{k}}$ clustering.

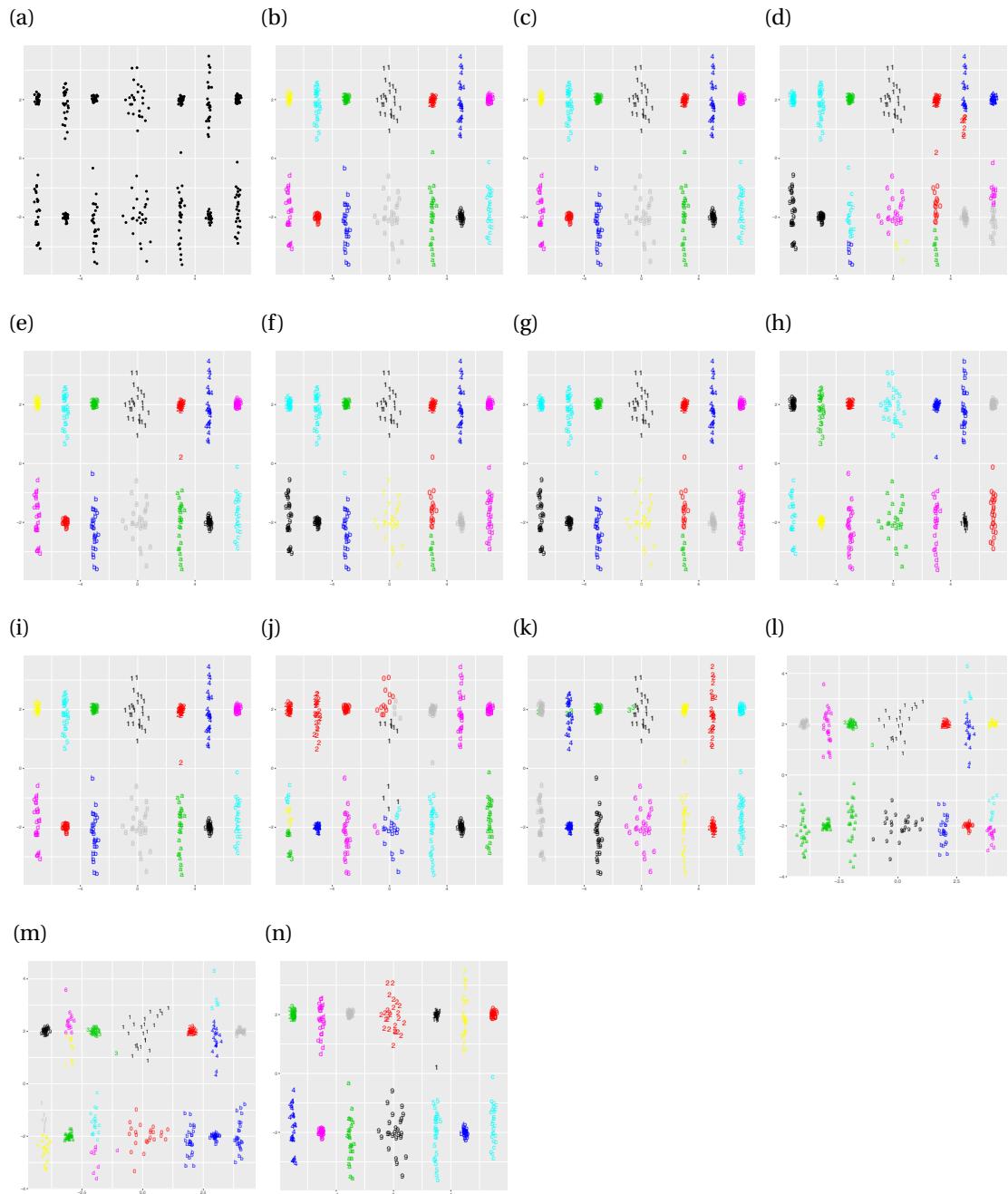


Figure B.15 Clustering results from clustering methods for a simulated dataset from Model 13, (a) the raw data, (b) data plotting against true labels, hierarchical clustering solution with (c) single linkage, (d) complete linkage, (e) average linkage, (f) Ward's method, (g) Mcquitty similarity, (h) k -means clustering (i) PAM clustering (j) spectral clustering (k) model-based clustering (l) PAMSIL $_k$ clustering (m) PAMSIL $_{\hat{k}}$ clustering (n) HOSil $_k$ /HOSil $_{\hat{k}}$ clustering.

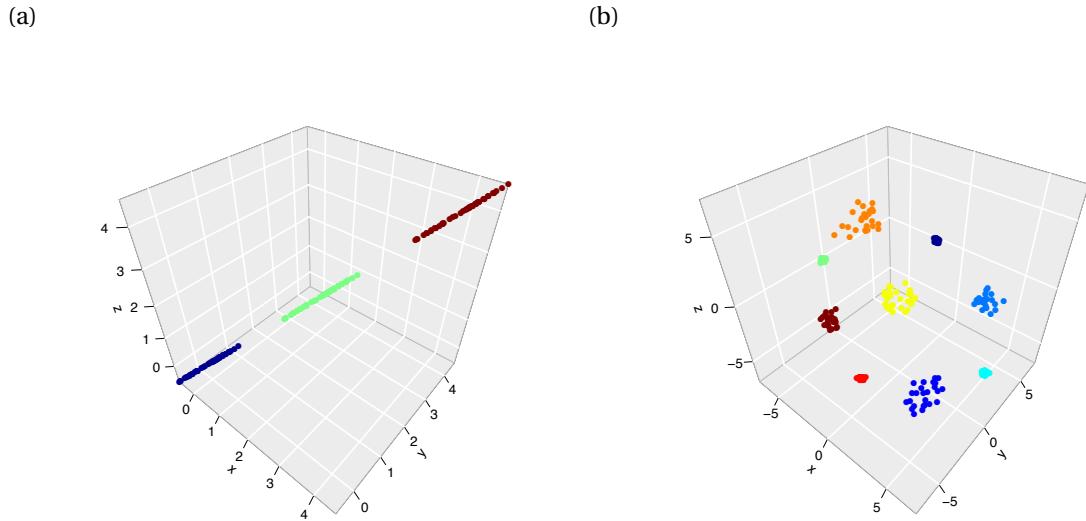


Figure B.16 Data plot generated from (a) Model 14 (b) Model 15. All the clustering methods included in the study were successful retrieving the true clustering

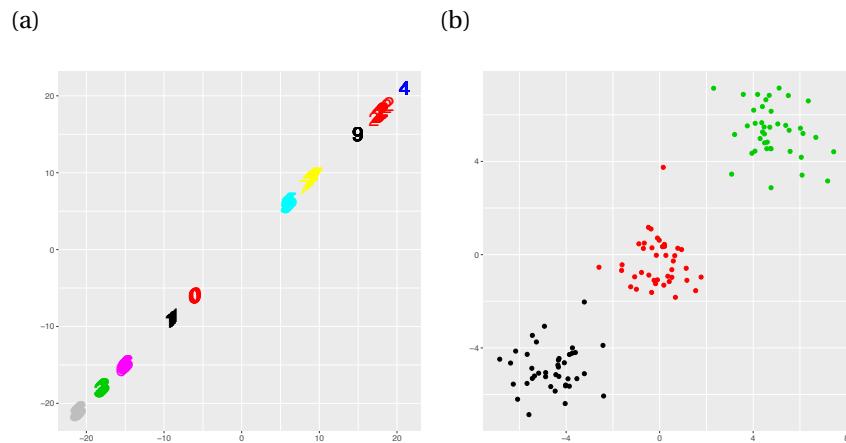


Figure B.17 The left panel represents Model 16 and right panel is Model 17, where the colours represents the true labels according to the data generating clustering models.

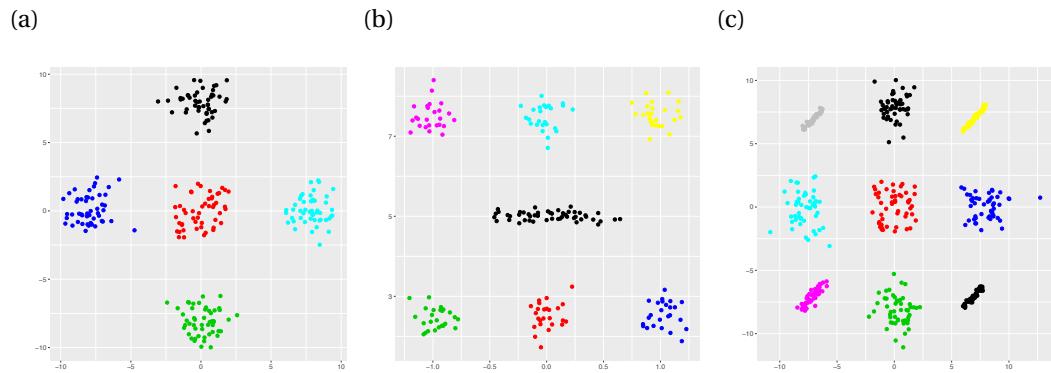


Figure B.18 HOSil \hat{k} clustering results (a) Model 18 (b) Model 19 (c) Model 20.

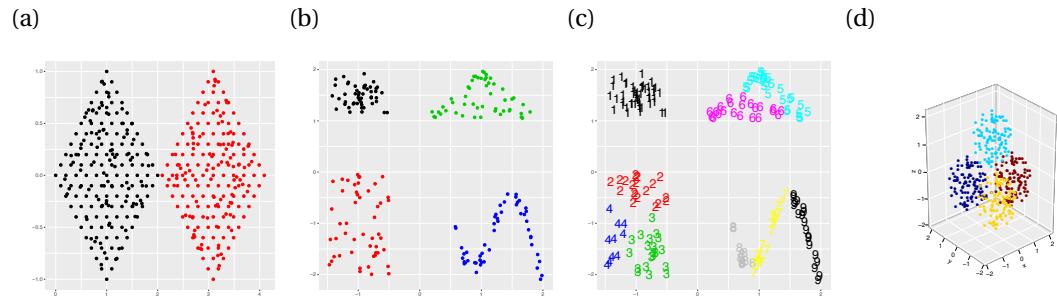


Figure B.19 HOSil \hat{k} clustering results on (a) 2 Diamonds data (b) Four Shapes data, (c) represents the model-based clustering results on Four Shapes data, (d) Tetra data

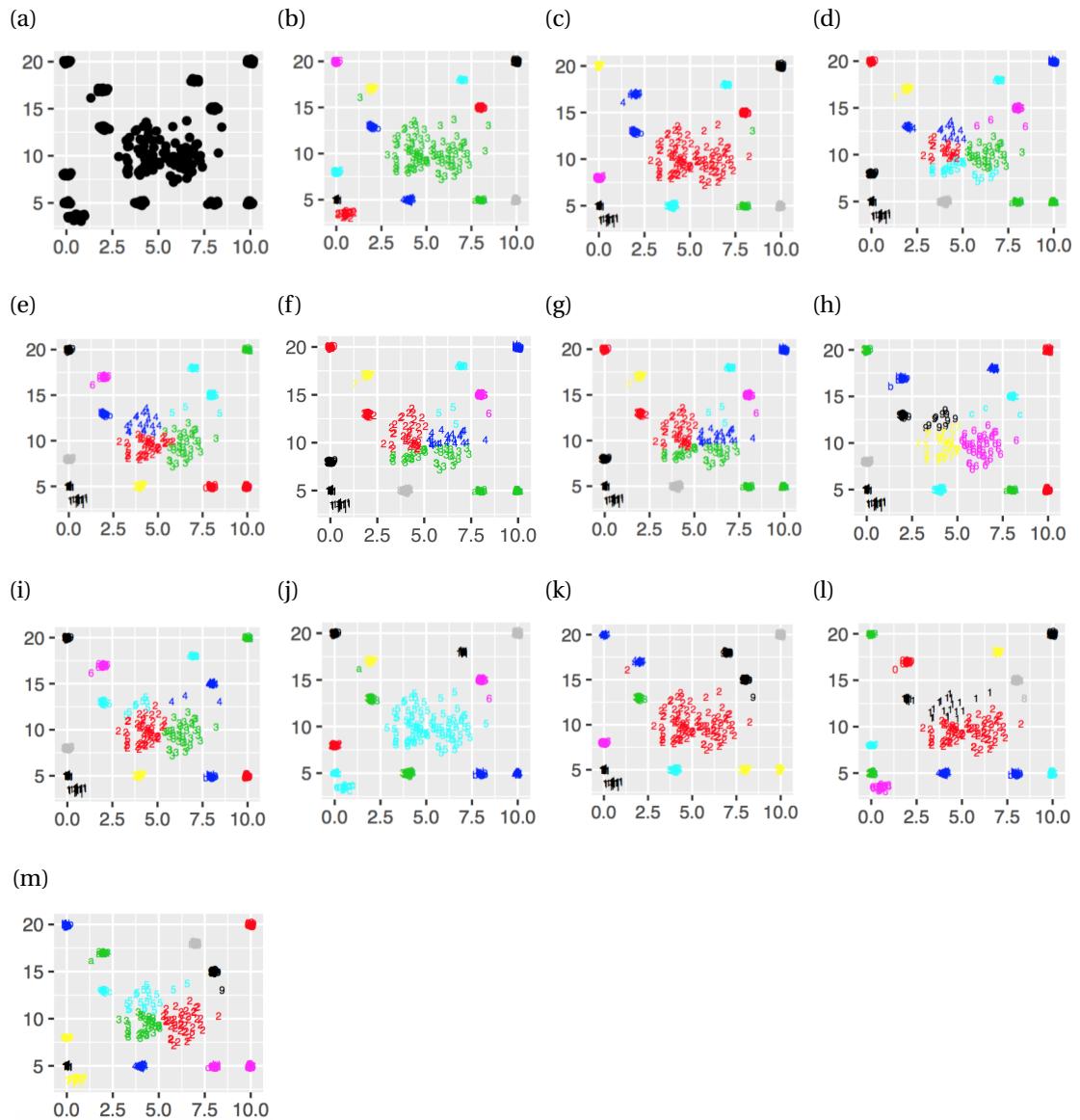


Figure B.20 Clustering results for Model 21, (a) raw data (b) cluster colours against true labels, hierarchical clustering results using (c) single (d) complete (e) average (f) Ward's (g) McQuitty methods, (h) k -means clustering (i) PAM clustering (j) spectral clustering (k) model-based clustering (l) HOSil_k (m) $\text{HOSil}_{\hat{k}}$.

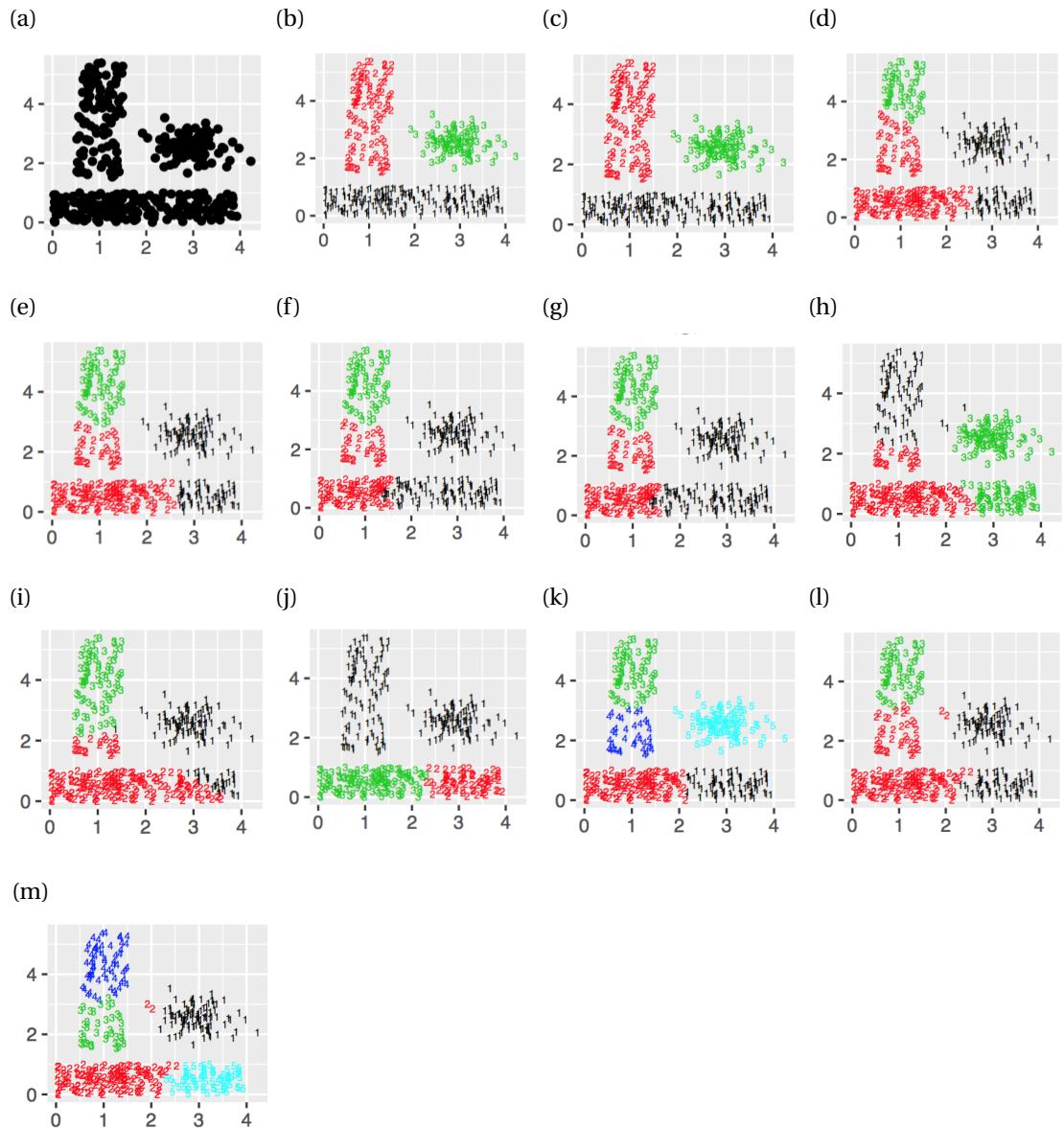


Figure B.21 Clustering results on Lsun data (a) raw data (b) cluster colours for the true labels, hierarchical clustering results using (c) single (d) complete (e) average (f) Ward's (g) McQuitty methods, (h) k -means clustering (i) pam clustering (j) spectral clustering (k) model-based clustering (l) $HOSil_k$ (m) $HOSil_{\hat{k}}$.

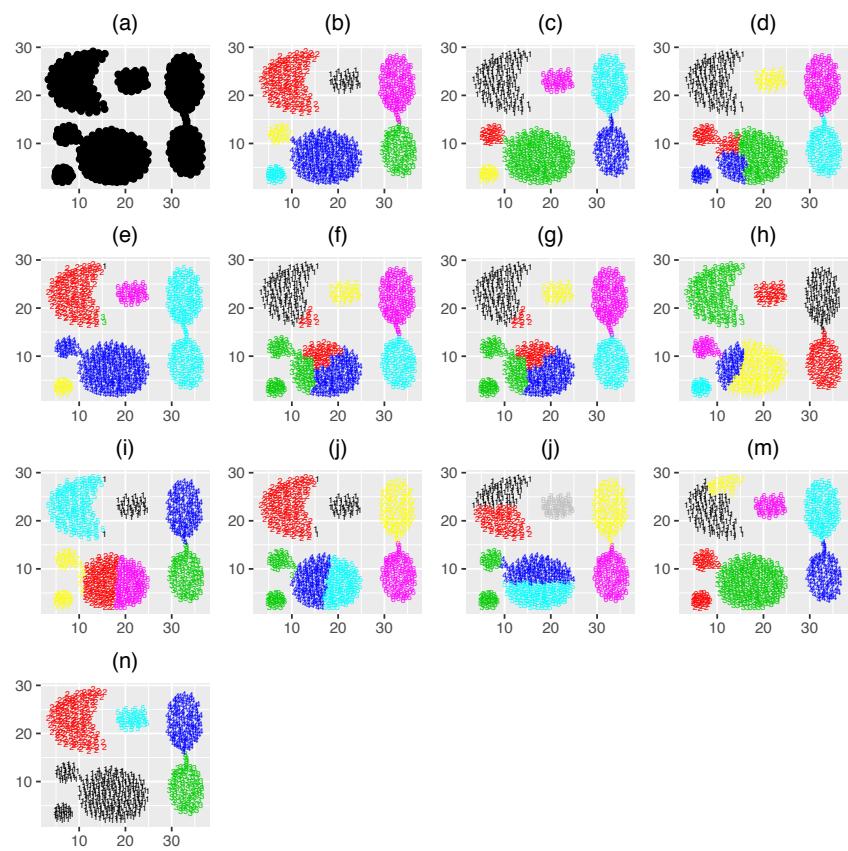


Figure B.22 Clustering results on Aggregation data (a) raw data (b) plotting against true labels (c) average (d) complete (e) single (f) McQuitty (g) Ward's (h) spectral (i) k -means (j) PAM (l) model-based clustering (m) HOSil_k (n) $\text{HOSil}_{\hat{k}}$. ($v = 400$)

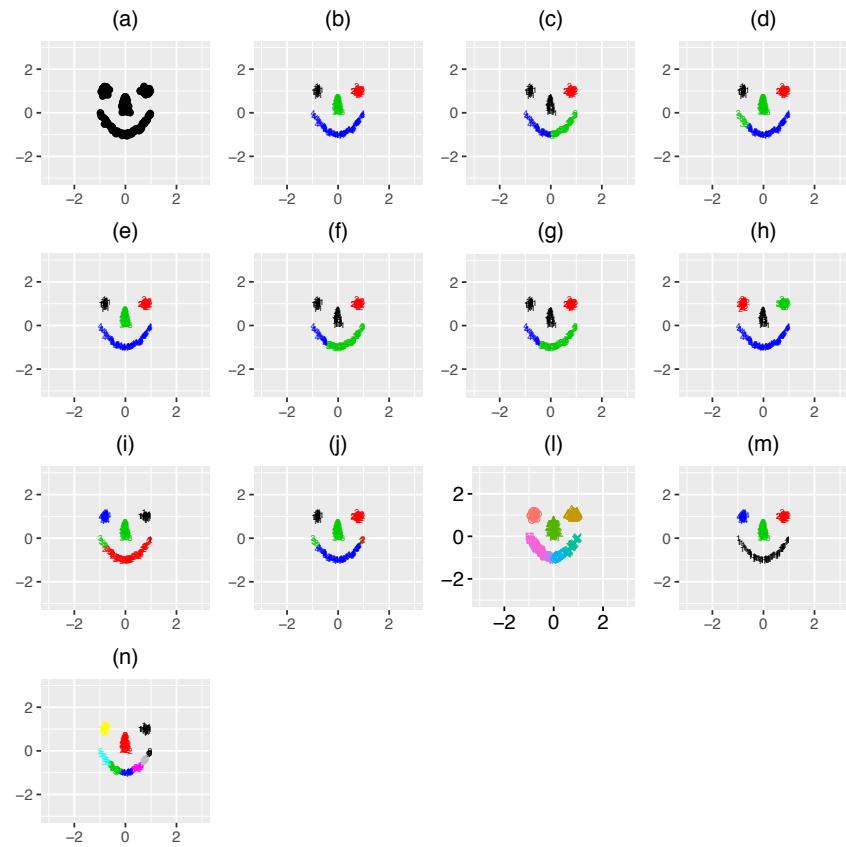


Figure B.23 HOSil clustering comparison with other clustering methods for Smiley data set (a) the raw data (b) color represent true cluster numbers (c), (d), (e), (f) and (g) represents clustering by average, complete, single, McQuitty and Ward's methods (h) shows results obtained by spectral clustering (i) and (j) are k -means and pam clustering results (l) represents model-based clustering results, (m) and (n) are HOSil _{k} and HOSil _{\hat{k}} clusterings.

B.2 HOSil estimation of number of clusters

Table B.1: Results for estimation of number of clusters \hat{k} from indices and clustering methods included in the study for Model 1. The true number of clusters are made bold. Note that CH , KL , ASW , BI , $PAMSIL$ and $HOSil$ can not estimate $\hat{k} = 1$.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	22	13	7	5	3	0	0	0	0	0	0	0	0	0	0
Complete	0	0	0	0	0	0	3	1	0	1	5	5	35	0	0
Average	1	1	0	0	2	2	4	15	13	4	3	4	1	0	0
Ward	0	0	0	3	3	2	4	4	6	2	4	7	15	0	0
McQuitty	0	0	0	3	3	2	4	4	6	2	4	7	15	0	0
kmeans	0	3	5	7	9	12	8	3	3	0	0	0	0	0	0
PAM	0	0	0	0	0	0	0	2	7	2	3	7	29	0	0
CH															
Single	0	39	4	0	1	0	1	0	2	1	0	0	1	0	1
Complete	0	37	0	3	10	0	0	0	0	0	0	0	0	0	0
Average	0	33	0	0	11	1	0	0	0	0	0	0	1	1	3
Ward	0	34	0	2	11	1	0	0	0	0	0	0	0	0	2
McQuitty	0	34	0	2	11	1	0	0	0	0	0	0	0	0	2
kmeans	0	36	0	4	7	0	0	0	1	0	1	1	0	0	0
PAM	0	29	0	2	14	0	0	1	0	0	0	1	1	1	1
KL															
Single	0	3	4	3	3	2	9	4	4	8	5	5	0	0	0
Complete	0	2	7	14	5	2	1	3	3	3	8	2	0	0	0
Average	0	2	7	13	2	7	3	3	4	3	3	3	0	0	0
Ward	0	1	13	7	2	4	1	5	2	5	4	6	0	0	0
McQuitty	0	1	13	7	2	4	1	5	2	5	4	6	0	0	0
kmeans	0	22	10	4	3	2	2	0	2	1	4	0	0	0	0
PAM	0	7	3	7	4	5	2	3	5	5	4	5	0	0	0
Gap															
Single	10	36	4	0	0	0	0	0	0	0	0	0	0	0	0
Complete	0	18	27	5	0	0	0	0	0	0	0	0	0	0	0
Average	0	19	26	5	0	0	0	0	0	0	0	0	0	0	0
Ward	0	28	16	3	0	0	0	0	0	0	0	0	0	0	0
McQuitty	0	26	21	3	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	11	31	7	1	0	0	0	0	0	0	0	0	0	0
PAM	0	12	24	14	0	0	0	0	0	0	0	0	0	0	0
Jump															
p/2	0	1	0	4	9	0	0	0	2	3	4	6	7	8	6
p/3	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
p/4	43	7	0	0	0	0	0	0	0	0	0	0	0	0	0
p/5	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p/6	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p/7	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PS															
Single	47	3	0	0	0	0	0	0	0	0	0	0	0	0	0
Complete	47	2	1	0	0	0	0	0	0	0	0	0	0	0	0
Average	18	28	4	0	0	0	0	0	0	0	0	0	0	0	0
Ward	18	44	3	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	47	3	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0

PAM	0	44	6	0	0	0	0	0	0	0	0	0	0	0	0
BI															
Single	0	0	0	0	0	0	0	0	0	0	2	4	18	26	
Complete	0	0	2	3	10	22	12	0	0	0	0	0	0	0	1
Average	0	2	0	0	1	4	14	11	10	4	2	0	2	0	0
Ward	0	28	4	3	7	6	1	0	0	0	0	0	0	0	1
McQuitty	0	0	1	0	7	23	13	4	1	0	0	0	0	0	0
kmeans	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	32	18	0	0	0	0	0	0	0	0	0	0	0	0
ASW															
Single	0	39	4	0	1	0	1	1	1	1	0	0	0	0	1
Complete	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Ward	0	48	0	1	0	0	1	0	0	0	0	0	0	0	0
McQuitty	0	48	1	0	0	1	0	0	0	0	0	0	0	0	0
kmeans	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Spectral	0	48	0	0	1	0	1	0	0	0	0	0	0	0	0
Model-based	0	49	0	0	0	0	0	1	0	0	0	0	0	0	0
BIC															
Model-based	0	43	3	4	0	0	0	0	0	0	0	0	0	0	0
PAMSIL	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
HOSil	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0

Table B.2: Results for the estimation of number of clusters from indices and clustering methods for Model 2.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
Complete	0	0	2	0	3	3	1	8	1	4	6	9	2	11	0
Average	0	10	6	4	5	1	4	3	1	0	0	0	0	9	0
Ward	0	5	6	3	2	9	4	1	7	3	1	3	0	0	0
McQuitty	0	5	6	3	2	9	4	1	7	3	1	3	0	0	0
kmeans	0	0	0	5	13	13	8	11	0	0	0	0	0	0	0
PAM	0	0	0	0	0	0	0	6	5	12	12	6	4	5	0
CH															
Single	0	2	8	3	5	5	6	3	2	4	2	2	2	2	4
Complete	0	4	1	3	1	2	0	3	3	2	0	3	3	10	15
Average	0	24	0	1	1	0	1	0	1	1	0	5	4	2	10
Ward	0	11	2	2	4	0	1	1	3	3	2	2	3	7	9
McQuitty	0	11	2	2	4	0	1	1	3	3	2	2	3	7	9
kmeans	0	29	2	6	4	3	2	1	0	0	0	0	1	2	0
PAM	0	29	1	4	0	2	2	0	1	3	2	1	1	1	3
KL															
Single	0	1	5	6	10	2	8	3	5	3	4	3	0	0	0
Complete	0	3	4	12	4	6	7	3	4	2	3	2	0	0	0
Average	0	5	4	4	1	6	4	5	4	7	4	6	0	0	0
Ward	0	4	4	2	2	11	4	5	6	7	3	2	0	0	0
McQuitty	0	4	4	2	2	11	4	5	6	7	3	2	0	0	0
kmeans	0	23	2	2	3	2	4	3	1	3	3	4	0	0	0
PAM	0	10	4	1	4	5	5	4	5	3	6	3	0	0	0

	Gap														
Single	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Complete	19	28	2	1	0	0	0	0	0	0	0	0	0	0	0
Average	17	33	0	0	0	0	0	0	0	0	0	0	0	0	0
Ward	25	22		3	0	0	0	0	0	0	0	0	0	0	0
McQuitty	27	20	3	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	40	9	1	0	0	0	0	0	0	0	0	0	0	0
PAM	0	42	7	1	0	0	0	0	0	0	0	0	0	0	0
	Jump														
$p/2$	0	0	0	3	0	0	3	0	1	4	5	3	8	12	11
$p/3$	34	16	0	0	0	0	0	0	0	0	0	0	0	0	0
$p/4$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$p/5$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$p/6$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$p/7$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	PS														
Single	1	20	23	6	0	0	0	0	0	0	0	0	0	0	0
Complete	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	40	10	0	0	0	0	0	0	0	0	0	0	0	0	0
Ward	42	8	0	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	49	1	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	1	49	0	0	0	0	0	0	0	0	0	0	0	0	0
PAM	3	45	2	0	0	0	0	0	0	0	0	0	0	0	0
	BI														
Single	0	8	0	1	0	0	0	0	0	0	0	0	1	11	29
Complete	0	0	0	0	0	0	0	0	5	4	10	10	9	12	
Average	0	0	0	0	0	0	0	0	0	1	1	3	9	36	
Ward	0	0	0	0	0	0	1	3	10	11	3	5	0	6	11
McQuitty	0	0	0	0	0	0	0	0	0	0	6	5	9	12	18
kmeans	0	47	0	0	0	0	0	0	0	0	0	0	0	0	3
PAM	0	28	6	0	0	0	1	1	0	0	0	0	2	6	6
	BIC														
Model-based	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
	ASW														
Single	0	28	8	2	2	3	1	2	1	1	0	1	0	0	1
Complete	0	13	3	5	1	1	1	3	2	1	5	3	3	3	6
Average	0	32	10	1	0	2	2	0	0	0	0	1	1	0	1
Ward	0	15	5	0	3	5	3	4	2	0	1	2	3	5	2
McQuitty	0	15	5	0	3	5	3	4	2	0	1	2	3	5	2
kmeans	0	49	0	1	0	0	0	0	0	0	0	0	0	0	0
PAM	0	49	0	1	0	0	0	0	0	0	0	0	0	0	0
Spectral	0	49	0	0	0	1	0	0	0	0	0	0	0	0	0
Model-based	0	49	0	0	0	0	0	0	0	0	0	0	0	1	0
PAMSIL	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
HOSil	0	49	0	0	0	0	0	0	0	0	0	0	0	0	0

Table B.3: Results for the estimation of number of clusters from indices and clustering methods for Model 3.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	0	2	8	11	6	9	4	4	2	3	0	0	0	1	0
Complete	0	0	0	0	0	5	3	4	9	6	8	6	5	3	0
Average	0	0	3	6	3	5	5	7	8	5	5	3	0	0	0
Ward	0	1	0	0	0	5	3	4	9	6	8	6	5	3	0
McQuitty	0	1	0	0	0	5	3	4	9	6	8	6	5	3	0
kmeans	0	2	24	18	6	0	0	0	0	0	0	0	0	0	0
PAM	0	5	0	0	0	1	9	7	7	9	7	1	2	2	0
CH															
Single	0	41	8	1	0	0	0	0	0	0	0	0	0	0	0
Complete	0	32	11	0	2	1	1	1	0	0	0	0	1	0	1
Average	0	42	2	1	0	2	0	1	1	1	0	0	0	0	0
Ward	0	39	4	2	2	2	0	0	1	0	0	0	0	0	0
McQuitty	0	39	4	2	2	2	0	0	1	0	0	0	0	0	0
kmeans	0	47	3	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	37	0	0	0	0	0	0	0	0	0	0	1	3	9
KL															
Single	0	5	4	2	4	6	6	3	6	4	3	7	0	0	0
Complete	0	18	3	4	2	6	2	2	5	2	2	4	0	0	0
Average	0	7	5	4	2	6	5	4	4	5	5	3	0	0	0
Ward	0	5	3	3	8	4	6	1	7	6	4	3	0	0	0
McQuitty	0	5	3	3	8	4	6	1	7	6	4	3	0	0	0
kmeans	0	4	5	6	3	2	4	3	6	7	5	5	0	0	0
PAM	0	12	6	5	10	4	0	2	2	4	3	2	0	0	0
Gap															
Single	8	34	8	0	0	0	0	0	0	0	0	0	0	0	0
Complete	9	25	15	1	0	0	0	0	0	0	0	0	0	0	0
Average	5	35	7	3	0	0	0	0	0	0	0	0	0	0	0
Ward	6	29	14	1	0	0	0	0	0	0	0	0	0	0	0
McQuitty	6	29	14	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	38	12	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	49	1	0	0	0	0	0	0	0	0	0	0	0	0
Jump															
p/2	0	0	0	0	0	1	1	1	4	5	6	6	6	11	9
p/3	0	30	0	0	0	0	0	1	3	2	1	4	1	5	3
p/4	0	47	0	0	0	0	0	0	0	0	1	0	0	1	1
p/5	0	49	0	0	0	0	0	0	0	0	0	0	0	0	0
p/6	28	22	0	0	0	0	0	0	0	0	0	0	0	0	0
p/7	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PS															
Single	2	10	32	6	0	0	0	0	0	0	0	0	0	0	0
Complete	2	48	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	1	42	7	0	0	0	0	0	0	0	0	0	0	0	0
Ward	0	49	1	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	4	45	1	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	34	16	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	18	31	1	0	0	0	0	0	0	0	0	0	0	0
BI															
Single	0	27	4	0	0	0	0	0	0	0	0	0	0	3	0

Complete	0	17	0	0	0	0	0	1	5	10	11	4	2	0	0
Average	0	29	0	0	0	0	0	0	0	0	0	2	0	9	10
Ward	0	49	1	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	0	18	0	0	0	0	0	0	0	1	5	5	8	7	6
kmeans	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
BIC															
Model-based	0	47	1	0	0	0	1	0	1	0	0	0	0	0	0
ASW															
Single	0	41	8	1	0	0	0	0	0	0	0	0	0	0	0
Complete	0	36	12	1	0	1	0	0	0	0	0	0	0	0	0
Average	0	45	5	0	0	0	0	0	0	0	0	0	0	0	0
Ward	0	43	5	1	1	0	0	0	0	0	0	0	0	0	0
McQuitty	0	43	5	1	1	0	0	0	0	0	0	0	0	0	0
kmeans	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Spectral	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Model-based	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
PAMSIL	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
HOSil	0	49	1	0	0	0	0	0	0	0	0	0	0	0	0

Table B.4: Results for the estimation of number of clusters from indices and clustering methods for Model 4.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	0	16	13	4	2	1	0	0	0	0	0	0	0	0	0
Complete	0	0	0	2	0	0	1	0	0	2	9	4	7	25	0
Average	0	0	10	10	3	6	5	5	4	3	1	2	0	1	0
Ward	0	3	0	4	3	6	4	12	7	2	2	3	4	0	0
McQuitty	0	3	0	4	3	6	4	12	7	2	2	3	4	0	0
kmeans	0	0	1	12	15	11	6	12	1	1	0	0	0	0	0
PAM	0	0	0	0	0	0	0	1	1	4	12	12	6	14	0
CH															
Single	0	1	20	15	5	7	1	1	0	0	0	0	0	0	0
Complete	0	0	1	2	2	2	3	3	4	9	8	6	2	3	5
Average	0	0	15	1	0	0	2	0	3	5	3	6	3	5	7
Ward	0	0	2	0	5	4	3	6	4	2	1	6	5	4	8
McQuitty	0	0	2	0	5	4	3	6	4	2	1	6	5	4	8
kmeans	0	0	0	3	15	17	6	4	4	1	0	0	0	0	0
PAM	0	0	0	2	9	9	9	7	3	1	4	2	3	0	1
KL															
Single	0	13	6	7	2	5	2	3	2	6	2	2	0	0	0
Complete	0	13	4	13	6	5	2	0	4	1	2	0	0	0	0
Average	0	21	0	2	4	1	2	5	5	1	4	5	0	0	0
Ward	0	7	8	3	4	1	5	6	4	4	3	5	0	0	0
McQuitty	0	7	8	3	4	1	5	6	4	4	3	5	0	0	0
kmeans	0	19	3	9	5	3	1	0	3	1	4	2	0	0	0
PAM	0	1	12	4	5	6	4	4	3	5	2	4	0	0	0
Gap															
Single	14	16	20	0	0	0	0	0	0	0	0	0	0	0	0
Complete	4	11	17	9	7	1	1	0	0	0	0	0	0	0	0
Average	2	0	44	4	0	0	0	0	0	0	0	0	0	0	0

Ward	6	13	19	7	4	1	0	0	0	0	0	0	0	0	0
McQuitty	6	13	18	8	4	1	0	0	0	0	0	0	0	0	0
kmeans	2	0	9	23	16	0	0	0	0	0	0	0	0	0	0
PAM	6	0	6	18	19	1	0	0	0	0	0	0	0	0	0
Jump															
$p/2$	0	0	14	0	1	0	3	1	2	1	1	4	6	6	11
$p/3$	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
$p/4$	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
$p/5$	40	0	10	0	0	0	0	0	0	0	0	0	0	0	0
$p/6$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$p/7$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PS															
Single	15	33	2	0	0	0	0	0	0	0	0	0	0	0	0
Complete	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	42	8	0	0	0	0	0	0	0	0	0	0	0	0	0
Ward	49	0	0	1	0	0	0	0	0	0	0	0	0	0	0
McQuitty	49	1	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
PAM	46	0	1	3	0	0	0	0	0	0	0	0	0	0	0
BI															
Single	0	0	0	0	0	0	0	0	0	0	1	0	5	10	34
Complete	0	0	0	0	0	0	3	10	8	9	5	5	1	3	6
Average	0	0	0	0	0	0	1	1	1	1	3	6	10	10	17
Ward	0	0	0	3	6	10	10	9	3	2	0	0	0	2	5
McQuitty	0	0	0	0	0	0	3	5	8	5	8	8	7	4	2
kmeans	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	18	23	6	2	0	0	0	0	0	0	1	0
BIC															
Model-based	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
ASW															
Single	0	6	20	15	3	5	0	1	0	0	0	0	0	0	0
Complete	0	0	19	1	5	5	4	3	4	5	1	0	0	1	2
Average	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
Ward	0	1	22	3	5	2	3	4	3	0	1	1	2	1	2
McQuitty	0	1	22	3	5	2	3	4	3	0	1	1	2	1	2
kmeans	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
Spectral	0	0	38	6	3	1	0	1	1	0	0	0	0	0	0
Model-based	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
PAMSIL	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
HOSil	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0

Table B.5: Results for the estimation of number of clusters from indices and clustering methods for Model 5.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	13	13	19	5	0	0	0	0	0	0	0	0	0	0	0
Complete	0	1	1	0	0	2	2	3	2	2	8	6	4	21	0
Average	0	5	15	11	5	3	2	2	2	1	2	2	0	0	0
Ward	0	6	2	3	3	3	4	5	6	7	4	4	1	2	0
McQuitty	0	6	3	3	3	3	5	5	6	7	4	3	1	1	0
kmeans	0	0	1	6	6	10	12	12	2	2	1	1	0	0	0

PAM	0	0	0	0	0	0	0	1	1	3	6	7	9	23	0
CH															
Single	0	10	21	9	6	2	1	0	0	0	0	0	0	0	1
Complete	0	0	3	0	1	0	0	1	0	1	2	2	3	11	26
Average	0	2	19	2	0	0	0	1	1	0	2	2	4	3	14
Ward	0	1	11	2	3	0	1	2	1	2	1	6	4	4	13
McQuitty	0	1	11	2	3	1	1	2	1	2	1	6	4	4	13
kmeans	0	1	2	1	1	0	1	1	1	1	2	10	7	10	12
PAM	0	0	2	0	0	0	0	1	1	0	2	2	5	9	28
KL															
Single	0	7	6	4	5	2	3	5	5	2	6	5	0	0	0
Complete	0	6	6	5	6	3	4	5	5	3	3	2	0	0	0
Average	0	2	2	4	3	6	7	5	5	5	6	4	0	0	0
Ward	0	8	8	6	3	5	3	4	3	3	2	4	0	0	0
McQuitty	0	8	8	6	3	5	3	4	3	5	2	3	0	0	0
kmeans	0	13	4	4	5	5	3	3	3	2	4	2	0	0	0
PAM	0	4	8	3	2	10	5	3	4	3	4	2	0	0	0
Gap															
Single	13	13	24	0	0	0	0	0	0	0	0	0	0	0	0
Complete	3	24	21	2	0	0	0	0	0	0	0	0	0	0	0
Average	3	8	39	0	0	0	0	0	0	0	0	0	0	0	0
Ward	7	23	18	2	0	0	0	0	0	0	0	0	0	0	0
McQuitty	5	23	20	2	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	19	30	1	0	0	0	0	0	0	0	0	0	0	0
PAM	0	6	40	3	1	0	0	0	0	0	0	0	0	0	0
Jump															
p/2	0	0	0	0	0	0	0	1	1	0	2	5	8	15	17
p/3	4	12	19	2	1	0	0	1	0	0	1	2	3	1	4
p/4	37	13	0	0	0	0	0	0	0	0	0	0	0	0	0
p/5	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p/6	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p/7	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PS															
Single	21	27	3	0	0	0	0	0	0	0	0	0	0	0	0
Complete	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	48	2	0	0	0	0	0	0	0	0	0	0	0	0	0
Ward	38	11	0	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	3	31	15	0	0	0	0	0	0	0	0	0	0	0	0
PAM	17	32	0	0	0	0	0	0	0	0	0	0	0	0	0
BI															
Single	0	1	0	0	0	1	3	3	2	4	4	5	3	9	14
Complete	0	0	0	0	0	0	0	0	0	0	0	0	1	5	43
Average	0	0	0	0	0	0	0	1	0	1	1	3	2	11	30
Ward	0	1	0	0	0	0	0	0	0	0	0	1	2	10	38
McQuitty	0	0	0	0	0	0	0	0	0	0	1	1	1	8	39
kmeans	0	27	3	0	0	0	0	0	0	0	0	0	1	7	12
PAM	0	11	0	1	0	0	0	0	1	0	0	1	1	3	32
BIC															
Model-based	0	16	32	2	0	0	0	0	0	0	0	0	0	0	0
ASW															
Single	0	20	16	9	3	1	1	0	0	0	0	0	0	0	0
Complete	0	28	16	4	0	0	1	0	0	0	0	0	0	0	1
Average	0	19	27	4	0	0	0	0	0	0	0	0	0	0	0

Table B.6: Results for the estimation of number of clusters from indices and clustering methods for Model 6.

	PS														
Single	33	12	5	0	0	0	0	0	0	0	0	0	0	0	0
Complete	8	42	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	10	20	20	0	0	0	0	0	0	0	0	0	0	0	0
Ward	10	1	29	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	30	20	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	7	42	1	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	45	5	0	0	0	0	0	0	0	0	0	0	0
	BI														
Single	0	8	6	1	1	0	0	1	3	0	0	6	6	18	
Complete	0	0	0	0	0	0	0	1	1	12	12	5	7	9	4
Average	0	0	3	3	1	0	0	0	1	0	1	5	10	25	
Ward	0	0	26	0	2	0	1	8	3	10	0	0	0	0	0
McQuitty	0	0	0	0	0	0	0	1	2	1	3	3	12	16	13
kmeans	0	27	23	1	0	0	0	0	0	0	0	0	0	0	0
PAM	0	1	44	5	0	0	0	0	0	0	0	0	0	0	0
	BIC														
Model-based	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
	ASW														
Single	1	22	0	0	4	4	2	0	6	6	1	1	1	1	1
Complete	0	0	22	10	6	0	6	0	1	3	0	2	0	0	0
Average	0	1	36	1	2	2	3	0	1	2	1	1	0	1	0
Ward	0	6	23	1	2	2	5	0	1	4	2	1	0	1	0
McQuitty	0	6	23	2	3	2	5	0	1	3	2	1	0	1	0
kmeans	0	0	48	2	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	49	1	0	0	0	0	0	0	0	0	0	0	0
Spectral	0	5	33	9	3	0	0	0	0	0	0	0	0	0	0
Model-based	0	3	38	5	1	0	0	0	0	0	0	0	1	2	0
PAMSIL	0	0	44	3	1	0	1	0	0	1	0	0	0	0	0
HOSil	0	0	44	6	0	0	0	0	0	0	0	0	0	0	0

Table B.7: Results for the estimation of number of clusters from indices and clustering methods for Model 6.A.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	41	4	0	1	1	0	0	0	0	0	0	0	0	0	0
Complete	0	1	0	0	0	1	1	0	5	11	9	6	16	0	0
Average	4	9	7	3	7	5	7	4	3	0	0	1	0	0	0
Ward	1	0	5	5	2	9	8	5	3	6	0	3	2	0	0
McQuitty	1	0	5	5	2	9	8	5	3	6	0	3	2	0	0
kmeans	0	16	9	13	5	4	3	0	0	0	0	0	0	0	0
PAM	0	0	0	0	0	0	2	3	6	11	10	10	8	0	0
CH															
Single	0	41	6	3	0	0	0	0	0	0	0	0	0	0	0
Complete	0	0	0	0	0	1	0	1	4	5	5	3	3	10	18
Average	0	0	1	0	1	0	0	3	0	5	7	5	7	14	7
Ward	0	0	2	1	1	1	0	3	4	3	3	3	3	12	14
McQuitty	0	0	2	1	1	1	0	3	4	3	3	3	3	12	14
kmeans	0	0	4	2	3	1	6	3	0	5	4	11	2	5	4
PAM	0	0	0	0	2	0	2	2	4	5	11	3	6	6	9
KL															

Single	0	3	10	2	5	5	4	4	3	7	4	3	0	0	0
Complete	0	8	11	4	4	1	4	5	4	4	5	0	0	0	0
Average	0	6	4	5	3	4	8	2	9	6	1	2	0	0	0
Ward	0	11	4	2	2	3	5	5	3	6	3	6	0	0	0
McQuitty	0	11	4	2	2	3	5	5	3	6	3	6	0	0	0
<i>k</i> -means	0	2	12	4	7	8	6	6	2	1	1	1	0	0	0
PAM	0	6	8	1	5	0	8	4	8	2	3	5	0	0	0
Gap															
Single	3	43	4	0	0	0	0	0	0	0	0	0	0	0	0
Complete	4	20	22	4	0	0	0	0	0	0	0	0	0	0	0
Average	0	12	37	0	1	0	0	0	0	0	0	0	0	0	0
Ward	3	22	22	3	0	0	0	0	0	0	0	0	0	0	0
McQuitty	3	22	24	1	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	0	44	5	1	0	0	0	0	0	0	0	0	0	0
PAM	0	0	34	12	1	3	0	0	0	0	0	0	0	0	0
Jump															
<i>p</i> /2	0	0	0	0	0	0	0	1	3	2	12	5	8	6	13
<i>p</i> /3	0	1	25	6	1	0	2	1	2	1	3	3	0	3	2
<i>p</i> /4	15	17	18	0	0	0	0	0	0	0	0	0	0	0	0
<i>p</i> /5	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>p</i> /6	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>p</i> /7	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PS															
Single	1	17	26	4	1	0	1	0	0	0	0	0	0	0	0
Complete	15	35	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	0	30	17	2	1	0	0	0	0	0	0	0	0	0	0
Ward	0	7	42	1	0	0	0	0	0	0	0	0	0	0	0
McQuitty	4	44	2	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	2	48	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	1	48	1	0	0	0	0	0	0	0	0	0	0	0
BI															
Single	0	40	2	0	0	1	1	0	0	0	0	0	2	0	4
Complete	0	10	0	0	0	0	0	0	1	2	0	2	14	21	0
Average	0	44	0	0	0	0	0	0	0	0	0	1	0	0	5
Ward	0	37	11	0	0	0	0	0	0	0	1	1	0	0	0
McQuitty	0	20	0	0	0	0	0	0	0	0	0	1	1	4	24
kmeans	0	41	9	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	18	31	0	0	1	0	0	0	0	0	0	0	0	0
BIC															
Model-based	0	0	49	1	0	0	0	0	0	0	0	0	0	0	0
ASW															
Single	0	46	3	1	0	0	0	0	0	0	0	0	0	0	0
Complete	0	36	4	4	0	1	0	0	1	1	2	0	0	1	0
Average	0	47	3	0	0	0	0	0	0	0	0	0	0	0	0
Ward	0	36	7	3	2	0	1	0	0	0	1	0	0	0	0
McQuitty	0	36	7	3	2	0	1	0	0	0	1	0	0	0	0
kmeans	0	43	7	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	40	8	2	0	0	0	0	0	0	0	0	0	0	0
Spectral	0	48	2	0	0	0	0	0	0	0	0	0	0	0	0
Model-based	0	48	2	0	0	0	0	0	0	0	0	0	0	0	0
PAMSIL	0	48	2	0	0	0	0	0	0	0	0	0	0	0	0
HOSil	0	47	2	1	0	0	0	0	0	0	0	0	0	0	0

Table B.8: Results for the estimation of number of clusters from indices and clustering methods for Model 6.B.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	1	37	6	4	1	0	0	1	0	0	0	0	0	0	0
Complete	0	0	0	0	0	0	0	0	0	2	2	46	0	0	0
Average	0	7	5	7	3	5	3	4	7	4	2	0	3	0	0
Ward	0	2	1	1	2	7	5	9	6	5	4	1	7	0	0
McQuitty	0	2	1	1	2	7	5	9	6	5	4	1	7	0	0
kmeans	6	1	17	13	8	4	0	0	1	0	0	0	0	0	0
PAM	0	0	0	0	0	0	1	0	2	6	8	6	27	0	0
CH															
Single	0	0	23	2	13	4	4	0	1	0	1	0	0	1	1
Complete	0	0	0	0	1	0	0	2	2	0	1	5	5	12	22
Average	0	0	0	0	0	0	0	1	3	3	3	3	10	9	18
Ward	0	0	0	0	1	0	1	1	5	3	2	8	5	11	13
McQuitty	0	0	0	0	1	0	1	1	5	3	2	8	5	11	13
kmeans	0	0	0	1	5	3	3	3	5	6	2	8	4	9	1
PAM	0	0	0	0	0	1	0	2	4	8	3	7	8	10	7
KL															
Single	0	3	10	5	8	7	4	5	0	5	0	3	0	0	0
Complete	0	7	6	17	2	3	2	3	1	4	3	2	0	0	0
Average	0	3	6	6	1	2	4	11	3	3	6	5	0	0	0
Ward	0	4	17	6	2	3	3	0	4	5	1	5	0	0	0
McQuitty	0	4	17	6	2	3	3	0	4	5	1	5	0	0	0
kmeans	0	34	1	1	3	4	0	5	1	0	1	0	0	0	0
PAM	0	5	11	4	1	11	5	3	0	6	4	0	0	0	0
Gap															
Single	0	1	47	2	0	0	0	0	0	0	0	0	0	0	0
Complete	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	26	0	7	8	9	0	0	0	0	0	0	0	0	0	0
Ward	34	0	11	3	2	0	0	0	0	0	0	0	0	0	0
McQuitty	31	1	12	3	3	0	0	0	0	0	0	0	0	0	0
kmeans	49	0	0	1	0	0	0	0	0	0	0	0	0	0	0
PAM	45	0	0	0	3	2	0	0	0	0	0	0	0	0	0
Jump															
$p/2$	0	0	0	0	0	1	0	8	2	3	7	7	6	7	9
$p/3$	0	0	2	30	5	0	0	2	0	0	3	2	2	1	3
$p/4$	0	0	32	18	0	0	0	0	0	0	0	0	0	0	0
$p/5$	43	0	6	1	0	0	0	0	0	0	0	0	0	0	0
$p/6$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$p/7$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PS															
Single	3	0	24	10	6	4	3	0	0	0	0	0	0	0	0
Complete	33	1	7	9	0	0	0	0	0	0	0	0	0	0	0
Average	0	0	15	33	2	0	0	0	0	0	0	0	0	0	0
Ward	0	0	4	45	1	0	0	0	0	0	0	0	0	0	0
McQuitty	27	2	14	7	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	0	8	42	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
BI															
Single	0	2	29	0	0	3	1	0	0	0	1	0	1	6	7

Complete	0	0	1	1	0	0	0	0	0	0	0	3	18	27
Average	0	0	13	17	3	0	0	0	0	0	0	1	2	14
Ward	0	0	13	33	0	0	0	0	0	0	0	1	3	0
McQuitty	0	0	2	3	0	0	0	0	1	0	0	1	8	35
kmeans	0	0	9	35	4	0	1	0	0	0	0	1	0	0
PAM	0	0	1	49	0	0	0	0	0	0	0	0	0	0
BIC														
Model-based	0	0	0	50	0	0	0	0	0	0	0	0	0	0
ASW														
Single	0	0	47	2	1	0	0	0	0	0	0	0	0	0
Complete	0	0	23	8	11	1	0	0	1	1	2	2	0	1
Average	0	0	37	13	0	0	0	0	0	0	0	0	0	0
Ward	0	0	35	7	3	0	2	1	0	0	0	1	1	0
McQuitty	0	0	35	7	3	0	2	1	0	0	0	1	1	0
kmeans	0	0	23	23	3	0	0	1	0	0	0	0	0	0
PAM	0	0	28	19	3	0	0	0	0	0	0	0	0	0
Spectral	0	0	31	11	4	1	2	1	0	0	0	0	0	0
Model-based	0	0	38	10	2	0	0	0	0	0	0	0	0	0
PAMSIL	0	0	28	14	7	1	0	0	0	0	0	0	0	0
HOSil	0	0	36	10	4	0	0	0	0	0	0	0	0	0

Table B.9: Results for the estimation of number of clusters from indices and clustering methods for Model 7.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0
Complete	0	0	0	1	1	0	0	3	4	1	7	4	6	23	0
Average	0	0	0	35	4	4	6	0	0	0	0	0	1	0	0
Ward	0	0	1	9	2	5	4	4	4	7	2	3	6	2	0
McQuitty	0	0	1	9	2	5	4	4	4	7	2	3	6	2	0
kmeans	0	0	0	0	8	4	4	9	10	7	4	1	2	1	0
PAM	0	0	0	0	0	0	0	0	0	1	0	2	6	41	0
CH															
Single	0	1	2	0	4	3	4	7	5	7	2	5	5	4	1
Complete	0	0	0	42	7	0	1	0	0	0	0	0	0	0	0
Average	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
Ward	0	0	0	42	4	0	0	0	0	0	0	0	1	1	1
McQuitty	0	0	0	42	4	0	0	0	0	0	0	0	1	1	1
kmeans	0	0	0	48	2	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
KL															
Single	0	0	4	4	3	7	5	7	3	6	7	4	0	0	0
Complete	0	0	24	1	4	2	1	4	5	1	5	3	0	0	0
Average	0	0	15	3	5	5	4	6	5	0	5	2	0	0	0
Ward	0	21	1	3	3	5	4	2	3	1	3	4	0	0	0
McQuitty	0	21	1	3	3	5	4	2	3	1	3	4	0	0	0
kmeans	0	48	2	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	16	0	5	3	0	8	3	2	4	5	4	0	0	0
Gap															
Single	44	3	3	0	0	0	0	0	0	0	0	0	0	0	0
Complete	14	9	0	26	1	0	0	0	0	0	0	0	0	0	0
Average	5	14	0	31	0	0	0	0	0	0	0	0	0	0	0

Ward	19	15	0	16	0	0	0	0	0	0	0	0	0	0	0
McQuitty	18	12	0	20	0	0	0	0	0	0	0	0	0	0	0
kmeans	19	17	0	14	0	0	0	0	0	0	0	0	0	0	0
PAM	27	12	0	11	0	0	0	0	0	0	0	0	0	0	0
Jump															
$p/2$	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
$p/3$	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
$p/4$	29	0	0	21	0	0	0	0	0	0	0	0	0	0	0
$p/5$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$p/6$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$p/7$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PS															
Single	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Complete	13	0	0	37	0	0	0	0	0	0	0	0	0	0	0
Average	1	0	0	49	0	0	0	0	0	0	0	0	0	0	0
Ward	1	0	0	49	0	0	0	0	0	0	0	0	0	0	0
McQuitty	28	0	0	22	0	0	0	0	0	0	0	0	0	0	0
kmeans	1	0	0	49	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
BI															
Single	0	0	1	0	1	1	1	1	4	3	0	4	5	7	22
Complete	0	0	0	49	0	0	0	0	0	0	0	0	0	0	1
Average	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
Ward	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
McQuitty	0	0	0	36	8	0	0	0	0	0	0	1	0	1	4
kmeans	0	0	0	49	1	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
BIC															
Model-based	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
ASW															
Single	0	16	5	4	9	3	4	3	2	1	1	0	2	0	0
Complete	0	0	0	49	1	0	0	0	0	0	0	0	0	0	0
Average	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
Ward	0	0	0	48	1	1	0	0	0	0	0	0	0	0	0
McQuitty	0	0	0	48	1	1	0	0	0	0	0	0	0	0	0
kmeans	0	0	0	48	2	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
Spectral	0	0	1	44	5	0	0	0	0	0	0	0	0	0	0
Model-based	0	0	0	48	2	0	0	0	0	0	0	0	0	0	0
PAMSIL	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
HOSil	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0

Table B.10: Estimation of number of clusters from different indexes and clustering methods for Model 8.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	0	15	10	9	7	4	1	0	0	0	0	0	0	0	0
Complete	0	1	1	3	4	0	1	0	0	2	3	5	7	24	0
Average	0	2	11	8	8	2	4	6	6	7	3	0	2	1	0
Ward	0	1	4	8	4	3	3	6	7	4	3	3	4	1	0
McQuitty	0	1	4	9	13	6	6	6	7	4	3	3	3	2	0
kmeans	0	0	5	16	13	6	6	2	1	0	0	0	0	0	0

PAM	0	0	0	0	0	1	0	3	7	8	10	4	4	1	0
CH															
Single	0	0	1	1	2	4	2	2	3	5	4	6	4	5	10
Complete	0	0	0	0	3	4	7	12	11	8	9	8	12	13	6
Average	0	0	0	0	1	6	6	5	8	7	5	6	3	1	0
Ward	0	0	0	0	1	5	3	7	7	4	6	5	2	2	5
McQuitty	0	0	0	0	2	6	3	7	7	4	6	5	2	2	5
kmeans	0	0	0	1	7	8	12	5	3	2	1	1	2	3	3
PAM	0	0	0	0	3	10	5	3	2	1	1	4	3	6	12
KL															
Single	0	3	4	5	5	7	6	5	2	5	3	3	0	0	0
Complete	0	6	5	7	12	6	5	2	2	1	1	0	0	0	0
Average	0	12	1	9	10	5	5	4	2	2	0	1	0	0	0
Ward	0	2	6	14	10	6	5	1	2	1	0	2	0	0	0
McQuitty	0	3	7	14	10	6	5	1	1	1	0	1	0	0	0
kmeans	0	13	14	10	4	3	1	1	1	2	1	0	0	0	0
PAM	0	3	9	8	2	1	5	3	3	6	5	4	0	0	0
Gap															
Single	4	25	18	2	0	0	0	0	0	0	0	0	0	0	0
Complete	1	7	28	10	2	2	1	0	0	0	0	0	0	0	0
Average	3	12	20	10	3	2	0	1	0	0	0	0	0	0	0
Ward	5	10	20	9	3	2	0	1	0	0	0	0	0	0	0
McQuitty	5	11	20	8	2	2	1	0	0	0	0	0	0	0	0
kmeans	0	0	16	16	7	3	1	0	0	0	0	0	0	0	0
PAM	0	4	0	1	8	27	10	1	0	0	0	0	0	0	0
Jump															
$p/2$	0	0	0	0	8	3	1	1	1	1	1	6	7	13	8
$p/3$	0	0	0	10	25	4	1	1	1	1	1	1	2	2	2
$p/4$	0	0	0	22	24	2	0	1	0	0	0	0	1	1	0
$p/5$	0	3	0	26	18	2	0	0	0	0	0	0	0	0	0
$p/6$	42	0	0	0	26	18	0	0	0	0	0	0	0	0	0
$p/7$	49	0	0	1	0	0	0	0	0	0	0	0	0	0	0
PS															
Single	10	16	6	2	0	0	0	0	0	0	0	0	0	0	0
Complete	17	3	30	1	0	0	0	0	0	0	0	0	0	0	0
Average	1	23	25	2	0	0	0	0	0	0	0	0	0	0	0
Ward	0	5	44	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	25	1	22	1	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	46	8	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	2	46	2	0	0	0	0	0	0	0	0	0	0	0
BI															
Single	0	6	1	0	0	0	0	0	0	0	1	1	2	6	34
Complete	0	2	10	0	0	1	3	11	7	2	3	1	2	2	5
Average	0	15	3	0	0	1	1	1	7	8	3	3	3	1	2
Ward	0	29	6	1	1	1	2	6	13	11	2	0	1	1	1
McQuitty	0	0	8	0	0	0	1	6	13	11	3	2	1	2	3
kmeans	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	22	16	7	1	0	1	2	1	0	1	1	0	0	0
ASW															
Single	0	42	3	1	1	1	1	1	1	1	0	0	0	0	0
Complete	0	30	0	1	4	7	3	2	1	1	1	1	0	0	0
Average	0	37	0	1	5		5	0	1	1	1	0	0	0	0
Ward	0	28	0	2	3	5	8	4	5	2	0	1	1	0	0
McQuitty	0	28	0	2	3	5	4	2	3	1	0	1	1	0	0

kmeans	0	38	0	5	6	0	1	0	0	0	0	0	0	0	0
PAM	0	28	0	8	10	2	1	0	0	0	0	0	0	0	0
Spectral	0	34	0	2	7	7	0	0	0	0	0	0	0	0	0
Model-based	0	32	0	6	5	5	0	2	0	0	0	0	0	0	0
BIC															
Model-based	0	0	0	33	15	2	0	0	0	0	0	0	0	0	0
PAMSIL	0	28	0	9	8	3	2	0	0	0	0	0	0	0	0
HOSil	0	34	0	5	10	1	0	1	0	0	0	0	0	0	0

Table B.11: Results for the estimation of number of clusters from indices and clustering methods for Model 9.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	20	6	6	1	0	0	0	0	0	0	0	0	0	0	0
Average	0	0	18	4	3	4	3	4	6	3	2	0	3	0	0
Complete	0	0	0	0	0	0	0	3	2	5	8	6	26	0	0
Ward	1	3	5	1	2	3	8	9	5	6	4	1	2	0	0
McQuitty	1	3	5	1	2	3	8	9	5	6	4	1	2	0	0
kmeans	0	2	0	4	5	13	10	10	3	1	2	0	0	0	0
PAM	0	0	0	0	0	0	0	1	0	4	2	6	35	0	0
CH															
Single	0	4	2	7	11	5	10	4	2	1	1	2	0	1	0
Complete	0	0	0	46	0	0	1	1	0	1	0	0	0	1	0
Average	0	0	0	13	4	3	1	8	5	3	0	7	1	0	5
Ward	0	0	0	14	3	2	3	10	4	2	2	4	2	0	4
McQuitty	0	0	0	14	3	2	3	10	4	2	2	4	2	0	4
kmeans	0	0	0	35	4	0	1	3	1	3	0	2	1	0	0
PAM	0	0	0	30	1	0	1	9	3	2	2	1	1	0	0
KL															
Single	0	4	8	5	11	5	2	5	2	4	2	2	0	0	0
Complete	0	0	18	7	3	5	4	3	2	2	4	2	0	0	0
Average	0	0	14	6	8	4	7	4	1	0	3	3	0	0	0
Ward	0	14	8	6	3	2	2	3	5	2	2	3	0	0	0
McQuitty	0	14	8	6	3	2	2	3	5	2	2	3	0	0	0
kmeans	0	45	5	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	34	3	0	0	0	6	0	0	2	2	1	20	0	0
Gap															
Single	17	21	6	6	0	0	0	0	0	0	0	0	0	0	0
Complete	20	23	1	2	2	2	0	0	0	0	0	0	0	0	0
Average	20	28	0	2	0	0	0	0	0	0	0	0	0	0	0
Ward	21	28	0	0	0	0	0	1	0	0	0	0	0	0	0
McQuitty	21	29	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	23	2	23	2	0	0	0	0	0	0	0	0	0	0
PAM	1	21	0	20	8	0	0	0	0	0	0	0	0	0	0
Jump															
p/2	0	0	0	49	0	0	0	1	0	0	0	0	0	0	0
p/3	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
p/4	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
p/5	36	0	0	14	0	0	0	0	0	0	0	0	0	0	0
p/6	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p/7	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	PS														
Single	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Complete	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ward	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	15	0	35	0	0	0	0	0	0	0	0	0	0	0
PAM	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	BI														
Single	0	0	0	0	0	0	0	0	0	1	0	2	11	36	
Complete	0	0	0	0	0	0	0	0	0	0	0	0	2	48	
Average	0	0	0	0	0	0	0	0	0	0	0	0	7	43	
Ward	0	0	0	0	0	0	0	0	0	0	1	1	5	43	
McQuitty	0	0	0	0	0	0	0	0	0	0	0	0	2	48	
kmeans	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	0	0	0	0	0	0	0	0	0	8	42	
	BIC														
Model-based	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
	ASW														
Single	0	17	6	9	8	3	2	3	1	1	0	0	0	0	0
Complete	0	10	0	40	0	0	0	0	0	0	0	0	0	0	0
Average	0	26	5	18	0	1	0	0	0	0	0	0	0	0	0
Ward	0	26	7	16	1	0	0	0	0	0	0	0	0	0	0
McQuitty	0	26	7	16	1	0	0	0	0	0	0	0	0	0	0
kmeans	0	11	0	39	0	0	0	0	0	0	0	0	0	0	0
PAM	0	7	0	43	0	0	0	0	0	0	0	0	0	0	0
Spectral	0	2	6	36	5	1	0	0	0	0	0	0	0	0	0
Model-based	0	0	0	47	3	0	0	0	0	0	0	0	0	0	0
PAMSIL	0	5	0	45	0	0	0	0	0	0	0	0	0	0	0
HOSil	0	10	0	40	0	0	0	0	0	0	0	0	0	0	0

Table B.12: Results for the estimation of number of clusters from indices and clustering methods for Model 10.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	0	1	32	12	3	1	1	0	0	0	0	0	0	0	0
Complete	1	0	1	0	0	0	0	0	0	0	1	2	45	0	0
Average	1	0	1	1	11	6	7	5	3	2	4	4	5	0	0
Ward	0	0	1	1	1	1	1	3	4	3	3	8	24	0	0
McQuitty	0	0	1	1	1	1	1	3	4	3	3	8	24	0	0
kmeans	0	1	9	23	16	1	0	0	0	0	0	0	0	0	0
PAM	0	0	0	0	0	0	0	0	0	0	0	1	49	0	0
CH															
Single	0	0	0	6	9	8	11	4	4	3	1	2	0	1	1
Complete	0	0	0	0	0	0	1	0	1	0	2	3	3	6	34
Average	0	0	0	0	6	1	0	1	1	1	3	4	5	7	21
Ward	0	0	0	0	3	0	1	2	3	3	4	3	8	7	16
McQuitty	0	0	0	0	3	0	1	2	3	3	4	3	8	7	16
kmeans	0	0	0	0	11	6	3	6	4	5	3	4	4	2	2
PAM	0	0	0	0	0	0	1	0	0	3	1	1	5	17	22
KL															

Single	0	5	9	9	9	6	5	1	1	2	2	1	0	0	0
Complete	0	0	0	32	11	1	2	0	1	2	1	0	0	0	0
Average	0	0	0	38	3	1	2	0	0	3	0	3	0	0	0
Ward	0	0	36	3	1	4	1	1	0	1	3	0	0	0	0
McQuitty	0	0	36	3	1	4	1	1	0	1	3	0	0	0	0
kmeans	0	0	29	9	6	3	1	0	0	1	0	1	0	0	0
PAM	0	0	34	2	4	0	4	0	1	1	2	2	0	0	0
Gap															
Single	0	0	1	40	9	0	0	0	0	0	0	0	0	0	0
Complete	0	6	4	4	26	6	3	1	0	0	0	0	0	0	0
Average	0	1	0	1	45	3	0	0	0	0	0	0	0	0	0
Ward	0	11	1	2	29	7	0	0	0	0	0	0	0	0	0
McQuitty	1	12	1	2	30	4	0	0	0	0	0	0	0	0	0
kmeans	4	2	1	8	29	6	0	0	0	0	0	0	0	0	0
PAM	10	0	0	0	24	5	9	1	1	0	0	0	0	0	0
Jump															
$p/2$	0	0	0	0	28	0	0	0	0	1	5	4	5	7	
$p/3$	0	0	0	0	49	0	0	0	0	0	0	0	1	0	
$p/4$	0	0	0	0	50	0	0	0	0	0	0	0	0	0	
$p/5$	0	0	0	0	50	0	0	0	0	0	0	0	0	0	
$p/6$	1	0	0	0	49	0	0	0	0	0	0	0	0	0	
$p/7$	43	0	0	0	7	0	0	0	0	0	0	0	0	0	
PS															
Single	0	0	0	42	6	2	0	0	0	0	0	0	0	0	0
Complete	7	0	0	0	43	0	0	0	0	0	0	0	0	0	0
Average	0	0	0	0	43	7	0	0	0	0	0	0	0	0	0
Ward	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0
McQuitty	8	0	0	0	40	2	0	0	0	0	0	0	0	0	0
kmeans	5	42	0	0	3	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0
BI															
Single	0	0	0	37	3	5	2	0	0	1	0	0	0	0	2
Complete	0	0	0	0	24	2	0	0	0	0	0	0	4	3	17
Average	0	0	0	6	43	1	0	0	0	0	0	0	0	0	0
Ward	0	0	0	5	45	0	0	0	0	0	0	0	0	0	0
McQuitty	0	0	0	0	31	3	0	1	0	0	0	0	0	2	13
kmeans	0	2	0	1	9	19	8	8	0	1	0	1	0	0	1
PAM	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0
BIC															
Model-based	0	0	0	1	17	19	10	3	0	0	0	0	0	0	0
ASW															
Single	0	0	0	24	10	8	6	1	1	0	0	0	0	0	0
Complete	0	0	0	0	36	12	1	0	0	0	0	0	1	0	0
Average	0	0	0	0	48	2	0	0	0	0	0	0	0	0	0
Ward	0	0	0	0	44	4	1	1	0	0	0	0	0	0	0
McQuitty	0	0	0	0	44	4	1	1	0	0	0	0	0	0	0
kmeans	0	0	0	4	38	4	3	1	0	0	0	0	0	0	0
PAM	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0
Spectral	0	0	1	7	20	13	2	2	3	1	1	0	0	0	0
Model-based	0	0	0	0	49	1	0	0	0	0	0	0	0	0	0
PAMSIL	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0
HOSil	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0

Table B.13: Results for the estimation of number of clusters from indices and clustering methods for Model 11.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	14	8	9	7	4	2	2	1	0	0	0	0	0	0	0
Complete	0	2	3	11	16	6	0	0	0	0	1	1	10	0	0
Average	5	2	10	15	6	0	1	3	4	1	1	2	0	0	0
Ward	4	4	11	13	14	2	1	0	0	1	0	0	0	0	0
McQuitty	4	4	11	13	14	2	1	0	0	1	0	0	0	0	0
kmeans	0	7	14	11	9	6	2	1	0	0	0	0	0	0	0
PAM	0	0	0	11	0	0	0	0	1	0	1	0	37	0	0
CH															
Single	0	0	8	10	4	1	1	1	2	1	5	0	7	9	
Complete	0	0	0	0	0	0	0	8	10	6	4	3	6	9	4
Average	0	0	0	0	0	0	0	3	13	8	9	5	1	5	6
Ward	0	0	0	0	0	0	0	2	10	7	8	6	6	6	5
McQuitty	0	0	0	0	0	0	0	2	10	7	8	6	6	6	5
kmeans	0	0	0	0	4	5	11	8	7	5	3	2	3	1	1
PAM	0	0	0	0	0	0	0	1	0	1	3	7	12	12	14
KL															
Single	0	3	5	4	4	4	4	8	3	5	5	5	0	0	0
Complete	0	0	0	0	0	0	9	21	14	4	2	0	0	0	0
Average	0	5	1	0	1	6	10	11	7	7	1	1	0	0	0
Ward	0	0	0	0	2	4	7	15	9	9	2	2	0	0	0
McQuitty	0	0	0	0	2	4	7	15	9	9	2	2	0	0	0
kmeans	0	2	6	6	3	6	4	5	5	5	6	2	0	0	0
PAM	0	0	0	1	17	12	2	3	2	3	4	6	0	0	0
Gap															
Single	11	17	11	9	2	0	0	0	0	0	0	0	0	0	0
Complete	0	39	11	0	0	0	0	0	0	0	0	0	0	0	0
Average	2	17	31	0	0	0	0	0	0	0	0	0	0	0	0
Ward	1	39	10	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	1	39	10	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	9	0	17	19	5	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	16	32	2	0	0	0	0	0	0	0	0	0
Jump															
p/2	0	0	0	0	0	19	1	0	0	1	1	7	5	9	7
p/3	0	0	0	0	0	40	1	0	0	1	1	0	2	2	3
p/4	9	0	0	0	0	37	1	0	0	1	0	0	0	0	2
p/5	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p/6	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p/7	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PS															
Single	18	10	15	5	2	0	0	0	0	0	0	0	0	0	0
Complete	0	3	47	0	0	0	0	0	0	0	0	0	0	0	0
Average	0	0	46	4	0	0	0	0	0	0	0	0	0	0	0
Ward	0	0	0	44	6	0	0	0	0	0	0	0	0	0	0
McQuitty	0	4	45	1	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	0	4	45	1	0	0	0	0	0	0	0	0
BI															
Single	0	8	2	0	0	3	6	5	6	5	3	3	3	2	4

Complete	0	36	7	0	1	0	2	0	0	0	0	1	1	1	1
Average	0	42	5	0	0	0	0	0	2	0	0	0	0	0	1
Ward	0	39	0	11	0	0	0	0	0	0	0	0	0	0	0
McQuitty	0	26	2	6	2	0	1	3	1	1	0	0	0	4	4
kmeans	0	1	0	0	1	0	3	4	4	6	5	11	7	4	4
PAM	0	0	0	23	16	8	2	0	1	0	0	0	0	0	0
BIC															
Model-based	0	0	0	0	45	4	0	0	1	0	0	0	0	0	0
ASW															
Single	0	37	6	1	0	1	0	1	0	1	0	0	1	2	0
Complete	0	6	0	0	0	0	0	3	13	12	7	7	2	0	0
Average	0	8	0	0	0	0	0	3	9	5	7	8	4	3	3
Ward	0	10	0	0	0	0	0	1	4	4	11	6	5	7	2
McQuitty	0	10	0	0	0	0	0	1	4	4	11	6	5	7	2
kmeans	0	9	0	0	11	17	6	3	3	1	0	0	0	0	0
PAM	0	0	0	0	2	40	7	0	1	0	0	0	0	0	0
Spectral	0	6	0	0	15	20	7	1	1	0	0	0	0	0	0
Model-based	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0
PAMSIL	0	0	0	0	0	47	3	0	0	0	0	0	0	0	0
HOSil	0	0	0	0	1	48	0	1	0	0	0	0	0	0	0

Table B.14: Results for the estimation of number of clusters from indices and clustering methods for Model 12.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	11	7	15	12	4	1	0	0	0	0	0	0	0	0	0
Complete	0	0	0	0	0	1	2	1	3	0	1	3	39	0	0
Average	0	6	1	2	18	5	3	3	2	0	4	2	4	0	0
Ward	0	1	2	0	3	4	1	5	4	1	5	3	21	0	0
McQuitty	0	1	2	0	3	4	1	5	4	1	5	3	21	0	0
kmeans	0	0	4	5	10	14	11	2	3	0	0	0	1	0	0
PAM	0	0	0	0	0	0	0	0	1	0	0	0	49	0	0
CH															
Single	0	0	0	3	9	5	7	6	6	1	5	6	0	2	0
Complete	0	0	0	0	0	0	0	0	0	0	0	5	3	4	38
Average	0	0	0	0	1	19	0	2	0	1	1	0	4	13	9
Ward	0	0	0	0	1	2	1	2	0	3	1	5	3	11	21
McQuitty	0	0	0	0	1	2	1	2	0	3	1	5	3	11	21
kmeans	0	0	0	0	0	2	1	1	2	3	6	4	7	10	14
PAM	0	0	0	0	0	0	0	0	1	0	1	1	3	11	33
KL															
Single	0	9	7	4	5	3	4	2	5	6	2	3	0	0	0
Complete	0	10	2	8	1	5	2	8	3	5	4	2	0	0	0
Average	0	9	5	1	5	4	4	7	3	4	6	2	0	0	0
Ward	0	4	3	9	8	7	2	3	1	6	4	3	0	0	0
McQuitty	0	4	3	9	8	7	2	3	1	6	4	3	0	0	0
kmeans	0	7	13	8	1	1	1	6	3	5	3	2	0	0	0
PAM	0	0	1	18	4	1	0	11	1	2	7	5	0	0	0
Gap															
Single	0	13	21	12	2	2	0	0	0	0	0	0	0	0	0
Complete	0	0	25	13	5	5	2	0	0	0	0	0	0	0	0
Average	0	0	7	8	3	32	0	0	0	0	0	0	0	0	0

Ward	0	0	19	14	4	9	3	0	1	0	0	0	0	0	0
McQuitty	0	0	18	14	8	8	2	0	0	0	0	0	0	0	0
kmeans	0	2	18	10	15	5	0	0	0	0	0	0	0	0	0
PAM	0	0	0	0	6	36	8	0	0	0	0	0	0	0	0
Jump															
$p/2$	0	0	0	0	0	2	0	0	0	4	6	7	2	11	18
$p/3$	0	0	5	0	34	6	0	0	0	1	2	0	1	1	0
$p/4$	0	0	42	0	8	0	0	0	0	0	0	0	0	0	0
$p/5$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$p/6$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$p/7$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PS															
Single	0	39	8	2	1	0	0	0	0	0	0	0	0	0	0
Complete	0	7	43	0	0	0	0	0	0	0	0	0	0	0	0
Average	0	0	17	21	6	6	0	0	0	0	0	0	0	0	0
Ward	0	0	1	24	10	15	0	0	0	0	0	0	0	0	0
McQuitty	0	21	29	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	3	31	16	0	0	0	0	0	0	0	0	0
BI															
Single	0	48	0	0	0	2	0	0	0	0	0	0	0	0	0
Complete	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	0	30	17	3	0	0	0	0	0	0	0	0	0	0	0
Ward	0	36	8	6	0	0	0	0	0	0	0	0	0	0	0
McQuitty	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	0	45	0	0	0	0	0	0	0	0	0	0	1	4
PAM	0	41	0	0	9	0	0	0	0	0	0	0	0	0	0
BIC															
Model-based	0	0	0	0	0	36	12	1	1	0	0	0	0	0	0
ASW															
Single	0	33	2	7	4	2	2	0	0	0	0	0	0	0	0
Complete	0	27	14	3	6	0	0	0	0	0	0	0	0	0	0
Average	0	1	8	2	36	3	0	0	0	0	0	0	0	0	0
Ward	0	25	8	2	15	0	0	0	0	0	0	0	0	0	0
McQuitty	0	25	8	2	15	0	0	0	0	0	0	0	0	0	0
kmeans	0	1	7	5	33	4	0	0	0	0	0	0	0	0	0
PAM	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0
Spectral	0	12	1	5	4	4	2	0	1	0	0	0	0	0	0
Model-based	0	1	0	0	28	0	0	0	0	0	0	0	0	0	0
PAMSIL	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0
HOSil	0	0	0	5	45	0	0	0	0	0	0	0	0	0	0

Table B.15: Results for the estimation of number of clusters from indices and clustering methods for Model 13.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	0	0	0	3	0	0	0	0	0	0	0	1	21	0	0
Complete	0	0	0	0	0	0	0	0	7	2	2	1	13	0	0
Average	0	0	0	0	0	0	0	0	2	0	0	0	23	0	0
Ward	0	0	0	0	0	0	0	0	6	0	2	0	17	0	0
McQuitty	0	0	0	0	0	0	0	0	6	0	2	0	17	0	0
kmeans	0	0	1	5	2	3	3	5	2	3	0	0	1	0	0

PAM	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0
CH															
Single	0	0	0	0	0	0	0	0	0	0	0	0	0	24	1
Complete	0	0	0	0	0	0	0	0	8	0	1	0	0	16	
Average	0	0	0	0	0	0	0	0	0	0	0	0	0	23	2
Ward	0	0	0	0	0	0	0	0	0	0	0	0	0	6	19
McQuitty	0	0	0	0	0	0	0	0	0	0	0	0	0	6	19
kmeans	0	0	0	0	0	0	0	0	2	3	2	6	6	12	
PAM	0	0	0	0	0	0	0	0	0	0	0	0	2	23	
KL															
Single	0	3	1	1	2	2	3	1	0	0	0	12	0	0	0
Complete	0	2	0	3	11	0	0	0	7	0	2	0	0	0	0
Average	0	2	0	6	15	0	0	0	2	0	0	0	0	0	0
Ward	0	0	1	8	0	0	0	1	0	0	2	13	0	0	0
McQuitty	0	0	1	8	0	0	0	1	0	0	2	13	0	0	0
kmeans	0	0	0	4	6	0	0	0	1	5	5	4	0	0	0
PAM	0	0	4	2	0	0	3	0	0	0	0	16	0	0	0
Gap															
Single	0	3	6	7	5	4	0	0	0	0	0	0	0	0	0
Complete	5	18	1	1	0	0	0	0	0	0	0	0	0	0	0
Average	0	24	0	1	0	0	0	0	0	0	0	0	0	0	0
Ward	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	3	17	4	1	0	0	0	0	0	0	0	0	0	0	0
PAM	0	24	0	0	1	0	0	0	0	0	0	0	0	0	0
Jump															
$p/2$	0	0	0	0	0	0	0	0	0	0	0	0	12	13	
$p/3$	0	0	0	0	0	0	0	0	0	0	0	0	0	12	13
$p/4$	0	0	0	0	0	0	0	0	0	0	0	0	0	12	13
$p/5$	10	0	0	0	0	0	0	0	0	0	0	0	0	5	10
$p/6$	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$p/7$	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PS															
Single	4	21	0	0	0	0	0	0	0	0	0	0	0	0	0
Complete	7	18	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0
Ward	3	22	0	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	12	13	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	13	12	0	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0
BI															
Single	0	17	0	0	0	0	0	0	0	0	0	0	1	7	
Complete	0	8	0	4	0	0	0	0	0	3	1	0	1	8	
Average	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0
Ward	0	2	0	0	0	0	0	0	0	0	0	0	0	0	23
McQuitty	0	2	0	3	0	1	1	0	5	4	2	2	0	3	2
kmeans	0	0	0	0	0	0	0	0	0	0	0	0	3	1	21
PAM	0	13	0	0	0	0	0	0	0	0	0	0	0	1	11
BIC															
Model-based	0	0	0	0	0	0	0	5	20	0	0	0	0	0	0
ASW															
Single	0	0	0	0	0	0	0	0	0	0	0	0	23	2	
Complete	0	0	0	0	0	0	0	0	10	0	3	1	2	9	
Average	0	0	0	0	0	0	0	0	0	0	0	0	22	3	

Ward	0	0	0	0	0	0	0	0	0	0	0	0	11	14
McQuitty	0	0	0	0	0	0	0	0	0	0	0	0	11	14
kmeans	0	0	0	0	0	0	0	0	0	2	6	0	7	10
PAM	0	0	0	0	0	0	0	0	0	0	0	0	9	16
Spectral	0	0	0	19	11	13	2	5	0	0	0	0	0	0
Model-based	0	0	0	40	5	0	0	0	0	0	0	1	0	4
PAMSIL	0	0	0	0	0	0	0	0	0	0	0	1	1	23
HOSil	0	0	0	0	0	0	0	0	0	0	0	0	21	3

Table B.16: Results for the estimation of number of clusters from indices and clustering methods for Model 14.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
H																
Single	0	26	0	0	11	0	0	8	0	0	4	1	0	0	0	
Complete	0	0	0	0	0	0	0	2	0	0	4	44	0	0	0	
Average	0	1	0	0	4	0	0	4	0	0	9	32	0	0	0	
Ward	0	1	0	0	1	0	0	3	0	0	13	32	0	0	0	
McQuitty	0	1	0	0	1	0	0	3	0	0	13	32	0	0	0	
kmeans	0	0	9	13	13	10	3	1	0	0	1	0	0	0	0	
PAM	0	0	0	0	0	0	0	0	0	2	1	1	46	0	0	
CH																
Single	0	0	19	0	0	3	0	0	6	0	0	9	0	0	13	
Complete	0	0	0	0	0	0	0	0	2	0	0	6	0	0	42	
Average	0	0	0	0	0	0	0	0	1	0	0	11	0	0	38	
Ward	0	0	0	0	0	0	0	0	2	0	0	13	0	0	35	
McQuitty	0	0	0	0	0	0	0	0	2	0	0	13	0	0	35	
kmeans	0	0	0	0	0	0	0	0	1	0	1	3	10	15	20	
PAM	0	0	0	0	0	0	0	0	0	0	0	5	1	0	44	
KL																
Single	0	4	5	8	1	2	13	0	0	17	0	0	0	0	0	
Complete	0	49	0	0	1	0	0	0	0	0	0	0	0	0	0	
Average	0	45	0	0	3	0	0	1	0	0	1	0	0	0	0	
Ward	0	0	0	17	0	0	13	0	0	20	0	0	0	0	0	
McQuitty	0	0	0	17	0	0	13	0	0	20	0	0	0	0	0	
kmeans	0	0	0	2	4	5	7	8	5	6	8	5	0	0	0	
PAM	0	0	0	37	0	0	10	0	0	3	0	0	0	0	0	
Gap																
Single	0	0	27	0	0	14	0	0	8	0	0	1	0	0	0	
Complete	0	0	28	0	0	20	0	0	1	0	0	1	0	0	0	
Average	0	0	41	0	0	7	0	0	2	0	0	0	0	0	0	
Ward	0	0	34	0	0	12	0	0	2	0	0	1	0	1	0	
McQuitty	0	0	41	0	0	7	0	0	2	0	0	0	0	0	0	
kmeans	0	0	33	5	11	1	0	0	0	0	0	0	0	0	0	
PAM	0	0	28	0	0	13	0	0	2	0	0	4	0	3	0	
Jump																
p/2	0	0	0	0	0	0	0	0	0	7	0	0	11	2	4	33
p/3	0	0	0	0	0	0	0	0	3	0	0	16	4	0	27	
p/4	0	0	0	0	0	1	0	0	8	0	0	18	4	0	19	
p/5	0	0	7	0	0	7	0	0	10	0	0	12	2	0	12	
p/6	0	0	35	0	0	7	0	0	4	0	0	2	2	0	0	
p/7	0	0	47	0	0	2	0	0	1	0	0	0	0	0	0	

	PS														
Single	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
Complete	0	0	49	0	0	1	0	0	0	0	0	0	0	0	0
Average	0	0	48	0	0	2	0	0	0	0	0	0	0	0	0
Ward	0	0	47	0	0	3	0	0	0	0	0	0	0	0	0
McQuitty	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	46	0	0	4	0	0	0	0	0	0	0	0	0
	BI														
Single	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
Complete	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
Average	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
Ward	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
	BIC														
Model-based	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ASW														
Single	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
Complete	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
Average	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
Ward	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
Spectral	0	3	39	4	4	0	0	0	0	0	0	0	0	0	0
Model-based	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
PAMSIL	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
HOSil	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0

Table B.17: Results for the estimation of number of clusters from indices and clustering methods for Model 15.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	0	0	0	0	0	0	0	36	10	2	2	0	0	0	0
Complete	0	0	0	0	0	0	0	0	0	4	4	18	24	0	0
Average	0	0	0	0	0	0	0	18	18	6	2	0	6	0	0
Ward	0	0	0	0	0	0	0	8	10	10	10	4	8	0	0
Mcquitty	0	0	0	0	0	0	0	8	10	10	10	4	8	0	0
<i>k</i> -means	0	0	4	4	14	6	14	6	2	0	0	0	0	0	0
pam	0	0	0	0	0	0	0	0	0	2	4	6	38	0	0
CH															
Single	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Complete	0	0	0	0	0	0	0	0	36	12	0	2	0	0	0
Average	0	0	0	0	0	0	0	0	46	2	2	0	0	0	0
Ward	0	0	0	0	0	0	0	0	40	4	6	0	0	0	0
Mcquitty	0	0	0	0	0	0	0	0	40	4	6	0	0	0	0
<i>k</i> -means	0	0	0	0	0	0	0	0	2	4	10	16	4	10	4
pam	0	0	0	0	0	0	0	0	24	0	8	2	8	6	2
KL															

	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0
Single	0	0	0	0	0	0	0	44	2	2	2	0	0	0	0
Complete	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
Average	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
Ward	0	0	0	0	0	0	44	4	0	0	2	0	0	0	0
Mcquitty	0	0	0	0	0	0	44	2	0	0	4	0	0	0	0
<i>k</i> -means	0	0	0	0	0	0	2	2	16	18	8	4	0	0	0
pam	0	4	0	0	0	0	40	0	2	0	2	2	0	0	0
	Gap														
Single	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
Complete	8	0	2	0	0	1	0	0	36	2	0	0	0	0	0
Average	26	0	0	0	0	0	0	0	24	0	0	0	0	0	0
Ward	16	0	0	0	0	0	0	0	30	4	0	0	0	0	0
Mcquitty	12	0	0	0	0	0	0	0	38	0	0	0	0	0	0
<i>k</i> -means	28	0	4	6	4	6	2	0	0	0	0	0	0	0	0
pam	12	0	0	30	2	0	0	0	6	0	0	0	0	0	0
	Jump														
<i>p</i> /2	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
<i>p</i> /3	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
<i>p</i> /4	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
<i>p</i> /5	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
<i>p</i> /6	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
<i>p</i> /7	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
	PS														
Single	2	0	0	0	0	0	0	0	46	2	0	0	0	0	0
Complete	30	0	0	0	0	0	0	2	18	0	0	0	0	0	0
Average	0	0	0	0	0	0	0	2	48	0	0	0	0	0	0
Ward	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Mcquitty	2	0	0	0	0	0	0	0	48	0	0	0	0	0	0
<i>k</i> -means	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
pam	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
	BI														
Single	0	0	0	0	0	0	0	2	44	2	2	0	0	0	0
Complete	0	0	0	0	0	0	0	0	6	12	18	12	2	0	0
Average	0	0	0	0	0	0	0	0	48	2	0	0	0	0	0
Ward	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Mcquitty	0	0	0	0	0	0	0	0	30	16	2	0	2	0	0
<i>k</i> -means	0	0	0	0	0	0	0	0	0	0	0	4	8	14	24
pam	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
	BIC														
Model-based	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
	ASW														
Single	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Complete	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Average	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Ward	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Mcquitty	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
<i>k</i> -means	0	0	0	0	0	0	4	16	4	6	12	8	0	0	0
pam	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Spectral	0	0	0	0	0	0	0	5	13	15	13	3	0	1	0
Model-based	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
PAMSIL	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
HOSil	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0

Table B.18: Results for the estimation of number of clusters from indices and clustering methods for Model 16.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	0	0	0	0	0	0	0	0	4	14	11	10	11	0	0
Complete	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0
Average	0	0	0	0	0	0	0	0	0	0	5	6	39	0	0
Ward	0	0	0	0	0	0	0	0	0	0	1	1	48	0	0
McQuitty	0	0	0	0	0	0	0	0	0	1	1	48	0	0	0
kmeans	0	5	20	12	7	3	3	0	0	0	0	0	0	0	0
PAM	0	0	0	0	0	0	0	0	0	1	0	49	0	0	0
CH															
Single	0	0	0	0	0	0	0	0	20	5	7	5	6	7	
Complete	0	0	0	0	0	0	0	0	0	0	0	1	1	48	
Average	0	0	0	0	0	0	0	0	0	2	0	3	3	11	31
Ward	0	0	0	0	0	0	0	0	0	0	0	3	4	8	35
McQuitty	0	0	0	0	0	0	0	0	0	0	0	3	4	8	35
kmeans	0	0	0	0	0	0	0	0	0	2	8	5	11	10	14
PAM	0	0	0	0	0	0	0	0	0	0	0	0	0	3	47
KL															
Single	0	1	0	0	0	0	0	48	1	0	0	0	0	0	0
Complete	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Average	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Ward	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
McQuitty	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
kmeans	0	0	6	7	4	3	1	2	6	2	10	9	0	0	0
PAM	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
Gap															
Single	0	0	0	0	0	0	0	0	0	0	2	1	3	44	0
Complete	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
Average	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
Ward	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
McQuitty	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
kmeans	0	0	0	15	16	9	5	5	0	0	0	0	0	0	0
PAM	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
Jump															
p/2	0	0	0	0	0	0	0	0	0	26	13	4	4	0	3
p/3	0	0	0	0	0	0	0	0	1	0	1	2	6	40	
p/4	0	0	0	0	0	0	0	0	0	0	0	0	0	4	46
p/5	0	0	0	0	0	0	0	0	0	0	0	0	0	4	46
p/6	0	0	0	0	0	0	0	0	0	0	0	0	0	4	46
p/7	0	0	0	0	0	0	0	0	0	0	0	0	0	5	45
PS															
Single	0	0	0	0	0	0	0	0	0	46	3	1	0	0	0
Complete	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0
Average	0	0	0	0	0	0	0	0	0	46	3	1	0	0	0
Ward	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0
McQuitty	0	0	0	0	0	0	0	0	0	48	2	0	0	0	0
kmeans	0	47	3	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0
BI															
Single	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0

Complete	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Ward	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
BIC															
Model-based	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
ASW															
Single	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Complete	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Average	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Ward	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
McQuitty	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
kmeans	0	0	0	0	1	2	4	5	9	15	7	5	2	0	0
PAM	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
Spectral	0	0	0	0	0	0	4	7	5	11	12	8	2	1	
Model-based	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
PAMSIL	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
HOSil	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0

Table B.19: Results for the estimation of number of clusters from indices and clustering methods for Model 17.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H															
Single	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Complete	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Ward	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	15	35	0	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
CH															
Single	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Complete	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Ward	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
McQuitty	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
kmeans	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
PAM	0	24	26	0	0	0	0	0	0	0	0	0	0	0	0
KL															
Single	0	5	4	3	1	4	6	6	3	6	7	5	0	0	0
Complete	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
Ward	0	0	4	3	5	3	4	7	3	6	6	9	0	0	0
McQuitty	0	0	4	3	5	3	4	7	3	6	6	9	0	0	0
kmeans	0	16	6	2	5	2	3	6	4	1	2	3	0	0	0
PAM	0	0	7	4	6	4	4	5	7	2	6	5	0	0	0
Gap															
Single	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
Complete	0	0	4	13	19	13	1	0	0	0	0	0	0	0	0
Average	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0

Appendix C

OSil simulations results

This appendix has two part that consists of the results of the Simulation I and II conducted in Chapter 4. Appendix C.1 represents the histogram for the ASW value obtained from all the clustering methods included in Simulation I. Appendix C.2 represents 4 kind of results from Simulation II, which are described as the below.

- (i) It represents the box plots of the ASW values obtained for all the clustering methods included in the simulation.
- (ii) It represents the density plot of the ASW values obtained from OSil initialized by from the 9 initialization methods and density plot for the ASW values obtained from PAMSIL algorithm.
- (iii) It represents the numerical results in tables for the 10 DGPs used in the simulation
- (iv) It represents the frequency counts for the estimation of number of clusters for the 10 DGPs from all the clustering methods and cluster estimation methods included in the simulation.

C.1 Simulation I: Known k case

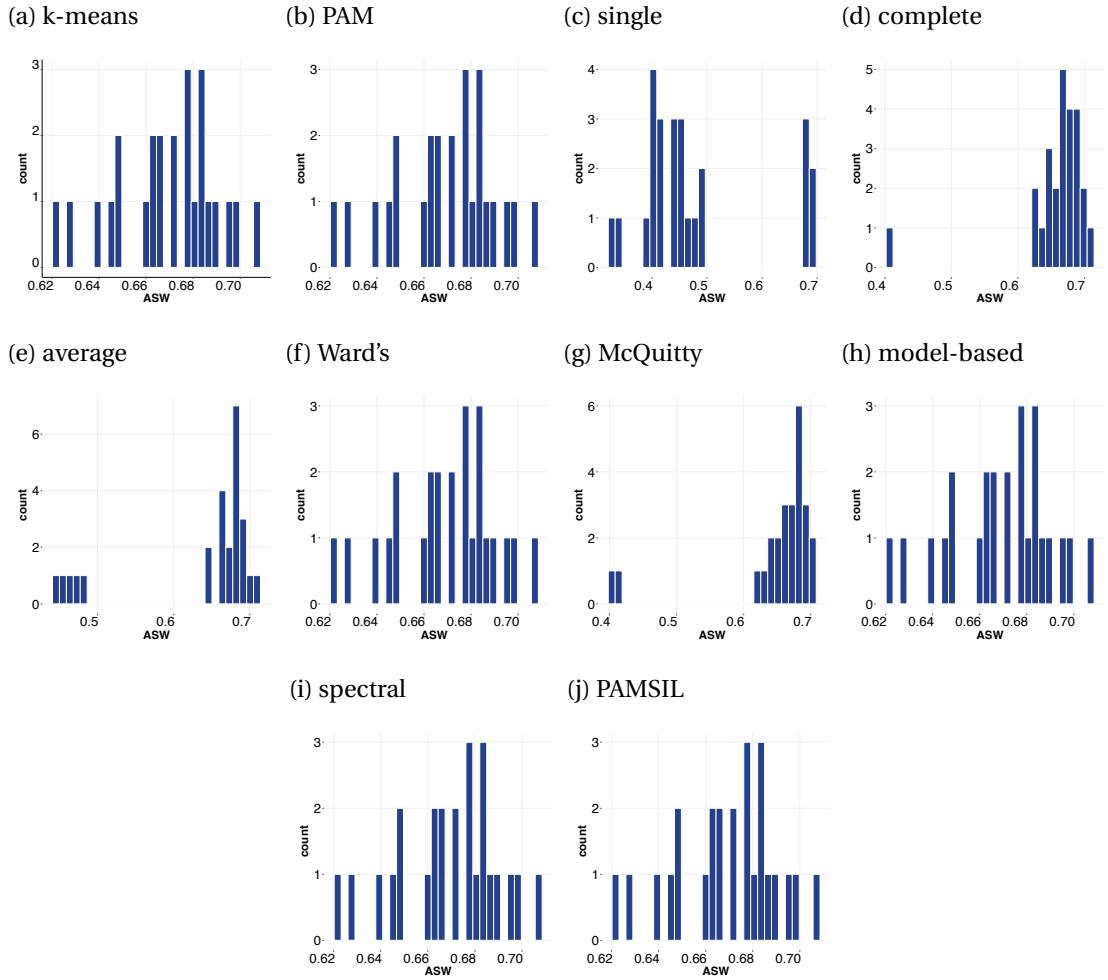


Figure C.1 Histogram for ASW obtained from $OSil$ initialized with clustering methods for Model 1.

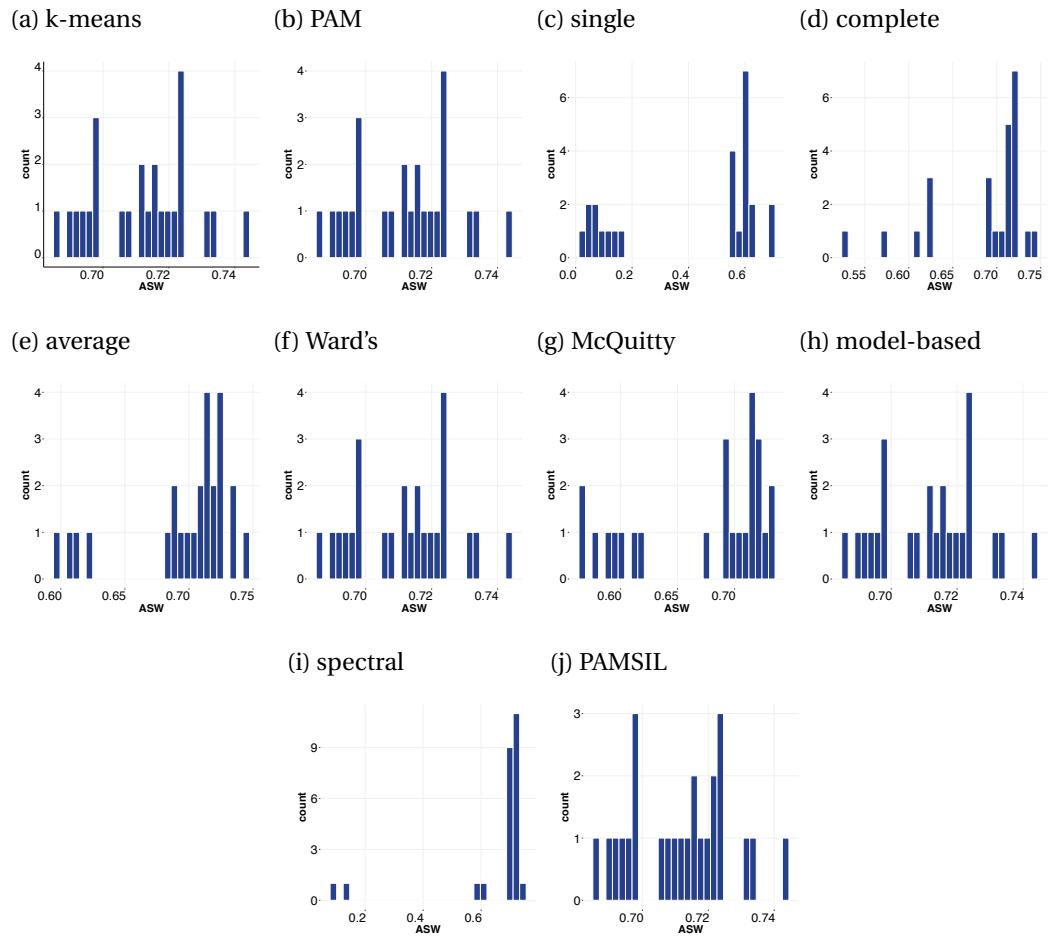


Figure C.2 Histogram for ASW obtained from $OSil$ initializing with clustering methods for Model 2.

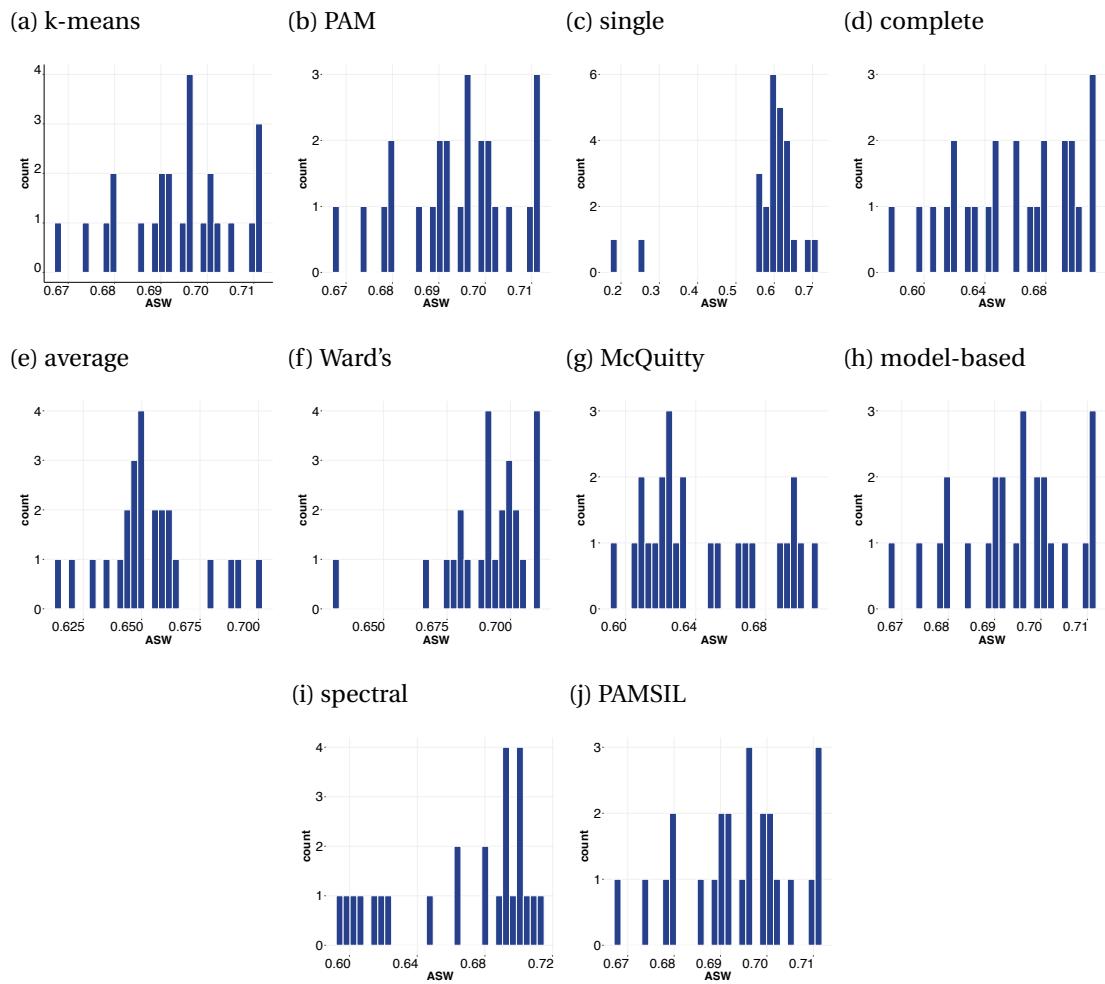


Figure C.3 Histogram for ASW obtained from $OSil$ initialized with clustering methods for Model 3.

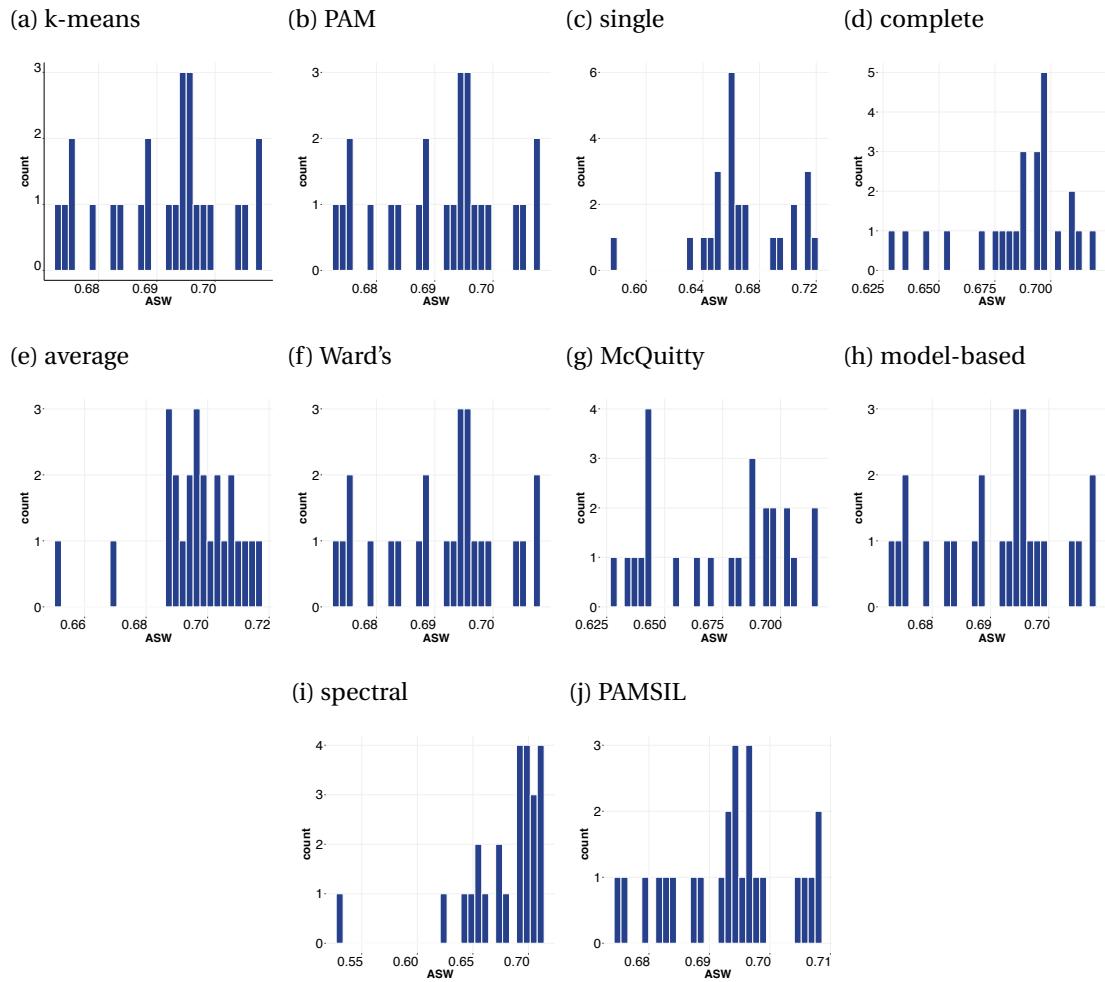


Figure C.4 Histogram for ASW obtained from $OSil$ initialized with clustering methods for Model 4.

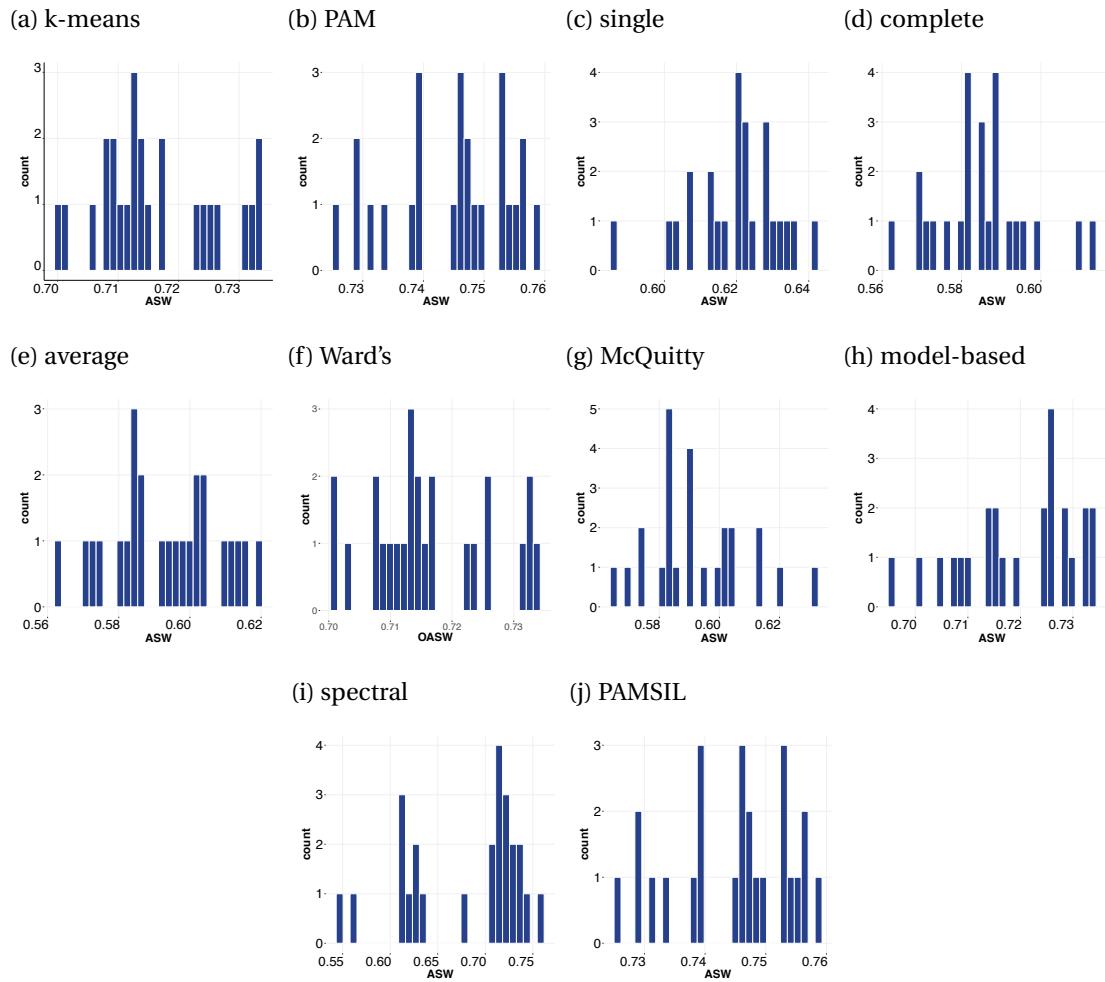


Figure C.5 Histogram for ASW obtained from $OSil$ initialized with clustering methods for Model 5.

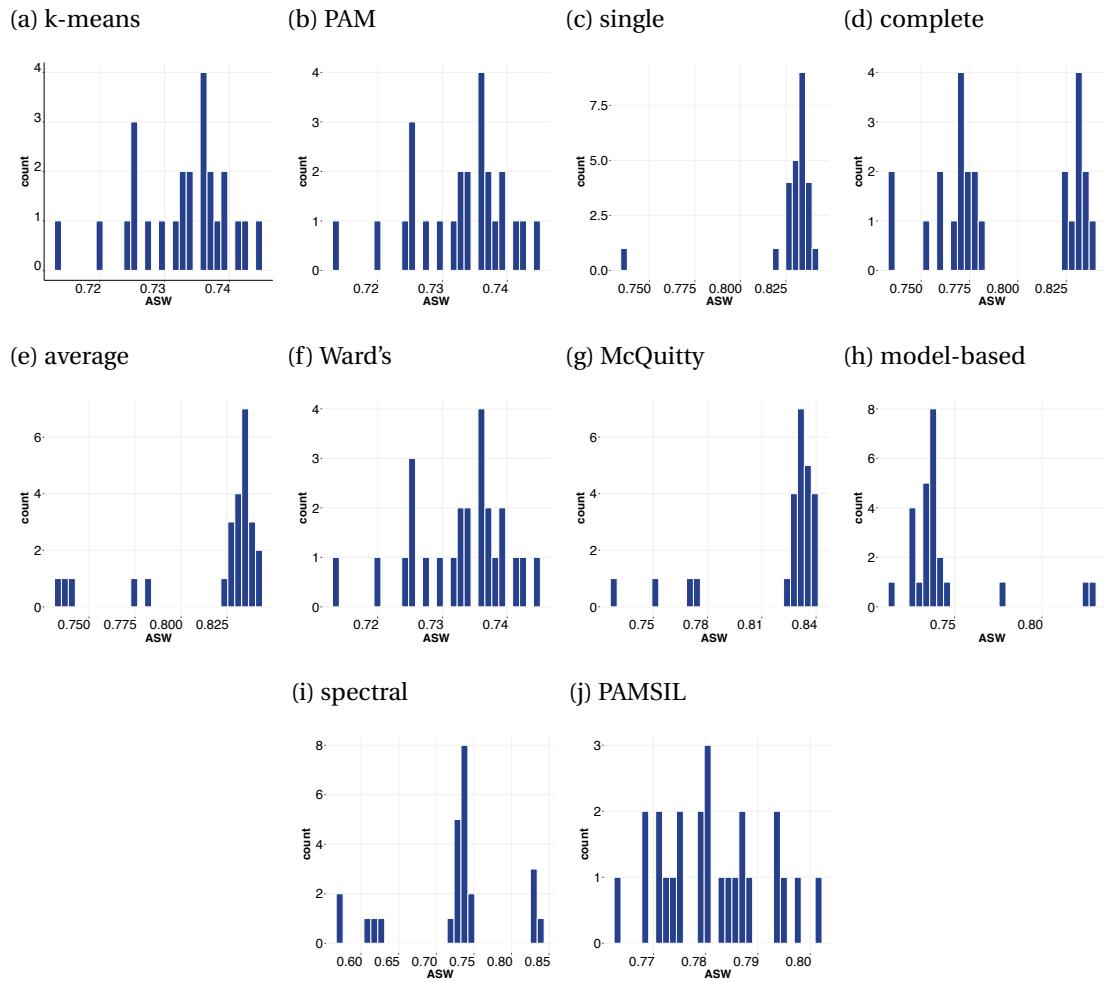


Figure C.6 Histogram for ASW obtained through $OSil$ initializing against clustering methods for Model 6.

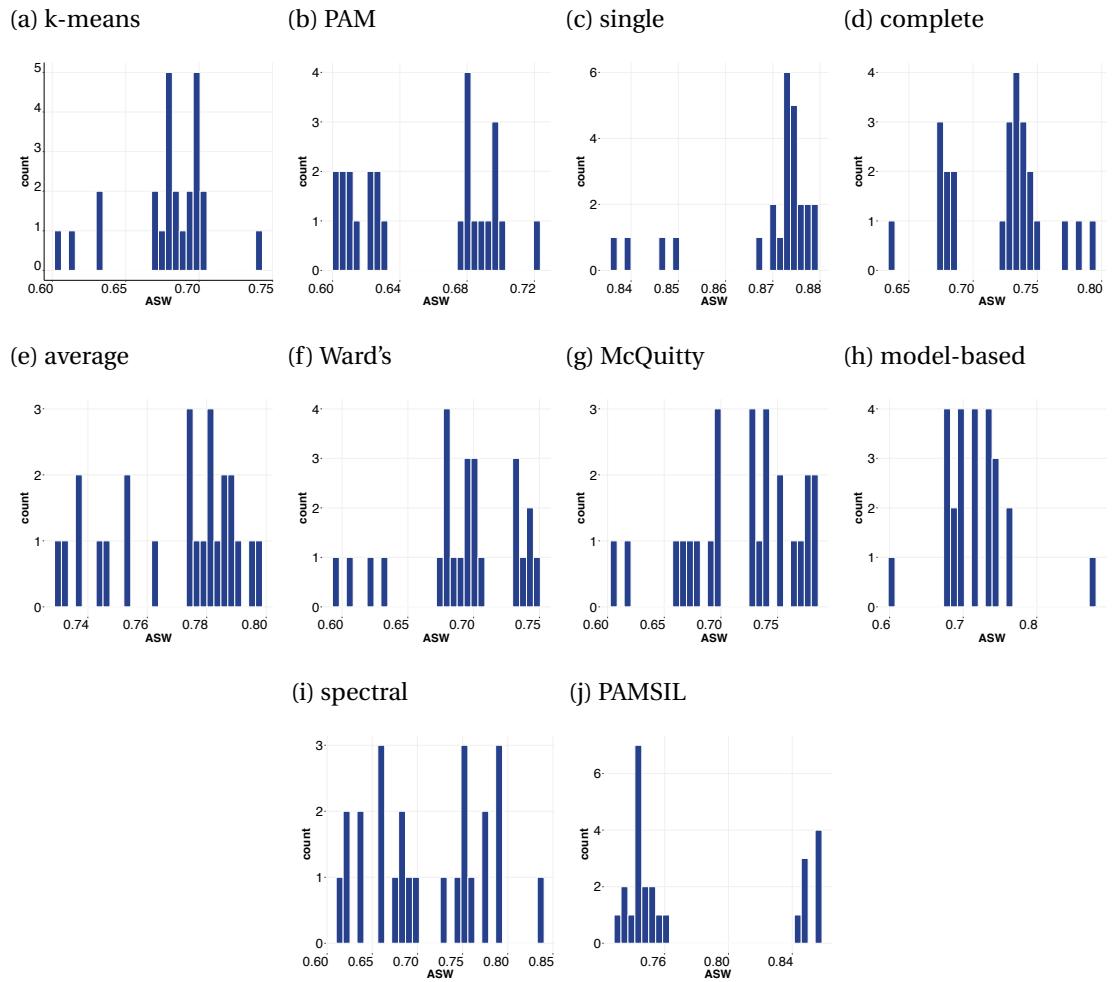


Figure C.7 Histogram for ASW obtained through $OSil$ initializing against clustering methods for Model 7.

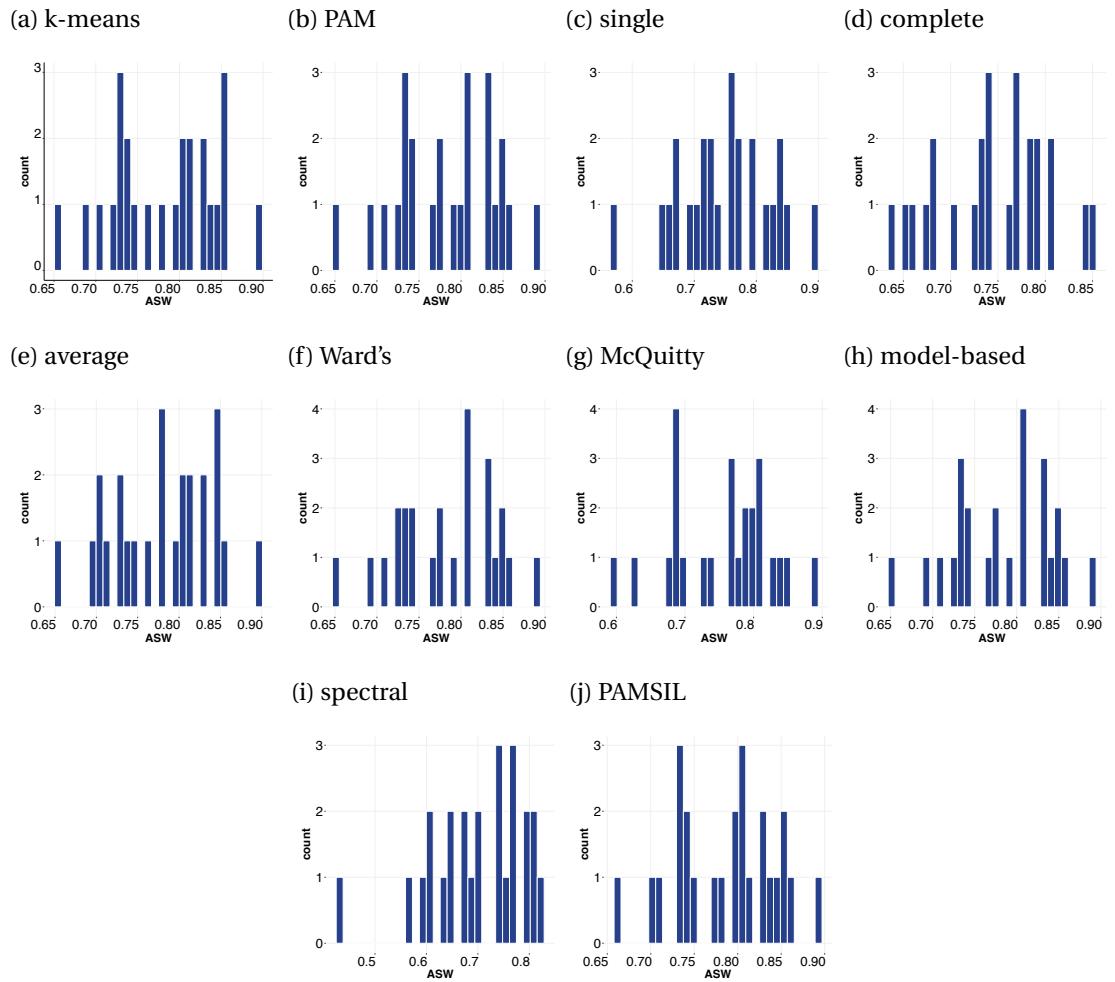


Figure C.8 Histogram for ASW obtained through $OSil$ initializing against clustering methods for Model 8.

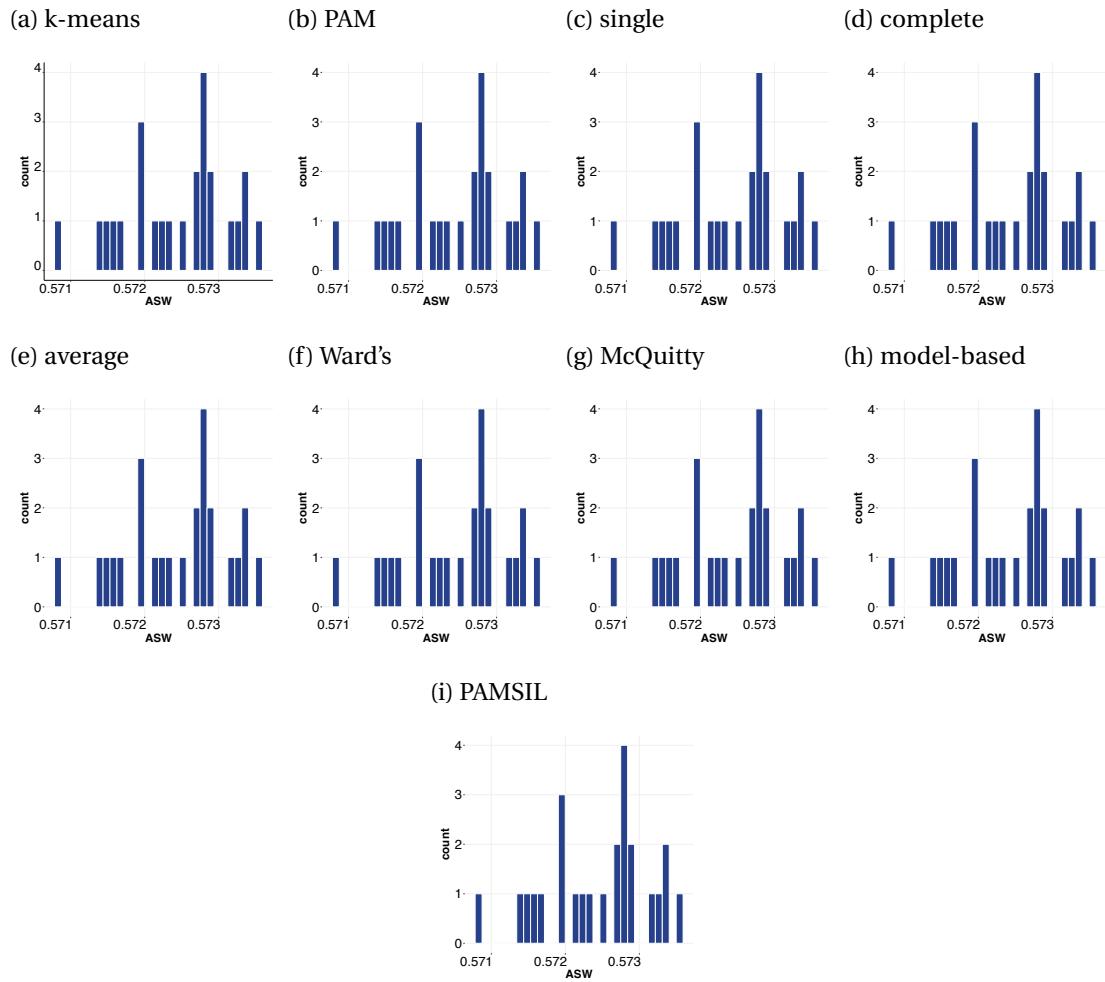


Figure C.9 Histogram for ASW obtained through $OSil$ initializing against clustering methods for Model 9.

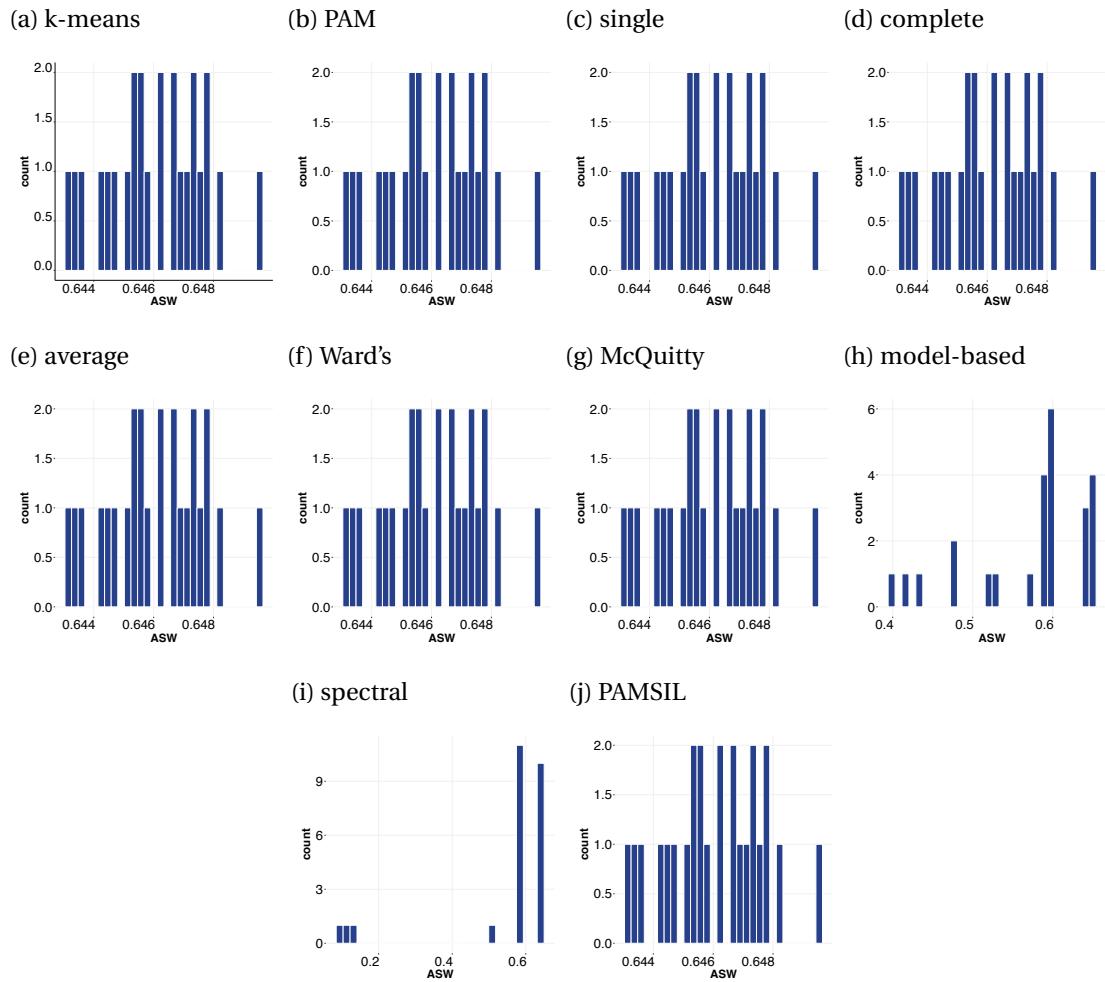


Figure C.10 Histogram for ASW obtained through *OSil* initializing against clustering methods for Model 9.

C.2 Simulation II: Estimation of k Case

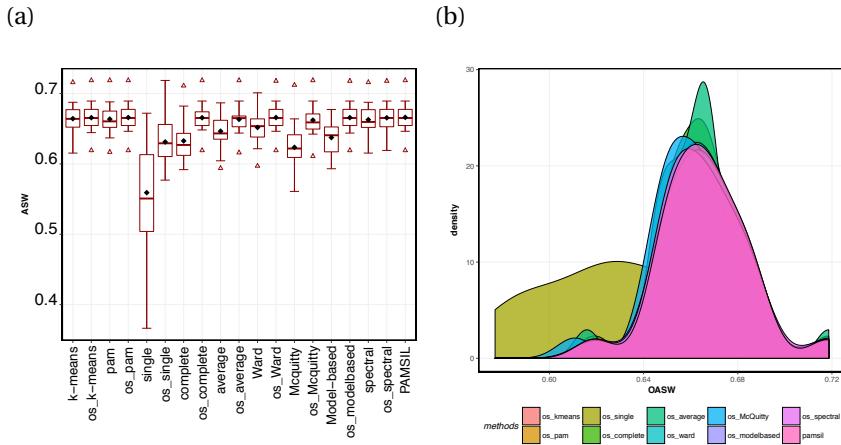


Figure C.11 (a) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods for Model 1. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles. (b) Density curve plots for ASW values obtained form OSil against each initialization methods for model 1.

Table C.1 Results for Model 1 for the estimation of k case.

Methods	ASW (init)	SE	ASW (OSil)	SE	ARI (init)	ARI (OSil)	iter
k-means	0.6643	0.0039	0.6657	0.0038	0.7393	0.7776	2
PAM	0.6636	0.0040	0.6659	0.0038	0.7497	0.7614	2
single	0.5591	0.0143	0.6312	0.0070	0.8270	0.7487	20
complete	0.6327	0.0061	0.6656	0.0038	0.6120	0.7622	13
average	0.6464	0.0042	0.6633	0.0038	0.7106	0.7604	6
Ward's	0.6518	0.0046	0.6663	0.0038	0.8009	0.7693	5
McQuitty	0.6235	0.0065	0.6622	0.0041	0.6201	0.7414	13
model-based	0.6374	0.0047	0.6658	0.0038	0.9177	0.7764	7
spectral	0.6629	0.0040	0.6655	0.0038	0.9355	0.7978	6
PAMSIL	-	-	0.6661	0.0038	-	0.7633	3

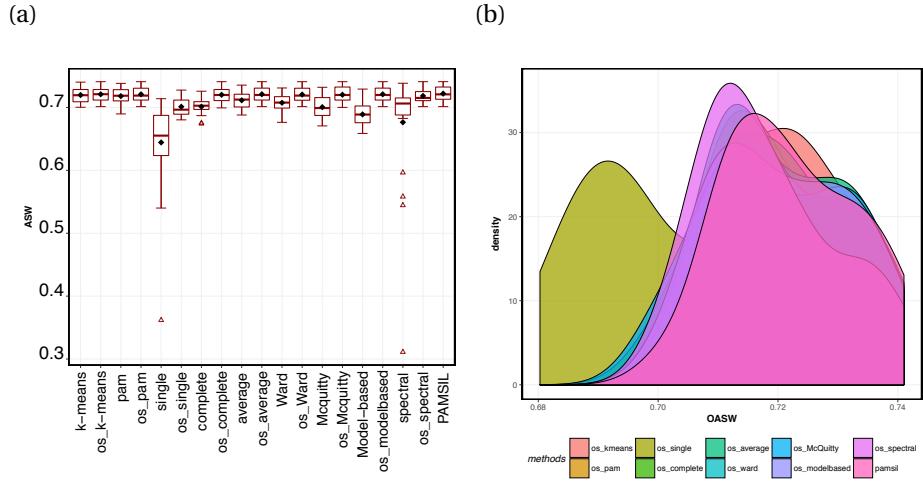


Figure C.12 (a) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods for Model 2. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles. (b) Density curve plots for ASW values obtained from OSil clustering against each initialization methods for Model 2.

Table C.2 Results for Model 2 for the estimation of k case

Methods	ASW (init)	SE	ASW (OSil)	SE	ARI (init)	ARI (OSil)	iter
k-means	0.7198	0.0023	0.7210	0.0021	0.8179	0.8232	2
PAM	0.7180	0.0025	0.7209	0.0022	0.8176	0.8141	4
single	0.6444	0.0147	0.7012	0.0026	0.8509	0.8361	20
complete	0.7013	0.0024	0.7202	0.0023	0.7703	0.8039	14
average	0.7114	0.0025	0.7210	0.0022	0.8300	0.8231	5
Ward's	0.7075	0.0031	0.7205	0.0022	0.8159	0.8156	7
McQuitty	0.7005	0.0031	0.7202	0.0024	0.7682	0.8121	15
model-based	0.6890	0.0037	0.7209	0.0023	0.9148	0.8178	12
spectral	0.6765	0.0182	0.7183	0.0022	0.8666	0.8118	13
PAMSIL	-	-	0.7220	0.0022	-	0.8088	3

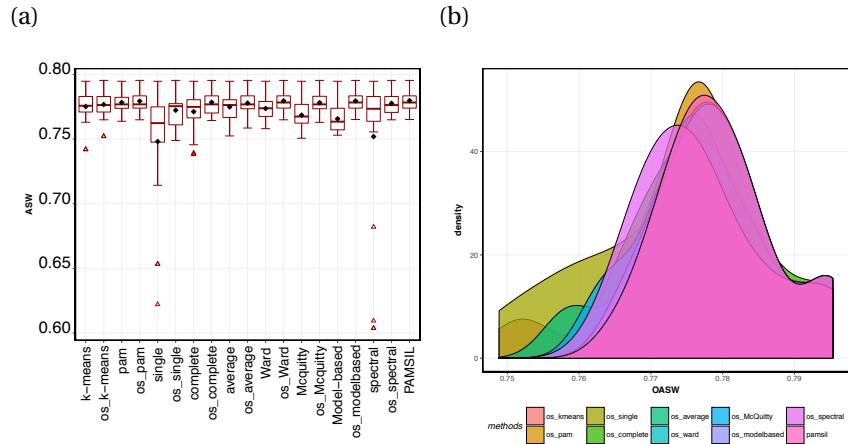


Figure C.13 (a) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods for Model 3. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles. (b) Density curve plots for ASW values obtained from OSil against each initialization methods for Model 3

Table C.3 Results for Model 3 for the estimation of k case.

Methods	ASW (init)	SE	ASW (OSil)	SE	ARI (init)	ARI (OSil)	iter
k-means	0.5895	0.0776	0.5905	0.0777	0.3816	0.4090	2
PAM	0.5912	0.0778	0.5920	0.0779	0.4301	0.4317	1
single	0.5715	0.0757	0.5873	0.0773	0.3127	0.2697	10
complete	0.5858	0.0771	0.5915	0.0778	0.3559	0.4031	5
average	0.5891	0.0776	0.5914	0.0778	0.4082	0.4282	2
Ward's	0.5880	0.0774	0.5922	0.0779	0.4400	0.4553	3
McQuitty	0.5837	0.0768	0.5916	0.0779	0.3872	0.4295	8
model-based	0.5820	0.0766	0.5922	0.0779	0.4014	0.4802	6
spectral	0.5729	0.0762	0.5910	0.0778	0.4392	0.3857	8
PAMSIL	-	-	0.5924	0.0780	-	0.4765	2

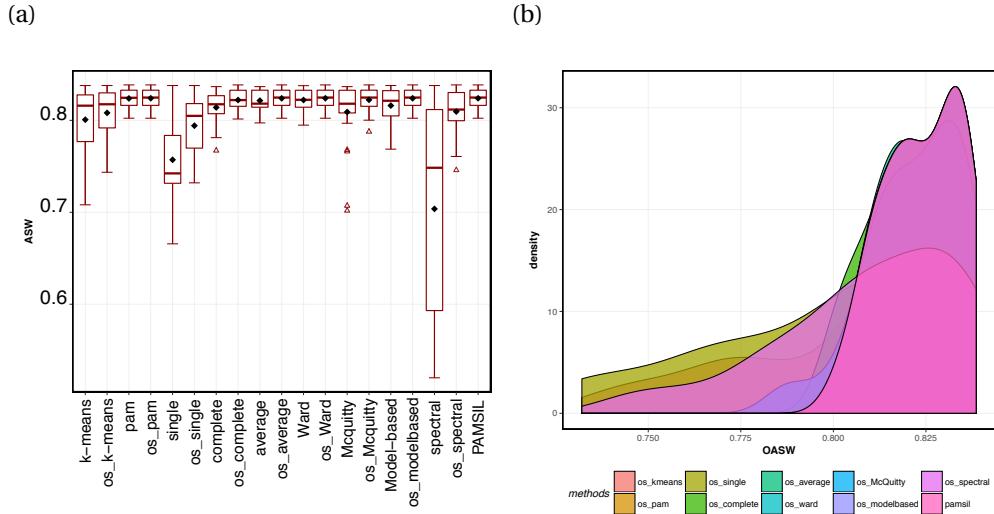


Figure C.14 (a) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods for Model 4. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles. (b) Density curve plots for ASW values obtained from OSil against each initialization methods for Model 4.

Table C.4 Results for Model 4 for the estimation of k case.

Methods	ASW (init)	SE	ASW (OSil)	SE	ARI (init)	ARI (OSil)	iter
k-means	0.8007	0.0076	0.8081	0.0057	0.9473	0.9623	3
PAM	0.8239	0.0021	0.8240	0.0021	0.9857	0.9853	1
single	0.7571	0.0083	0.7942	0.0061	0.8822	0.9393	7
complete	0.8141	0.0038	0.8221	0.0024	0.9681	0.9827	4
average	0.8215	0.0024	0.8240	0.0021	0.9814	0.9849	3
Ward's	0.8222	0.0023	0.8240	0.0021	0.9873	0.9857	2
McQuitty	0.8091	0.0073	0.8224	0.0026	0.9533	0.9810	5
model-based	0.8161	0.0038	0.8240	0.0021	0.9800	0.9853	3
spectral	0.7037	0.0228	0.8096	0.0050	0.9516	0.9674	6
PAMSIL	-	-	0.8240	0.0021	-	0.9853	2

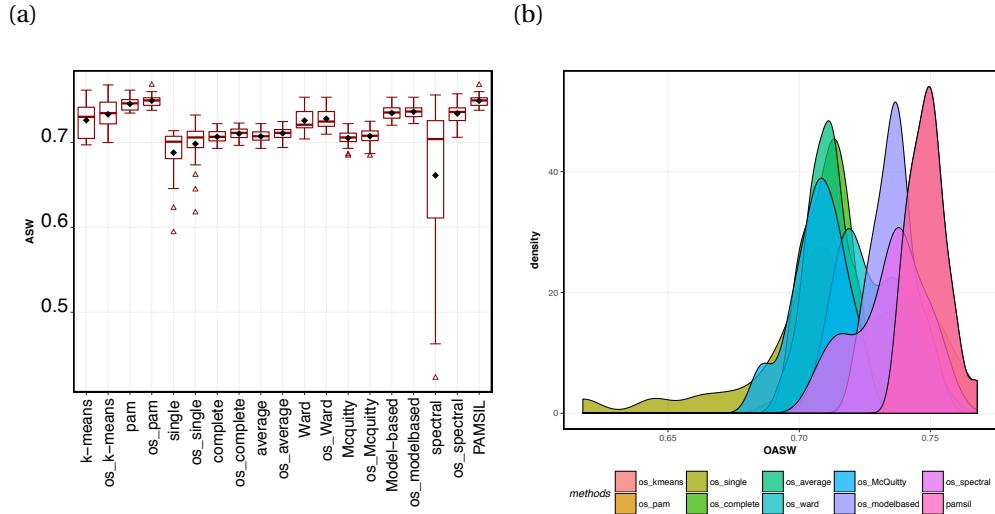


Figure C.15 (a) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods for Model 5. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles. (b) Density curve plots for ASW values obtained from OSil against each initialization methods for Model 5.

Table C.5 Results for Model 5 for the estimation of k case.

Methods	ASW (init)	SE	ASW (OSil)	SE	ARI (init)	ARI (OSil)	iter
k-means	0.7261	0.0040	0.7332	0.0037	0.7790	0.8657	6
PAM	0.7455	0.0015	0.7491	0.0014	0.9814	0.9968	3
single	0.6880	0.0060	0.6983	0.0051	0.2343	0.3344	3
complete	0.7067	0.0015	0.7104	0.0015	0.4280	0.5234	5
average	0.7072	0.0016	0.7108	0.0015	0.4810	0.5298	4
Ward's	0.7257	0.0028	0.7279	0.0025	0.7674	0.7904	3
McQuitty	0.7052	0.0019	0.7074	0.0021	0.3354	0.3313	3
model-based	0.7347	0.0018	0.7362	0.0016	0.8191	0.8163	2
spectral	0.6613	0.0172	0.7341	0.0028	0.6734	0.8722	10
PAMSIL	-	-	0.7491	0.0014	-	0.9968	3

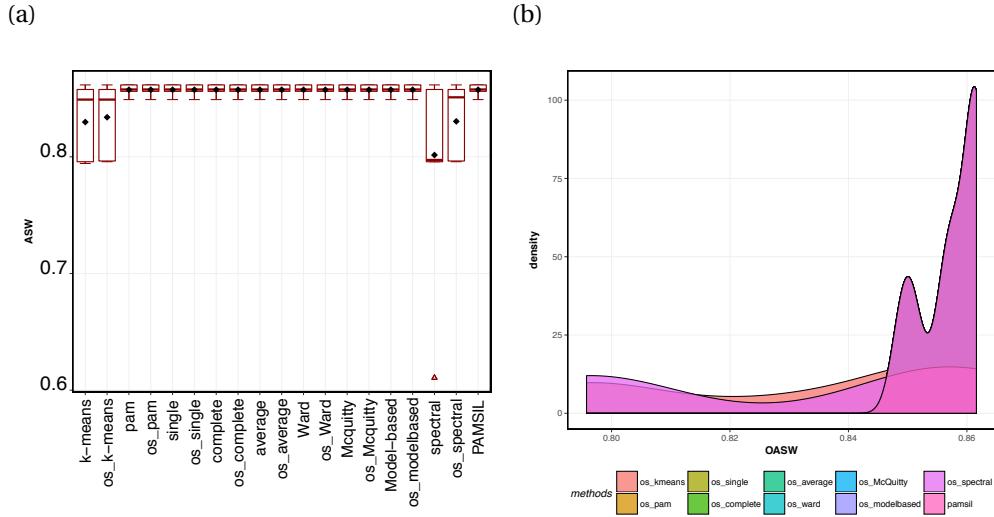


Figure C.16 (a) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods for Model 6. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles. (b) Density curve plots for ASW values obtained from OSil against each initialization methods for Model 6.

Table C.6 Results for Model 6 for the estimation of k case.

Methods	ASW (init)	SE	ASW (OSil)	SE	ARI (init)	ARI (OSil)	iter
k-means	0.8298	0.0062	0.8339	0.0056	0.6000	0.6478	3
PAM	0.8575	0.0010	0.8575	0.0010	0.7798	0.7798	1
single	0.8575	0.0010	0.8575	0.0010	0.7798	0.7798	1
complete	0.8575	0.0010	0.8575	0.0010	0.7798	0.7798	1
average	0.8575	0.0010	0.8575	0.0010	0.7798	0.7798	1
Ward's	0.8575	0.0010	0.8575	0.0010	0.7798	0.7798	1
McQuitty	0.8575	0.0010	0.8575	0.0010	0.7798	0.7798	1
model-based	0.8575	0.0010	0.8575	0.0010	0.7798	0.7798	1
spectral	0.8015	0.0156	0.8304	0.0062	0.6000	0.6000	1
PAMSIL	-	-	0.8575	0.0010	-	0.7798	2

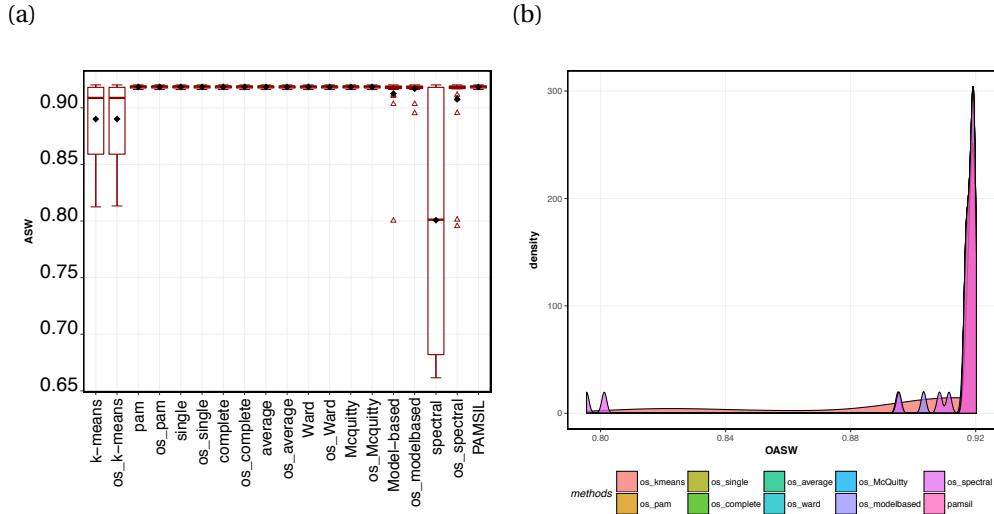


Figure C.17 (a) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods for Model 7. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles. (b) Density curve plots for ASW values obtained from OSil against each initialization methods for model 7.

Table C.7 Results for Model 7 for the estimation of k case.

Methods	ASW (init)	SE	ASW (OSil)	SE	ARI (init)	ARI (OSil)	iter
k-means	0.8901	0.00793	0.8901	0.0079	0.4752	0.4756	1
PAM	0.9185	0.0002	0.9185	0.0002	0.4125	0.4125	1
single	0.9185	0.000235	0.9185	0.0002	0.4125	0.4125	1
complete	0.9185	0.0002	0.9185	0.0002	0.4125	0.4125	1
average	0.9185	0.0002	0.9185	0.0002	0.4125	0.4125	1
Ward's	0.9185	0.0002	0.9185	0.0002	0.4125	0.4125	1
McQuitty	0.9185	0.0002	0.9185	0.0002	0.4125	0.4125	1
model-based	0.913	0.0047	0.9169	0.0011	0.4227	0.4229	6
spectral	0.8007	0.0207	0.9075	0.0066	0.4191	0.4168	5
PAMSIL	-	-	0.9185	0.0002	-	0.4125	2

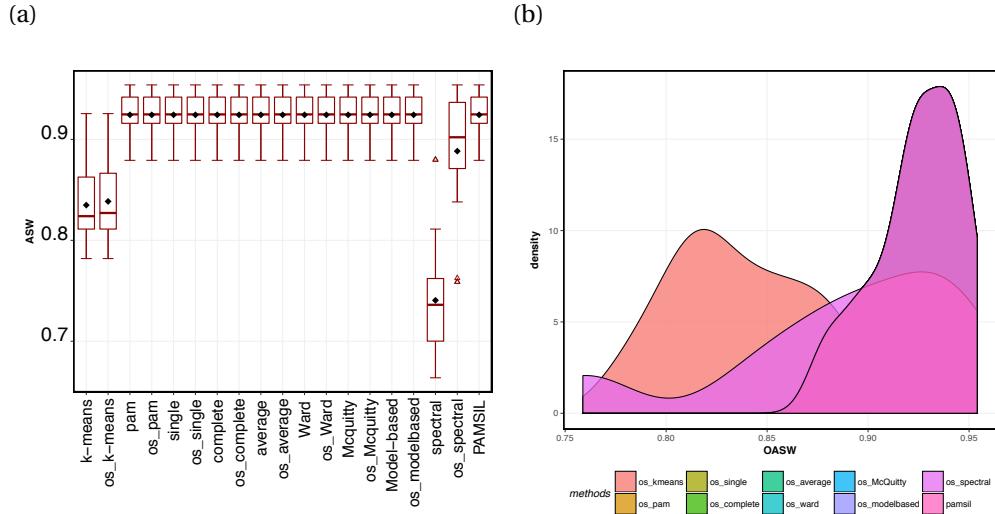


Figure C.18 (a) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods for Model 8. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles. (b) Density curve plots for ASW values obtained from OSil against each initialization methods for Model 8.

Table C.8 Results for Model 8 for the estimation of k case.

Methods	ASW (init)	SE	ASW (OSil)	SE	ARI (init)	ARI (OSil)	iter
k-means	0.8349	0.0071	0.8386	0.0073	0.8886	0.8899	2
PAM	0.9244	0.0043	0.9244	0.0043	1	1	1
single	0.9244	0.0043	0.9244	0.0043	1	1	1
complete	0.9244	0.0043	0.9244	0.0043	1	1	1
average	0.9244	0.0043	0.9244	0.0043	1	1	1
Ward's	0.9244	0.0043	0.9244	0.0043	1	1	1
McQuitty	0.9244	0.0043	0.9244	0.0043	1	1	1
model-based	0.9244	0.0043	0.9244	0.0043	1	1	1
spectral	0.7406	0.0115	0.8884	0.0118	0.9023	0.9399	11
PAMSIL	-	-	0.9244	0.0043	-	1	2

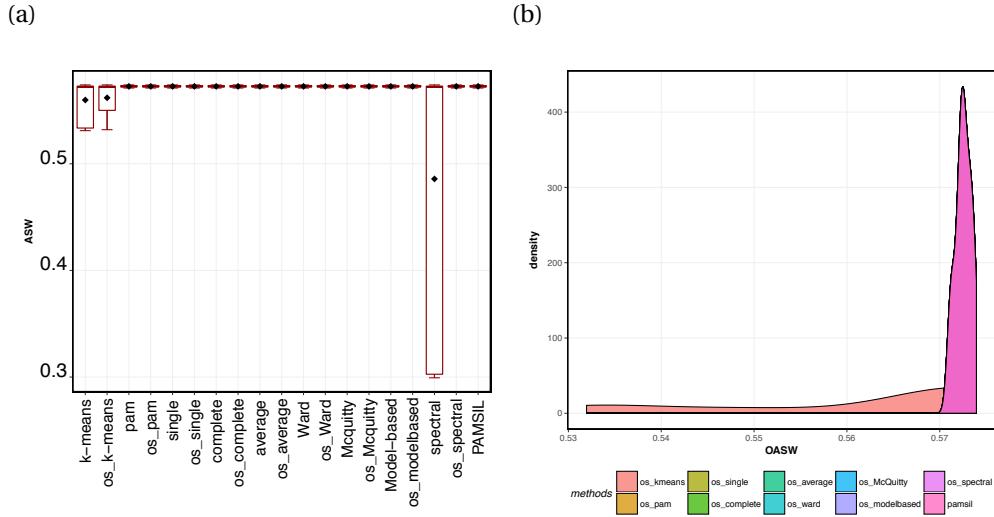


Figure C.19 (a) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods for Model 9. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles. (b) Density curve plots for ASW values obtained from OSil against each initialization methods for Model 9.

Table C.9 Results for Model 9 for the estimation of k case.

Methods	ASW (init)	SE	ASW (OSil)	SE	ARI (init)	ARI (OSil)	iter
k-means	0.5598	0.0038	0.5619	0.0033	0.8615	0.9104	3
PAM	0.5726	0.0002	0.5726	0.0002	1	1	1
single	0.5726	0.0002	0.5726	0.0002	1	1	1
complete	0.5726	0.0002	0.5726	0.0002	1	1	1
average	0.5726	0.0002	0.5726	0.0002	1	1	1
Ward's	0.5726	0.0002	0.5726	0.0002	1	1	1
McQuitty	0.5726	0.0002	0.5726	0.0002	1	1	1
model-based	0.5726	0.0002	0.5726	0.0002	1	1	1
spectral	0.0291	0.0022	0.5333	0.0002	0.0260	0.5673	48
PAMSIL	-	-	0.5726	0.0002	-	1	2

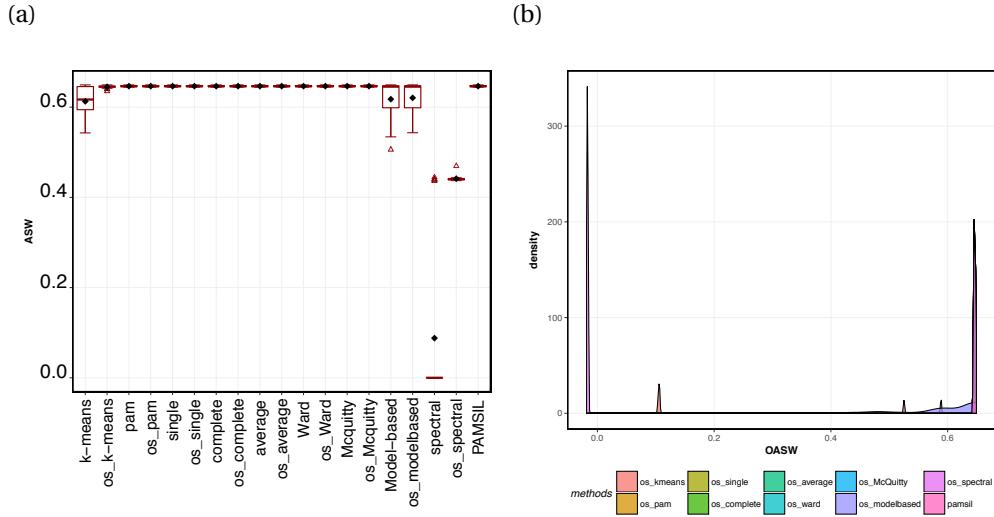


Figure C.20 (a) Boxplots for the average silhouette width values obtained from the clustering methods and OSil initialized with these methods for Model 10. The mean value for all the methods are plotted as black diamonds and the outliers are red triangles. (b) Density curve plots for ASW values obtained from OSil against each initialization methods for Model 10.

Table C.10 Results for Model 10 for the estimation of k case.

Methods	ASW (init)	SE	ASW (OSil)	SE	ARI (init)	ARI (OSil)	iter
k-means	0.6129	0.0076	0.6450	0.0006	0.9720	0.9998	95
PAM	0.6465	0.0003	0.6465	0.0003	1	1	1
single	0.6465	0.0003	0.6465	0.0003	1	1	1
complete	0.6465	0.0003	0.6465	0.0003	1	1	1
average	0.6465	0.0003	0.6465	0.0003	1	1	1
Ward's	0.6465	0.0003	0.6465	0.0003	1	1	1
McQuitty	0.6465	0.0003	0.6465	0.0003	1	1	1
spectral	0.0879	0.0359	0.4415	0.0012	0.0001	0.4270	230
model-based	0.6177	0.0081	0.6206	0.0068	0.9416	0.9484	4
PAMSIL	0	0	0.6465	0.0003	0	1	2

Table C.11: Results for estimation of number of clusters \hat{k} from indices and clustering methods included in the study for Model 1. The true number of clusters are made bold.

No. of clusters	1	2	3	4	5	6	7	8	9	10
CH										
kmeans	0	18	1	2	1	1	1	0	0	1
PAM	0	14	0	1	1	0	1	1	5	2
Single	0	3	7	3	5	1	1	2	2	1
Complete	0	3	0	1	1	0	4	4	4	8
Average	0	6	0	1	1	2	2	1	3	9
Ward	0	6	0	0	1	1	1	0	2	14
McQuitty	0	2	0	0	3	1	2	4	6	7
Model-based	0	6	1	2	2	1	2	2	3	6
Spectral	0	19	0	3	2	0	1	0	0	0
H										
kmeans	0	5	7	8	3	1	1	0	0	0
PAM	0	0	0	2	4	5	4	10	0	0
Single	5	0	0	1	0	0	0	0	0	0
Complete	0	4	0	1	0	4	1	15	0	0
Average	6	3	1	0	3	4	1	3	0	0
Ward	0	0	0	0	0	0	0	25	0	0
McQuitty	4	2	3	3	2	1	1	6	0	0
Model-based	1	8	6	3	6	1	0	0	0	0
Spectral	4	4	8	8	1	0	0	0	0	0
Gamma										
kmeans	0	1	3	6	8	5	2	0	0	0
PAM	0	0	0	0	2	4	4	5	5	5
Single	0	0	0	0	0	0	2	2	6	15
Complete	0	0	0	0	0	0	0	0	0	25
Average	0	0	0	0	0	0	0	0	0	25
Ward	0	0	0	0	0	0	0	1	0	24
McQuitty	0	0	0	0	0	0	0	0	0	25
Model-based	0	0	0	0	1	0	0	2	6	16
Spectral	0	0	0	2	6	4	4	6	2	1
C										
kmeans	0	1	1	6	8	5	2	0	0	2
PAM	0	0	0	0	2	4	4	5	6	4
Single	0	0	0	0	0	0	3	1	8	13
Complete	0	0	0	0	0	0	0	1	0	24
Average	0	0	0	0	0	0	0	1	0	24
Ward	0	0	0	0	0	0	0	0	1	24
McQuitty	0	0	0	0	0	0	0	0	0	25
Model-based	0	0	0	0	1	0	0	2	6	16
Spectral	0	0	0	2	6	4	4	5	2	2
KL										
kmeans	0	8	7	5	3	1	1	0	0	0
PAM	0	7	5	5	2	2	4	0	0	0
Single	0	7	7	0	2	6	3	0	0	0
Complete	0	3	7	9	2	3	1	0	0	0
Average	0	5	3	2	6	7	2	0	0	0
Ward	0	6	6	4	6	3	0	0	0	0
McQuitty	0	0	1	7	6	9	2	0	0	0
Model-based	0	2	4	6	6	4	3	0	0	0

Spectral	0	2	3	8	6	2	4	0	0	0
Gap										
kmeans	0	20	2	3	0	0	0	0	0	0
PAM	0	15	7	3	0	0	0	0	0	0
Single	22	3	0	0	0	0	0	0	0	0
Complete	7	14	4	0	0	0	0	0	0	0
Average	5	18	1	1	0	0	0	0	0	0
Ward	0	13	9	2	1	0	0	0	0	0
McQuitty	11	13	1	0	0	0	0	0	0	0
Jump										
$p/2$	0	6	0	9	10	0	0	0	0	0
$p/3$	1	22	0	1	1	0	0	0	0	0
$p/4$	23	2	0	0	0	0	0	0	0	0
$p/5$	25	0	0	0	0	0	0	0	0	0
$p/6$	25	0	0	0	0	0	0	0	0	0
$p/7$	25	0	0	0	0	0	0	0	0	0
PS										
kmeans	0	24	1	0	0	0	0	0	0	0
PAM	4	21	0	0	0	0	0	0	0	0
Single	6	12	7	0	0	0	0	0	0	0
Complete	25	0	0	0	0	0	0	0	0	0
Average	21	4	0	0	0	0	0	0	0	0
Ward	19	5	1	0	0	0	0	0	0	0
McQuitty	23	2	0	0	0	0	0	0	0	0
Model-based	0	25	0	0	0	0	0	0	0	0
Spectral	7	18	0	0	0	0	0	0	0	0
BI										
kmeans	0	25	0	0	0	0	0	0	0	0
PAM	0	6	11	3	2	0	1	0	1	10
Single	0	11	1	0	0	0	0	1	3	9
Complete	0	0	0	0	0	1	0	3	6	15
Average	0	0	0	0	1	0	0	1	7	16
Ward	0	0	3	0	0	3	6	7	3	3
McQuitty	0	0	0	0	0	0	0	2	2	21
Model-based	0	25	0	0	0	0	0	0	0	0
Spectral	0	5	12	2	1	0	0	0	0	5
CVNN										
kmeans	0	9	6	8	2	0	0	0	0	0
PAM	4	11	5	3	2	0	0	0	0	0
Single	0	11	8	3	2	1	0	0	0	0
Complete	0	12	7	3	2	1	0	0	0	0
Average	21	3	1	0	0	0	0	0	0	0
Ward	1	10	6	4	2	0	2	0	0	0
McQuitty	5	17	0	2	1	0	0	0	0	0
Model-based	8	8	1	5	3	0	0	0	0	0
Spectral	0	17	3	4	1	0	0	0	0	0
BIC										
Model-based	0	25	0	0	0	0	0	0	0	0
PAMSIL	0	20	1	0	2	1	0	0	1	0
ASW										
kmeans	0	21	1	2	1	0	0	0	0	0
PAM	0	20	1	1	1	0	0	1	1	0
Single	0	4	8	4	6	1	0	0	1	1
Complete	0	7	2	2	2	2	4	2	1	3

Average	0	13	1	2	2	2	1	2	1	1
Ward	0	19	0	0	1	2	1	0	1	1
McQuitty	0	7	3	0	4	3	1	4	1	2
Model-based	0	20	0	2	0	0	1	1	1	0
Spectral	0	24	0	1	0	0	0	0	0	0
						OSil				
kmeans	0	21	1	2	1	0	0	0	0	0
PAM	0	20	1	1	1	0	0	1	1	0
Single	0	7	7	3	3	1	3	1	0	0
Complete	0	19	2	0	2	1	0	0	1	0
Average	0	17	3	1	1	2	0	1	0	0
Ward	0	20	1	1	1	1	0	0	1	0
McQuitty	0	15	5	2	1	1	0	1	0	0
Model-based	0	21	0	0	3	0	0	1	0	0
Spectral	0	22	1	2	0	0	0	0	0	0

Table C.12: Results for estimation of number of clusters \hat{k} from indices and clustering methods included in the study for Model 2. The true number of clusters are made bold.

No. of clusters	1	2	3	4	5	6	7	8	9	10
CH										
kmeans	0	0	2	7	4	6	5	1	0	0
PAM	0	0	1	4	1	2	5	4	6	2
Single	0	3	6	5	4	2	1	3	1	0
Complete	0	0	0	0	0	4	5	2	3	11
Average	0	0	1	4	2	3	0	3	1	11
Ward	0	0	0	0	0	2	3	1	5	14
McQuitty	0	0	0	1	0	4	3	4	3	10
Model-based	0	0	4	2	3	6	4	1	1	4
Spectral	0	0	8	3	2	6	2	2	1	1
H										
kmeans	0	1	4	6	7	6	1	0	0	0
PAM	0	0	0	0	1	1	4	19	0	0
Single	6	2	0	2	1	0	0	0	0	0
Complete	0	0	0	0	0	0	0	25	0	0
Average	1	1	2	1	3	2	3	12	0	0
Ward	0	0	0	0	0	0	0	25	0	0
McQuitty	1	1	2	2	3	2	0	13	0	0
Model-based	0	1	5	6	5	4	3	1	0	0
Spectral	5	4	5	8	3	0	0	0	0	0
Gamma										
kmeans	0	0	0	3	5	10	5	2	0	0
PAM	0	0	0	0	0	0	4	7	8	6
Single	0	0	0	0	1	0	1	3	8	12
Complete	0	0	0	0	0	0	0	0	2	23
Average	0	0	0	0	0	0	0	0	0	25
Ward	0	0	0	0	0	0	0	0	0	25
McQuitty	0	0	0	0	0	0	0	1	1	23
Model-based	0	0	0	1	3	5	5	2	2	7
Spectral	0	0	1	0	3	10	4	3	2	2
C										

kmeans	0	0	0	4	4	10	5	2	0	0
PAM	0	0	0	0	0	0	4	7	8	6
Single	0	0	0	0	1	1	1	5	8	9
Complete	0	0	0	0	0	0	0	0	2	23
Average	0	0	0	0	0	0	1	0	0	24
Ward	0	0	0	0	0	0	0	0	0	25
McQuitty	0	0	0	0	0	0	0	1	0	24
Model-based	0	0	0	0	3	4	4	2	2	10
Spectral	0	1	0	0	4	10	5	3	1	1
KL										
kmeans	0	16	3	4	1	1	0	0	0	0
PAM	0	6	4	6	1	4	4	0	0	0
Single	0	6	5	3	7	3	1	0	0	0
Complete	0	2	7	5	10	0	1	0	0	0
Average	0	12	2	1	2	4	4	0	0	0
Ward	0	3	2	9	5	2	4	0	0	0
McQuitty	0	9	6	1	3	3	3	0	0	0
Model-based	0	3	6	5	4	5	2	0	0	0
Spectral	0	6	2	4	5	4	4	0	0	0
Gap										
kmeans	0	0	9	11	4	1	0	0	0	0
PAM	0	0	8	12	3	2	0	0	0	0
Single	14	7	3	1	0	0	0	0	0	0
Complete	0	9	5	6	3	2	0	0	0	0
Average	0	1	11	11	1	1	0	0	0	0
Ward	0	0	9	7	7	1	1	0	0	0
McQuitty	1	9	8	7	0	0	0	0	0	0
Jump										
$p/2$	0	0	8	8	9	0	0	0	0	0
$p/2$	0	0	24	1	0	0	0	0	0	0
$p/2$	0	4	20	1	0	0	0	0	0	0
$p/2$	24	0	1	0	0	0	0	0	0	0
$p/2$	25	0	0	0	0	0	0	0	0	0
$p/2$	25	0	0	0	0	0	0	0	0	0
PS										
kmeans	0	0	25	0	0	0	0	0	0	0
PAM	1	24	0	0	0	0	0	0	0	0
Single	13	10	2	0	0	0	0	0	0	0
Complete	25	0	0	0	0	0	0	0	0	0
Average	22	3	0	0	0	0	0	0	0	0
Ward	25	0	0	0	0	0	0	0	0	0
McQuitty	22	3	0	0	0	0	0	0	0	0
Model-based	0	0	25	0	0	0	0	0	0	0
Spectral	19	5	1	0	0	0	0	0	0	0
BI										
kmeans	0	3	22	0	0	0	0	0	0	0
PAM	0	16	0	5	4	0	0	0	0	0
Single	0	9	2	1	0	0	0	0	4	9
Complete	0	0	0	0	1	0	0	11	5	8
Average	0	0	0	0	0	0	0	2	6	17
Ward	0	0	0	0	1	16	7	1	0	0
McQuitty	0	0	0	0	0	0	0	4	11	10
Model-based	0	0	25	0	0	0	0	0	0	0
Spectral	0	1	1	8	7	6	0	2	0	0

	CVNN									
kmeans	0	0	21	3	1	0	0	0	0	0
PAM	0	1	21	3	0	0	0	0	0	0
Single	14	7	4	0	0	0	0	0	0	0
Complete	0	2	8	8	5	2	0	0	0	0
Average	0	1	20	4	0	0	0	0	0	0
Ward	0	0	21	4	0	0	0	0	0	0
McQuitty	1	3	8	8	4	1	0	0	0	0
Model-based	0	0	24	1	0	0	0	0	0	0
Spectral	1	13	3	3	1	3	1	0	0	0
	BIC									
Model-based	0	0	25	0	0	0	0	0	0	0
PAMSIL	0	0	7	12	0	1	2	1	1	1
	ASW									
kmeans	0	0	13	8	1	2	1	0	0	0
PAM	0	0	12	7	3	3	0	0	0	0
Single	0	3	5	5	4	1	2	3	2	0
Complete	0	1	1	3	1	12	3	2	2	0
Average	0	0	11	10	0	2	1	0	0	1
Ward	0	1	8	4	1	3	1	3	1	3
McQuitty	0	1	2	8	1	5	2	2	3	1
Model-based	0	0	13	3	2	5	1	0	0	1
Spectral	0	2	12	4	1	3	1	1	0	1
	OSil									
kmeans	0	0	12	9	1	2	1	0	0	0
PAM	0	0	9	9	2	2	3	0	0	0
Single	0	0	3	6	7	3	1	2	2	1
Complete	0	0	6	12	1	2	1	0	1	2
Average	0	0	9	13	0	2	1	0	0	0
Ward	0	0	10	8	1	1	2	0	1	2
McQuitty	0	0	7	14	1	1	1	1	0	0
Model-based	0	0	9	12	1	1	1	1	0	0
Spectral	0	0	7	9	2	1	3	2	1	0

Table C.13: Results for estimation of number of clusters \hat{k} from indices and clustering methods included in the study for Model 3. The true number of clusters are made bold.

No. of clusters	1	2	3	4	5	6	7	8	9	10
CH										
kmeans	0	0	0	0	9	5	0	0	5	6
PAM	0	0	0	0	3	10	4	1	4	2
Single	0	0	1	0	0	3	2	3	9	9
Complete	0	0	0	0	3	5	1	4	9	3
Average	0	0	0	0	1	5	2	6	4	7
Ward	0	0	0	0	1	3	3	3	1	14
McQuitty	0	0	0	0	1	2	1	9	4	8
Model-based	0	0	0	2	2	1	1	10	3	5
Spectral	0	0	0	1	9	7	8	0	0	0
H										
kmeans	3	2	6	10	3	1	0	0	0	0
PAM	0	0	0	0	0	1	3	21	0	0

	10	7	1	1	0	0	1	1	0	0
Single	0	0	2	2	0	0	0	21	0	0
Complete	4	4	2	1	2	2	1	9	0	0
Average	0	0	0	0	0	0	0	25	0	0
Ward	1	3	4	3	0	1	0	13	0	0
McQuitty	0	0	2	7	8	3	3	2	0	0
Model-based	4	8	1	6	5	1	0	0	0	0
	Gamma									
kmeans	0	10	1	2	9	3	0	0	0	0
PAM	0	3	0	0	4	6	9	3	0	0
Single	0	5	1	2	0	0	0	1	0	16
Complete	0	2	0	0	0	0	0	4	3	16
Average	0	2	0	0	0	0	2	3	2	16
Ward	0	2	0	0	0	2	5	5	2	9
McQuitty	0	2	0	0	0	0	0	2	3	18
Model-based	0	2	0	0	4	1	2	6	3	7
Spectral	0	4	0	0	8	4	8	1	0	0
	C									
kmeans	0	2	2	1	17	2	0	0	1	0
PAM	0	1	0	0	5	5	11	3	0	0
Single	0	0	0	1	0	0	1	2	2	19
Complete	0	0	0	0	0	0	0	0	3	22
Average	0	0	0	0	0	0	0	0	0	25
Ward	0	0	0	0	0	1	2	2	9	11
McQuitty	0	0	0	0	0	0	0	1	1	23
Model-based	0	1	0	0	5	1	1	7	2	8
Spectral	0	1	0	0	10	3	10	1	0	0
	KL									
kmeans	0	11	7	0	5	2	0	0	0	0
PAM	0	3	11	6	3	1	1	0	0	0
Single	0	1	4	5	7	2	6	0	0	0
Complete	0	1	1	9	5	9	0	0	0	0
Average	0	4	1	5	7	3	5	0	0	0
Ward	0	0	11	6	1	5	2	0	0	0
McQuitty	0	3	6	9	5	1	1	0	0	0
Model-based	0	12	5	1	1	3	3	0	0	0
Spectral	0	5	10	5	4	0	1	0	0	0
	Gap									
kmeans	0	2	10	4	8	1	0	0	0	0
PAM	0	0	0	0	4	15	4	2	0	0
Single	4	11	10	0	0	0	0	0	0	0
Complete	2	1	13	4	2	1	1	0	0	0
Average	0	4	19	1	1	0	0	0	0	0
Ward	0	0	0	0	5	7	9	3	1	0
McQuitty	4	9	7	5	0	0	0	0	0	0
	Jump									
p/2	0	0	0	1	24	0	0	0	0	0
p/3	0	0	0	3	22	0	0	0	0	0
p/4	0	0	0	10	15	0	0	0	0	0
p/5	0	2	2	15	6	0	0	0	0	0
p/6	20	0	0	2	3	0	0	0	0	0
p/7	25	0	0	0	0	0	0	0	0	0
	PS									
kmeans	0	22	2	1	0	0	0	0	0	0

PAM	0	0	24	1	0	0	0	0	0	0
Single	9	5	1	9	1	0	0	0	0	0
Complete	11	1	13	0	0	0	0	0	0	0
Average	1	9	12	3	0	0	0	0	0	0
Ward	0	0	24	1	0	0	0	0	0	0
McQuitty	13	0	10	2	0	0	0	0	0	0
Model-based	0	0	5	20	0	0	0	0	0	0
Spectral	0	21	4	0	0	0	0	0	0	0
BI										
kmeans	0	25	0	0	0	0	0	0	0	0
PAM	0	13	5	6	0	0	1	0	0	0
Single	0	8	2	3	7	4	1	0	0	0
Complete	0	1	7	0	2	2	5	3	3	1
Average	0	12	1	0	0	0	1	1	1	9
Ward	0	18	1	0	0	3	0	2	0	1
McQuitty	0	0	2	1	0	0	2	4	4	12
Model-based	0	13	10	2	0	0	0	0	0	0
Spectral	0	24	0	0	0	1	0	0	0	0
CVNN										
kmeans	0	0	11	11	1	2	0	0	0	0
PAM	0	0	3	20	2	0	0	0	0	0
Single	4	12	9	0	0	0	0	0	0	0
Complete	0	0	12	6	7	0	0	0	0	0
Average	0	4	14	2	5	0	0	0	0	0
Ward	0	0	0	21	4	0	0	0	0	0
McQuitty	0	2	11	8	4	0	0	0	0	0
Model-based	0	0	0	25	0	0	0	0	0	0
Spectral	0	2	11	4	6	1	1	0	0	0
BIC										
Model-based	0	0	0	15	1	0	0	0	0	0
PAMSIL	0	12	0	3	7	3	0	0	1	0
ASW										
kmeans	0	17	0	1	7	0	0	0	0	0
PAM	0	14	0	3	6	2	0	0	0	0
Single	0	19	1	0	0	1	0	0	0	4
Complete	0	19	0	0	5	0	0	0	1	0
Average	0	15	0	1	6	0	0	0	3	0
Ward	0	13	0	3	6	3	0	0	0	0
McQuitty	0	16	0	1	5	0	2	1	0	0
Model-based	0	15	0	5	4	0	0	1	0	0
Spectral	0	14	0	2	7	2	0	0	0	0
OSil										
kmeans	0	12	0	2	5	0	0	0	0	0
PAM	0	11	0	2	5	1	0	0	0	0
Single	0	17	1	0	0	0	1	0	0	0
Complete	0	12	0	0	4	1	1	0	1	0
Average	0	11	0	1	4	1	1	0	1	0
Ward	0	10	0	2	5	2	0	0	0	0
McQuitty	0	11	0	1	4	2	0	0	1	0
Model-based	0	9	0	3	4	2	0	0	1	0
Spectral	0	13	0	1	4	1	0	0	0	0

Table C.14: Results for estimation of number of clusters \hat{k} from indices and clustering methods included in the study for Model 4 . The true number of clusters are made bold.

No. of clusters	1	2	3	4	5	6	7	8	9	10
CH										
kmeans	0	0	0	0	4	5	3	4	2	7
PAM	0	0	0	0	0	0	4	1	7	13
Single	0	0	0	2	2	6	6	1	4	4
Complete	0	0	0	0	1	0	1	3	6	14
Average	0	0	0	0	11	1	4	2	2	5
Ward	0	0	0	0	0	0	1	0	4	20
McQuitty	0	0	0	0	3	0	4	3	5	10
Model-based	0	0	0	0	7	1	3	5	4	5
Spectral	0	0	0	0	9	7	5	2	2	0
H										
kmeans	0	1	8	11	2	3	0	0	0	0
PAM	0	0	0	0	0	0	0	25	0	0
Single	2	0	13	3	3	4	0	0	0	0
Complete	0	0	1	0	0	0	0	24	0	0
Average	0	1	0	1	3	4	4	12	0	0
Ward	0	0	0	0	0	0	0	25	0	0
McQuitty	0	1	1	0	1	1	1	20	0	0
Model-based	0	0	0	5	3	1	5	11	0	0
Spectral	4	10	4	4	1	2	0	0	0	0
Gamma										
kmeans	0	0	0	0	16	4	1	2	0	2
PAM	0	0	0	0	20	1	0	0	0	4
Single	0	0	0	0	1	3	8	3	3	7
Complete	0	0	0	0	14	5	2	1	0	3
Average	0	0	0	0	7	8	4	1	0	5
Ward	0	0	0	0	16	4	1	0	1	3
McQuitty	0	0	0	0	8	9	3	0	0	5
Model-based	0	0	0	0	13	1	2	2	3	4
Spectral	0	0	0	0	10	7	5	1	0	2
C										
kmeans	0	0	0	1	17	3	1	1	0	2
PAM	0	0	0	0	18	0	1	0	0	6
Single	0	0	0	1	2	3	8	2	2	7
Complete	0	0	0	0	10	4	1	0	0	10
Average	0	0	0	0	6	8	4	0	0	7
Ward	0	0	0	0	14	1	1	0	1	8
McQuitty	0	0	0	0	7	8	3	0	0	7
Model-based	0	0	0	0	14	1	2	2	3	3
Spectral	0	0	0	0	11	7	4	1	0	2
KL										
kmeans	0	0	15	8	2	0	0	0	0	0
PAM	0	0	18	0	4	1	2	0	0	0
Single	0	5	2	6	6	4	2	0	0	0
Complete	0	0	0	20	5	0	0	0	0	0
Average	0	0	0	23	1	1	0	0	0	0
Ward	0	0	24	0	1	0	0	0	0	0

McQuitty	0	0	20	4	0	0	1	0	0	0
Model-based	0	0	17	4	2	2	0	0	0	0
Spectral	0	0	9	9	6	0	1	0	0	0
Gap										
kmeans	2	2	3	7	10	1	0	0	0	0
PAM	6	0	0	0	10	2	6	0	1	0
Single	0	2	0	21	2	0	0	0	0	0
Complete	0	5	0	3	16	0	1	0	0	0
Average	0	0	1	0	22	2	0	0	0	0
Ward	0	0	0	0	21	2	2	0	0	0
McQuitty	1	3	3	0	14	2	2	0	0	0
Jump										
$p/2$	0	0	0	0	25	0	0	0	0	0
$p/3$	0	0	0	0	25	0	0	0	0	0
$p/4$	0	0	0	0	25	0	0	0	0	0
$p/5$	0	0	0	0	25	0	0	0	0	0
$p/6$	0	0	0	0	25	0	0	0	0	0
$p/7$	21	0	0	0	4	0	0	0	0	0
PS										
kmeans	5	19	0	0	1	0	0	0	0	0
PAM	0	0	0	0	25	0	0	0	0	0
Single	4	0	0	20	0	1	0	0	0	0
Complete	5	0	0	0	20	0	0	0	0	0
Average	0	0	0	0	22	3	0	0	0	0
Ward	0	0	0	0	25	0	0	0	0	0
McQuitty	3	0	0	0	21	1	0	0	0	0
Model-based	0	0	0	1	24	0	0	0	0	0
Spectral	25	0	0	0	0	0	0	0	0	0
BI										
kmeans	0	5	0	0	5	6	3	3	2	1
PAM	0	0	0	1	24	0	0	0	0	0
Single	0	4	3	13	1	1	1	2	0	0
Complete	0	0	0	0	19	0	2	0	0	4
Average	0	0	0	3	21	1	0	0	0	0
Ward	0	0	0	2	23	0	0	0	0	0
McQuitty	0	0	0	0	17	5	1	0	1	1
Model-based	0	0	0	8	17	0	0	0	0	0
Spectral	0	0	0	0	1	4	9	8	1	2
CVNN										
kmeans	0	0	0	5	17	3	0	0	0	0
PAM	0	0	0	0	25	0	0	0	0	0
Single	0	2	0	21	2	0	0	0	0	0
Complete	0	0	0	4	21	0	0	0	0	0
Average	0	0	1	1	23	0	0	0	0	0
Ward	0	0	0	2	23	0	0	0	0	0
McQuitty	0	0	1	3	20	1	0	0	0	0
Model-based	0	0	0	0	25	0	0	0	0	0
Spectral	0	0	2	14	7	2	0	0	0	0
BIC										
Model-based	0	0	0	0	11	8	3	3	0	0
PAMSIL	0	0	0	0	25	0	0	0	0	0
ASW										
kmeans	0	0	0	3	17	4	1	0	0	0
PAM	0	0	0	0	25	0	0	0	0	0

Single	0	0	0	11	2	6	3	2	0	1
Complete	0	0	0	0	20	5	0	0	0	0
Average	0	0	0	0	24	1	0	0	0	0
Ward	0	0	0	0	25	0	0	0	0	0
McQuitty	0	0	0	2	20	3	0	0	0	0
Model-based	0	0	0	0	24	1	0	0	0	0
Spectral	0	0	0	1	14	7	3	0	0	0
OSil										
kmeans	0	0	0	1	17	4	2	1	0	0
PAM	0	0	0	0	25	0	0	0	0	0
Single	0	0	0	2	2	6	6	4	2	3
Complete	0	0	0	0	21	4	0	0	0	0
Average	0	0	0	0	24	1	0	0	0	0
Ward	0	0	0	0	25	0	0	0	0	0
McQuitty	0	0	0	0	22	3	0	0	0	0
Model-based	0	0	0	0	25	0	0	0	0	0
Spectral	0	0	0	0	14	8	3	0	0	0

Table C.15: Results for estimation of number of clusters \hat{k} from indices and clustering methods included in the study for Model 5. The true number of clusters are made bold.

No. of clusters	1	2	3	4	5	6	7	8	9	10
CH										
kmeans	0	0	0	0	1	4	4	7	4	5
PAM	0	0	0	0	1	0	0	2	3	19
Single	0	3	8	3	4	3	1	1	0	2
Complete	0	0	0	2	0	0	0	3	9	11
Average	0	0	2	0	0	0	0	4	10	9
Ward	0	0	0	0	0	2	4	3	2	14
McQuitty	0	0	2	6	0	0	2	1	9	5
Model-based	0	0	0	0	1	7	2	4	4	7
Spectral	0	0	0	1	3	8	5	3	3	2
H										
kmeans	0	4	7	5	4	2	3	0	0	0
PAM	0	0	0	2	0	0	0	23	0	0
Single	3	6	5	3	1	2	2	0	0	0
Complete	0	0	1	7	6	2	1	8	0	0
Average	4	0	9	5	2	2	0	3	0	0
Ward	0	0	0	0	0	0	0	25	0	0
McQuitty	1	0	2	7	6	3	2	4	0	0
Model-based	0	0	0	0	5	1	4	15	0	0
Spectral	6	4	10	1	0	3	1	0	0	0
Gamma										
kmeans	0	5	3	0	0	4	3	8	2	0
PAM	0	7	14	0	0	0	0	0	0	4
Single	0	4	4	5	1	2	3	0	1	5
Complete	0	0	0	2	1	0	0	0	0	22
Average	0	0	0	1	3	1	0	0	0	20
Ward	0	0	0	0	0	0	0	3	3	19
McQuitty	0	4	0	7	3	0	0	0	0	11
Model-based	0	0	0	0	0	0	0	2	6	17
Spectral	0	3	1	0	0	4	4	5	4	4

	C									
kmeans	0	0	3	1	3	5	2	6	3	2
PAM	0	0	8	0	0	0	0	0	0	17
Single	0	0	3	9	2	2	3	0	1	5
Complete	0	0	0	1	2	1	0	0	0	21
Average	0	0	0	0	1	3	1	0	0	20
Ward	0	0	0	0	0	0	0	4	8	13
McQuitty	0	0	0	3	7	3	0	0	0	12
Model-based	0	0	0	0	0	0	2	3	6	14
Spectral	0	0	3	1	0	5	4	4	4	4
	KL									
kmeans	0	5	4	4	5	5	2	0	0	0
PAM	0	0	0	0	16	7	2	0	0	0
Single	0	2	1	8	7	6	1	0	0	0
Complete	0	3	16	1	0	0	5	0	0	0
Average	0	2	5	2	2	7	7	0	0	0
Ward	0	0	0	10	8	5	2	0	0	0
McQuitty	0	8	2	1	4	2	8	0	0	0
Model-based	0	23	0	0	0	1	1	0	0	0
Spectral	0	7	2	4	3	5	4	0	0	0
	Gap									
kmeans	4	2	10	5	3	1	0	0	0	0
PAM	0	0	0	3	22	0	0	0	0	0
Single	6	7	8	3	1	0	0	0	0	0
Complete	0	15	10	0	0	0	0	0	0	0
Average	0	22	3	0	0	0	0	0	0	0
Ward	0	0	0	2	0	11	11	1	0	0
McQuitty	0	19	6	0	0	0	0	0	0	0
	Jump									
$p/2$	0	0	1	3	21	0	0	0	0	0
$p/3$	0	0	22	1	2	0	0	0	0	0
$p/4$	25	0	0	0	0	0	0	0	0	0
$p/5$	25	0	0	0	0	0	0	0	0	0
$p/6$	25	0	0	0	0	0	0	0	0	0
$p/7$	25	0	0	0	0	0	0	0	0	0
	PS									
kmeans	0	21	3	1	0	0	0	0	0	0
PAM	0	0	0	0	0	25	0	0	0	0
Single	7	2	8	6	2	0	0	0	0	0
Complete	0	1	24	0	0	0	0	0	0	0
Average	0	1	20	4	0	0	0	0	0	0
Ward	0	0	0	23	2	0	0	0	0	0
McQuitty	0	4	20	1	0	0	0	0	0	0
Model-based	0	0	0	0	25	0	0	0	0	0
Spectral	1	14	10	0	0	0	0	0	0	0
	BI									
kmeans	0	0	0	0	2	1	4	6	7	5
PAM	0	3	0	7	6	6	3	0	0	0
Single	0	7	3	2	1	3	2	4	0	3
Complete	0	23	0	0	0	0	1	1	0	0
Average	0	24	0	0	0	0	0	1	0	0
Ward	0	22	0	3	0	0	0	0	0	0
McQuitty	0	20	1	0	1	0	0	0	2	1

Model-based	0	0	1	5	19	0	0	0	0	0
Spectral	0	18	0	1	0	0	0	3	2	1
CVNN										
kmeans	0	0	2	10	10	3	0	0	0	0
PAM	0	0	3	9	13	0	0	0	0	0
Single	8	6	8	3	0	0	0	0	0	0
Complete	0	14	3	7	0	1	0	0	0	0
Average	0	13	9	2	1	0	0	0	0	0
Ward	0	0	0	12	8	5	0	0	0	0
McQuitty	0	8	6	11	0	0	0	0	0	0
Model-based	0	0	0	0	25	0	0	0	0	0
Spectral	0	0	8	9	4	3	1	0	0	0
BIC										
Model-based	0	0	0	0	24	1	0	0	0	0
PAMSIL	0	0	0	0	0	25	0	0	0	0
ASW										
kmeans	0	4	0	0	5	9	2	3	2	0
PAM	0	0	0	0	0	24	0	0	1	0
Single	0	19	2	0	0	0	0	1	1	2
Complete	0	13	0	0	0	0	0	0	5	7
Average	0	11	0	0	0	0	0	2	7	5
Ward	0	1	0	0	10	11	3	0	0	0
McQuitty	0	17	0	0	0	0	0	1	5	2
Model-based	0	0	0	0	25	0	0	0	0	0
Spectral	0	6	0	0	5	10	3	0	1	0
OSil										
kmeans	0	2	0	0	3	9	4	4	2	1
PAM	0	0	0	0	0	25	0	0	0	0
Single	0	15	2	0	0	0	2	3	1	2
Complete	0	9	0	0	0	0	0	1	8	7
Average	0	9	0	0	0	0	0	4	6	6
Ward	0	0	0	0	10	12	3	0	0	0
McQuitty	0	17	0	0	0	0	0	1	6	1
Model-based	0	0	0	0	25	0	0	0	0	0
Spectral	0	1	0	0	5	6	8	3	2	0

Table C.16: Results for estimation of number of clusters \hat{k} from indices and clustering methods included in the study for Model 6. The true number of clusters are made bold.

No. of clusters	1	2	3	4	5	6	7	8
CH								
kmeans	0	0	0	8	7	0	10	0
PAM	0	0	0	8	25	0	0	0
Single	0	0	0	25	0	0	0	0
Complete	0	0	0	14	0	11	0	0
Average	0	0	0	14	3	4	4	0
Ward	0	0	0	0	25	0	0	0
McQuitty	0	0	0	10	0	12	3	0
Model-based	0	0	0	0	25	0	0	0
Spectral	0	0	0	6	7	12	0	0
H								

kmeans	4	7	7	7	0	0	0	0
PAM	0	0	0	0	0	0	11	14
Single	0	0	21	4	0	0	0	0
Complete	0	0	0	0	3	22	0	0
Average	0	0	7	4	7	7	0	0
Ward	0	0	0	0	0	0	0	25
McQuitty	0	0	7	0	4	0	7	7
Model-based	0	0	0	0	3	4	14	4
Spectral	6	8	7	0	4	0	0	0
Gamma								
kmeans	0	25	0	0	0	0	0	0
PAM	0	25	0	0	0	0	0	0
Single	0	25	0	0	0	0	0	0
Complete	0	25	0	0	0	0	0	0
Average	0	25	0	0	0	0	0	0
Ward	0	25	0	0	0	0	0	0
McQuitty	0	25	0	0	0	0	0	0
Model-based	0	25	0	0	0	0	0	0
Spectral	0	25	0	0	0	0	0	0
C								
kmeans	0	25	0	0	0	0	0	0
PAM	0	25	0	0	0	0	0	0
Single	0	25	0	0	0	0	0	0
Complete	0	25	0	0	0	0	0	0
Average	0	25	0	0	0	0	0	0
Ward	0	25	0	0	0	0	0	0
McQuitty	0	25	0	0	0	0	0	0
Model-based	0	25	0	0	0	0	0	0
Spectral	0	25	0	0	0	0	0	0
KL								
kmeans	0	4	4	4	9	0	4	0
PAM	0	7	4	11	0	3	0	0
Single	0	21	0	4	0	0	0	0
Complete	0	0	13	0	0	0	0	0
Average	0	0	25	0	0	0	0	0
Ward	0	17	0	8	0	0	0	0
McQuitty	0	16	0	3	0	0	0	0
Model-based	0	0	15	7	0	3	0	0
Spectral	0	7	8	4	0	6	0	0
Gap								
kmeans	0	4	11	7	3	0	0	0
PAM	0	0	0	0	4	21	0	0
Single	0	0	0	21	4	0	0	0
Complete	0	0	0	14	3	8	0	0
Average	0	0	0	22	0	3	0	0
Ward	0	0	0	0	0	15	7	0
McQuitty	0	0	0	25	0	0	0	0
Jump								
$p/2$	0	0	0	0	25	0	0	0
$p/3$	0	0	0	0	25	0	0	0
$p/4$	0	0	0	17	8	0	0	0
$p/5$	0	0	0	25	0	0	0	0
$p/6$	0	0	0	25	0	0	0	0
$p/7$	0	0	0	25	0	0	0	0

PS							
kmeans	0	25	0	0	0	0	0
PAM	0	0	0	0	25	0	0
Single	0	0	0	11	14	0	0
Complete	0	0	0	25	0	0	0
Average	0	0	0	15	10	0	0
Ward	0	0	0	0	25	0	0
McQuitty	0	0	0	18	7	0	0
Model-based	0	0	0	0	25	0	0
Spectral	0	14	7	4	0	0	0
BI							
kmeans	0	25	0	0	0	0	0
PAM	0	25	0	0	0	0	0
Single	0	25	0	0	0	0	0
Complete	0	25	0	0	0	0	0
Average	0	25	0	0	0	0	0
Ward	0	25	0	0	0	0	0
McQuitty	0	25	0	0	0	0	0
Model-based	0	25	0	0	0	0	0
Spectral	0	25	0	0	0	0	0
CVNN							
kmeans	0	0	11	11	3	0	0
PAM	0	0	0	17	8	0	0
Single	0	0	0	25	0	0	0
Complete	0	0	0	25	0	0	0
Average	0	0	0	22	3	0	0
Ward	0	0	0	10	15	0	0
McQuitty	0	0	0	25	0	0	0
Model-based	0	0	0	10	15	0	0
Spectral	0	0	14	11	0	0	0
BIC							
Model-based	0	0	0	0	25	0	0
PAMSIL	0	0	0	25	0	0	0
ASW							
kmeans	0	11	0	14	0	0	0
PAM	0	0	0	25	0	0	0
Single	0	0	0	25	0	0	0
Complete	0	0	0	25	0	0	0
Average	0	0	0	25	0	0	0
Ward	0	0	0	25	0	0	0
McQuitty	0	0	0	25	0	0	0
Model-based	0	0	0	25	0	0	0
Spectral	0	11	0	14	0	0	0
OSil							
kmeans	0	8	0	14	3	0	0
PAM	0	0	0	25	0	0	0
Single	0	0	0	25	0	0	0
Complete	0	0	0	25	0	0	0
Average	0	0	0	25	0	0	0
Ward	0	0	0	25	0	0	0
McQuitty	0	0	0	25	0	0	0
Model-based	0	0	0	25	0	0	0
Spectral	0	11	0	14	0	0	0

Table C.17: Frequency count for estimation of number of clusters \hat{k} from indices and clustering methods included in the study for Model 7. The true number of clusters are made bold.

No. of clusters	1	2	3	4	5	6	7	8
CH								
kmeans	0	0	0	0	8	3	5	3
PAM	0	0	0	0	0	0	1	10
Single	0	0	0	18	1	1	1	1
Complete	0	0	0	0	13	0	1	5
Average	0	0	0	0	21	0	0	1
Ward	0	0	0	0	6	1	1	2
McQuitty	0	0	0	0	17	0	1	3
Model-based	0	0	0	0	0	10	10	2
Spectral	0	0	0	0	3	7	3	3
H								
kmeans	8	2	3	3	2	3	1	1
PAM	0	0	0	0	0	0	0	23
Single	0	0	17	6	0	0	0	0
Complete	0	0	0	0	0	0	0	23
Average	0	0	0	2	6	0	1	14
Ward	0	0	0	0	0	0	0	23
McQuitty	0	0	0	0	0	1	1	21
Model-based	0	1	9	0	3	1	3	6
Spectral	0	7	10	4	0	1	1	0
Gamma								
kmeans	0	0	10	6	5	2	0	0
PAM	0	0	23	0	0	0	0	0
Single	0	0	23	0	0	0	0	0
Complete	0	0	23	0	0	0	0	0
Average	0	0	23	0	0	0	0	0
Ward	0	0	23	0	0	0	0	0
McQuitty	0	0	23	0	0	0	0	0
Model-based	0	0	20	2	0	1	0	0
Spectral	0	0	18	2	1	0	2	0
C								
kmeans	0	0	10	6	5	2	0	0
PAM	0	0	23	0	0	0	0	0
Single	0	0	23	0	0	0	0	0
Complete	0	0	23	0	0	0	0	0
Average	0	0	23	0	0	0	0	0
Ward	0	0	23	0	0	0	0	0
McQuitty	0	0	23	0	0	0	0	0
Model-based	0	0	20	2	0	0	1	0
Spectral	0	0	18	2	1	0	2	0
KL								
kmeans	0	14	2	4	3	0	0	0
PAM	0	0	23	0	0	0	0	0
Single	0	19	1	0	1	1	1	0
Complete	0	23	0	0	0	0	0	0
Average	0	6	1	16	0	0	0	0
Ward	0	0	23	0	0	0	0	0

McQuitty	0	0	20	1	1	0	1	0
Model-based	0	20	1	1	0	1	0	0
Spectral	0	3	3	10	4	2	1	0
Gap								
kmeans	0	13	2	4	3	0	1	0
PAM	0	0	0	0	0	0	0	0
Single	0	0	0	16	5	2	0	0
Complete	0	0	0	0	0	1	0	1
Average	0	0	0	1	15	4	0	1
Ward	0	0	0	0	0	0	0	0
McQuitty	0	0	0	0	8	2	5	3
Jump								
$p/2$	0	0	0	1	22	0	0	0
$p/3$	0	0	0	1	22	0	0	0
$p/4$	0	0	0	1	22	0	0	0
$p/5$	0	0	0	1	22	0	0	0
$p/6$	0	0	0	1	22	0	0	0
$p/7$	0	0	0	1	22	0	0	0
PS								
kmeans	12	11	2	0	0	0	0	0
PAM	0	0	0	0	25	0	0	0
Single	0	0	0	4	4	8	7	2
Complete	0	17	0	3	5	0	0	0
Average	0	0	0	2	22	1	0	0
Ward	0	0	0	0	22	3	0	0
McQuitty	0	0	2	14	9	0	0	0
Model-based	6	1	1	17	0	0	0	0
Spectral	3	0	16	6	0	0	0	0
BI								
kmeans	0	2	0	0	1	2	1	9
PAM	0	24	0	0	0	0	0	0
Single	0	24	0	0	0	0	0	0
Complete	0	23	0	0	1	0	0	0
Average	0	24	0	0	0	0	0	0
Ward	0	24	0	0	0	0	0	0
McQuitty	0	24	0	0	0	0	0	0
Model-based	0	1	0	6	9	4	0	0
Spectral	0	0	13	0	0	4	3	0
CVNN								
kmeans	0	1	6	11	4	1	0	0
PAM	0	0	0	23	0	0	0	0
Single	0	0	0	22	1	0	0	0
Complete	0	0	0	21	1	1	0	0
Average	0	0	0	9	14	0	0	0
Ward	0	0	0	22	1	0	0	0
McQuitty	0	0	0	16	6	1	0	0
Model-based	0	1	0	22	0	0	0	0
Spectral	0	3	12	4	2	1	1	0
BIC								
Model-based	25	0	0	0	0	0	0	0
PAMSIL	0	0	25	0	0	0	0	0
ASW								
kmeans	0	0	11	13	1	0	0	0
PAM	0	0	25	0	0	0	0	0

Single	0	0	25	0	0	0	0	0	0	0	0	0
Complete	0	0	25	0	0	0	0	0	0	0	0	0
Average	0	0	25	0	0	0	0	0	0	0	0	0
Ward	0	0	25	0	0	0	0	0	0	0	0	0
McQuitty	0	0	25	0	0	0	0	0	0	0	0	0
Model-based	0	1	21	3	0	0	0	0	0	0	0	0
Spectral	0	1	21	2	1	0	0	0	0	0	0	0
OSil												
kmeans	0	0	11	13	1	0	0	0	0	0	0	0
PAM	0	0	25	0	0	0	0	0	0	0	0	0
Single	0	0	25	0	0	0	0	0	0	0	0	0
Complete	0	0	25	0	0	0	0	0	0	0	0	0
Average	0	0	25	0	0	0	0	0	0	0	0	0
Ward	0	0	25	0	0	0	0	0	0	0	0	0
McQuitty	0	0	0	0	0	0	0	0	0	0	0	0
Model-based	0	0	25	0	0	0	0	0	0	0	0	0
Spectral	0	2	20	2	1	0	0	0	0	0	0	0

Table C.18: Frequency counts for estimation of number of clusters \hat{k} from indices and clustering methods included in the study for Model 8. The true number of clusters are made bold.

No. of clusters	1	2	3	4	5	6	7	8	9	10	11	12
CH												
kmeans	0	0	0	0	0	0	0	0	3	2	8	12
PAM	0	0	0	0	0	0	0	0	0	2	2	21
Single	0	0	0	0	0	0	0	0	0	18	5	2
Complete	0	0	0	0	0	0	0	0	0	0	0	25
Average	0	0	0	0	0	0	0	0	0	10	4	11
Ward	0	0	0	0	0	0	0	0	0	0	0	25
McQuitty	0	0	0	0	0	0	0	0	0	0	4	21
Model-based	0	0	0	0	0	0	0	0	0	0	0	25
Spectral	0	0	0	0	0	1	0	0	2	6	5	11
H												
kmeans	0	5	0	7	1	6	6	0	0	0	0	0
PAM	0	0	0	0	0	0	0	0	0	25	0	0
Single	0	1	0	0	0	0	0	0	0	24	0	0
Complete	0	0	0	0	0	0	0	0	0	25	0	0
Average	0	0	0	0	0	0	0	0	2	23	0	0
Ward	0	0	0	0	0	0	0	0	0	25	0	0
McQuitty	0	0	0	0	0	0	0	0	0	25	0	0
Model-based	0	0	0	0	0	0	0	0	0	25	0	0
Spectral	0	6	9	6	2	2	0	0	0	0	0	0
Gamma												
kmeans	0	0	0	2	1	4	0	7	1	0	4	6
PAM	0	0	0	0	0	0	0	0	0	25	0	0
Single	0	0	0	0	0	0	0	0	0	25	0	0
Complete	0	0	0	0	0	0	0	0	0	25	0	0
Average	0	0	0	0	0	0	0	0	0	25	0	0
Ward	0	0	0	0	0	0	0	0	0	25	0	0
McQuitty	0	0	0	0	0	0	0	0	0	25	0	0

Model-based	0	0	0	0	0	0	0	0	0	25	0	0
Spectral	0	0	0	1	2	0	0	0	2	8	5	7
C												
kmeans	0	0	0	0	0	4	2	7	2	0	4	6
PAM	0	0	0	0	0	0	0	0	0	25	0	0
Single	0	0	0	0	0	0	0	0	0	25	0	0
Complete	0	0	0	0	0	0	0	0	0	25	0	0
Average	0	0	0	0	0	0	0	0	0	25	0	0
Ward	0	0	0	0	0	0	0	0	0	25	0	0
McQuitty	0	0	0	0	0	0	0	0	0	25	0	0
Model-based	0	0	0	0	0	0	0	0	0	25	0	0
Spectral	0	0	0	0	0	1	2	0	2	8	5	7
KL												
kmeans	0	6	9	0	4	1	1	0	4	0	0	0
PAM	0	0	0	0	0	0	0	25	0	0	0	0
Single	0	0	0	0	0	0	0	25	0	0	0	0
Complete	0	0	0	0	0	0	0	0	25	0	0	0
Average	0	0	0	0	0	0	0	0	25	0	0	0
Ward	0	0	0	0	0	0	0	25	0	0	0	0
McQuitty	0	0	0	0	0	0	0	25	0	0	0	0
Model-based	0	0	0	0	0	0	0	25	0	0	0	0
Spectral	0	5	4	3	5	3	2	0	3	0	0	0
Gap												
kmeans	0	0	5	0	7	0	6	4	3	0	0	0
PAM	0	0	0	0	0	0	0	0	0	0	25	0
Single	0	0	0	0	0	0	0	0	0	0	25	0
Complete	0	0	0	0	0	0	0	0	0	0	25	0
Average	0	0	0	0	0	0	0	0	0	0	25	0
Ward	0	0	0	0	0	0	0	0	0	0	25	0
McQuitty	0	0	0	0	0	0	0	0	0	0	25	0
Jump												
$p/2$	0	0	0	0	25	0	0	0	0	0	0	0
$p/3$	0	0	0	0	25	0	0	0	0	0	0	0
$p/4$	0	0	0	0	25	0	0	0	0	0	0	0
$p/5$	0	0	0	0	25	0	0	0	0	0	0	0
$p/6$	0	0	0	0	25	0	0	0	0	0	0	0
$p/7$	0	0	0	0	25	0	0	0	0	0	0	0
PS												
kmeans	0	25	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	0	0	0	0	0	0	25	0	0
Single	0	0	0	0	0	0	0	0	0	25	0	0
Complete	0	0	0	0	0	0	0	0	0	25	0	0
Average	0	0	0	0	0	0	0	0	0	25	0	0
Ward	0	0	0	0	0	0	0	0	0	25	0	0
McQuitty	0	0	0	0	0	0	0	0	0	25	0	0
Model-based	0	0	0	0	0	0	0	0	0	25	0	0
Spectral	0	7	2	1	0	0	0	0	0	0	0	0
BI												
kmeans	0	25	0	0	0	0	0	0	0	0	0	0
PAM	0	25	0	0	0	0	0	0	0	0	0	0
Single	0	25	0	0	0	0	0	0	0	0	0	0
Complete	0	0	4	15	0	0	0	1	0	5	0	0
Average	0	4	0	21	0	0	0	0	0	0	0	0
Ward	0	0	0	19	0	0	0	0	0	6	0	0

McQuitty	0	1	0	17	0	0	0	0	7	0	0
Model-based	0	25	0	0	0	0	0	0	0	0	0
Spectral	0	25	0	0	0	0	0	0	0	0	0
CVNN											
kmeans	0	0	0	0	2	3	5	5	0	0	0
PAM	0	0	7	13	0	0	0	4	0	0	0
Single	0	0	0	0	0	0	0	0	0	0	0
Complete	0	0	0	0	0	0	0	0	0	0	0
Average	0	0	0	0	0	0	0	0	0	0	0
Ward	0	0	0	0	0	0	0	0	0	0	0
McQuitty	0	0	0	0	0	0	0	0	0	0	0
Model-based	0	0	0	0	0	0	0	0	0	0	0
Spectral	0	0	3	6	7	9	0	0	0	0	0
BIC											
Model-based	0	0	0	0	0	0	0	0	25	0	0
PAMSIL	0	0	0	0	0	0	0	0	0	25	0
ASW											
kmeans	0	0	0	0	0	0	0	2	7	2	6
PAM	0	0	0	0	0	0	0	0	0	25	0
Single	0	0	0	0	0	0	0	0	0	25	0
Complete	0	0	0	0	0	0	0	0	0	25	0
Average	0	0	0	0	0	0	0	0	0	25	0
Ward	0	0	0	0	0	0	0	0	0	25	0
McQuitty	0	0	0	0	0	0	0	0	0	25	0
Model-based	0	0	0	0	0	0	0	0	0	25	0
Spectral	0	1	0	0	0	2	0	0	3	7	5
OSil											
kmeans	0	0	0	0	0	0	0	0	5	2	6
PAM	0	0	0	0	0	0	0	0	0	25	0
Single	0	0	0	0	0	0	0	0	0	25	0
Complete	0	0	0	0	0	0	0	0	0	25	0
Average	0	0	0	0	0	0	0	0	0	25	0
Ward	0	0	0	0	0	0	0	0	0	25	0
McQuitty	0	0	0	0	0	0	0	0	0	25	0
Model-based	0	0	0	0	0	0	0	0	0	25	0
Spectral	0	0	0	0	0	0	0	0	4	12	5

Table C.19: Frequency counts for estimation of number of clusters \hat{k} from indices and clustering methods included in the study for Model 9. The true number of clusters are made bold.

No. of clusters	1	2	3	4	5	6	7	8	9	10
CH										
kmeans	0	1	17	7	0	0	0	0	0	0
PAM	0	0	25	0	0	0	0	0	0	0
Single	0	0	25	0	0	0	0	0	0	0
Complete	0	0	25	0	0	0	0	0	0	0
Average	0	0	25	0	0	0	0	0	0	0
Ward	0	0	25	0	0	0	0	0	0	0
McQuitty	0	0	25	0	0	0	0	0	0	0
Model-based	0	0	25	0	0	0	0	0	0	0

Spectral	0	0	25	0	0	0	0	0	0	0
H										
kmeans	8	17	0	0	0	0	0	0	0	0
PAM	0	25	0	0	0	0	0	0	0	0
Single	0	25	0	0	0	0	0	0	0	0
Complete	0	25	0	0	0	0	0	0	0	0
Average	0	25	0	0	0	0	0	0	0	0
Ward	0	25	0	0	0	0	0	0	0	0
McQuitty	0	25	0	0	0	0	0	0	0	0
Model-based	0	25	0	0	0	0	0	0	0	0
Spectral	0	25	0	0	0	0	0	0	0	0
Gamma										
kmeans	0	0	17	7	0	1	0	0	5	20
PAM	0	0	25	0	0	0	0	0	0	0
Single	0	0	25	0	0	0	0	0	0	0
Complete	0	0	25	0	0	0	0	0	0	0
Average	0	0	25	0	0	0	0	0	0	0
Ward	0	0	25	0	0	0	0	0	0	0
McQuitty	0	0	25	0	0	0	0	0	0	0
Model-based	0	0	25	0	0	0	0	0	0	0
C										
kmeans	0	0	17	7	1	0	0	3	5	17
PAM	0	0	25	0	0	0	0	0	0	0
Single	0	0	25	0	0	0	0	0	0	0
Complete	0	0	25	0	0	0	0	0	0	0
Average	0	0	25	0	0	0	0	0	0	0
Ward	0	0	25	0	0	0	0	0	0	0
McQuitty	0	0	25	0	0	0	0	0	0	0
Model-based	0	0	25	0	0	0	0	0	0	0
Spectral	0	0	25	0	0	0	0	0	0	0
KL										
kmeans	0	8	10	5	2	0	0	0	0	0
PAM	0	7	4	7	7	0	0	0	0	0
Single	0	6	6	6	7	0	0	0	0	0
Complete	0	25	0	0	0	0	0	0	0	0
Average	0	25	0	0	0	0	0	0	0	0
Ward	0	10	4	4	7	0	0	0	0	0
McQuitty	0	4	5	6	10	0	0	0	0	0
Model-based	0	5	9	7	4	0	0	0	0	0
Spectral	0	11	8	4	2	0	0	0	0	0
Gap										
kmeans	0	8	17	0	0	0	0	0	0	0
PAM	0	0	25	0	0	0	0	0	0	0
Single	0	0	25	0	0	0	0	0	0	0
Complete	0	0	25	0	0	0	0	0	0	0
Average	0	0	24	1	0	0	0	0	0	0
Ward	0	0	25	0	0	0	0	0	0	0
McQuitty	0	0	24	1	0	0	0	0	0	0
Jump										
p/2	0	0	25	0	0	0	0	0	0	0
p/3	0	0	25	0	0	0	0	0	0	0
p/4	0	0	0	0	25	0	0	0	0	0
p/5	0	0	0	0	25	0	0	0	0	0
p/6	0	0	0	0	25	0	0	0	0	0

<i>p</i> /7	0	0	0	0	25	0	0	0	0	0
PS										
kmeans	8	1	10	6	0	0	0	0	0	0
PAM	4	0	21	0	0	0	0	0	0	0
Single	25	0	0	0	0	0	0	0	0	0
Complete	0	0	0	13	11	1	0	0	0	0
Average	0	0	0	0	5	4	2	5	2	7
Ward	0	0	3	22	0	0	0	0	0	0
McQuitty	0	0	0	0	2	2	7	4	3	7
Model-based	0	0	0	12	12	1	0	0	0	0
Spectral	0	0	25	0	0	0	0	0	0	0
BI										
kmeans	0	0	2	11	11	1	0	0	0	0
PAM	0	0	3	3	0	0	0	2	4	13
Single	0	24	0	1	0	0	0	0	0	0
Complete	0	0	25	0	0	0	0	0	0	0
Average	0	1	9	10	5	0	0	0	0	0
Ward	0	0	17	8	0	0	0	0	0	0
McQuitty	0	0	13	12	0	0	0	0	0	0
Model-based	0	0	15	10	0	0	0	0	0	0
Spectral	0	0	20	5	0	0	0	0	0	0
CVNN										
kmeans	0	8	17	0	0	0	0	0	0	0
PAM	0	0	25	0	0	0	0	0	0	0
Single	0	0	25	0	0	0	0	0	0	0
Complete	0	0	25	0	0	0	0	0	0	0
Average	0	0	25	0	0	0	0	0	0	0
Ward	0	0	25	0	0	0	0	0	0	0
McQuitty	0	0	25	0	0	0	0	0	0	0
Model-based	0	0	25	0	0	0	0	0	0	0
Spectral	0	25	0	0	0	0	0	0	0	0
BIC										
Model-based	0	0	25	0	0	0	0	0	0	0
PAMSIL	0	0	25	0	0	0	0	0	0	0
ASW										
kmeans	0	8	17	0	0	0	0	0	0	0
PAM	0	0	25	0	0	0	0	0	0	0
Single	0	0	25	0	0	0	0	0	0	0
Complete	0	0	25	0	0	0	0	0	0	0
Average	0	0	25	0	0	0	0	0	0	0
Ward	0	0	25	0	0	0	0	0	0	0
McQuitty	0	0	25	0	0	0	0	0	0	0
Model-based	0	0	25	0	0	0	0	0	0	0
Spectral	0	0	25	0	0	0	0	0	0	0
OSil										
kmeans	0	5	17	3	0	0	0	0	0	0
PAM	0	0	25	0	0	0	0	0	0	0
Single	0	0	25	0	0	0	0	0	0	0
Complete	0	0	25	0	0	0	0	0	0	0
Average	0	0	25	0	0	0	0	0	0	0
Ward	0	0	25	0	0	0	0	0	0	0
McQuitty	0	0	25	0	0	0	0	0	0	0
Model-based	0	0	25	0	0	0	0	0	0	0

Spectral	0	0	25	0	0	0	0	0	0	0
----------	---	---	----	---	---	---	---	---	---	---

Table C.20: Frequency counts for estimation of number of clusters \hat{k} from indices and clustering methods included in the study for Model 10. The true number of clusters are made bold.

No. of clusters	1	2	3	4	5	6	7	8	9	10
CH										
kmeans	0	0	0	0	0	1	12	7	5	0
PAM	0	0	0	0	0	0	25	0	0	0
Single	0	0	0	0	0	0	25	0	0	0
Complete	0	0	0	0	0	0	25	0	0	0
Average	0	0	0	0	0	0	25	0	0	0
Ward	0	0	0	0	0	0	25	0	0	0
McQuitty	0	0	0	0	0	0	25	0	0	0
Model-based	0	0	0	1	0	4	14	0	4	2
Spectral	0	8	3	1	3	3	2	1	1	3
H										
Kmeans	1	2	1	7	6	8	0	0	0	0
PAM	0	0	0	0	0	25	0	0	0	0
Single	0	0	0	0	0	25	0	0	0	0
Complete	0	0	0	0	0	25	0	0	0	0
Average	0	0	0	0	0	25	0	0	0	0
Ward	0	0	0	0	0	25	0	0	0	0
McQuitty	0	0	0	0	0	25	0	0	0	0
Model-based	2	3	3	1	2	9	0	0	0	0
Spectral	25	0	0	0	0	0	0	0	0	0
Gamma										
kmeans	0	0	0	0	4	5	13	2	1	0
PAM	0	0	0	0	0	0	25	0	0	0
Single	0	0	0	0	0	0	25	0	0	0
Complete	0	0	0	0	0	0	25	0	0	0
Average	0	0	0	0	0	0	25	0	0	0
Ward	0	0	0	0	0	0	25	0	0	0
McQuitty	0	0	0	0	0	0	25	0	0	0
Model-based	0	0	0	0	0	4	14	0	4	3
Spectral	0	0	0	0	0	0	0	0	0	0
C										
kmeans	0	0	0	0	4	5	13	2	1	0
PAM	0	0	0	0	0	0	25	0	0	0
Single	0	0	0	0	0	0	25	0	0	0
Complete	0	0	0	0	0	0	25	0	0	0
Average	0	0	0	0	0	0	25	0	0	0
Ward	0	0	0	0	0	0	25	0	0	0
McQuitty	0	0	0	0	0	0	25	0	0	0
Model-based	0	0	0	0	0	4	14	0	4	3
Spectral	0	0	0	0	0	0	0	0	0	25
KL										
kmeans	0	0	0	1	11	6	7	0	0	0
PAM	0	0	0	0	25	0	0	0	0	0
Single	0	0	0	0	25	0	0	0	0	0

Complete	0	0	0	0	0	25	0	0	0	0
Average	0	0	0	0	0	25	0	0	0	0
Ward	0	0	0	0	25	0	0	0	0	0
McQuitty	0	0	0	0	25	0	0	0	0	0
Model-based	0	1	0	4	15	1	4	0	0	0
Spectral	0	2	7	3	5	3	5	0	0	0
Gap										
kmeans	0	1	2	1	6	6	4	3	2	0
PAM	0	0	0	0	0	0	21	4	0	0
Single	0	0	0	0	0	0	24	1	0	0
Complete	0	0	0	0	0	0	25	0	0	0
Average	0	0	0	0	0	0	25	0	0	0
Ward	0	0	0	0	0	0	23	2	0	0
McQuitty	0	0	0	0	0	0	25	0	0	0
Jump										
$p/2$	0	0	0	0	25	0	0	0	0	0
$p/3$	0	0	0	0	25	0	0	0	0	0
$p/4$	0	0	0	0	25	0	0	0	0	0
$p/5$	0	0	0	0	25	0	0	0	0	0
$p/6$	0	0	0	0	25	0	0	0	0	0
$p/7$	0	0	0	0	25	0	0	0	0	0
PS										
kmeans	0	0	0	0	0	0	0	0	0	0
PAM	0	0	0	0	0	0	25	0	0	0
Single	0	0	0	0	0	0	0	0	0	25
Complete	0	0	0	0	0	0	25	0	0	0
Average	0	0	0	0	0	0	14	10	1	0
Ward	0	0	0	0	0	0	25	0	0	0
McQuitty	0	0	0	0	0	0	15	9	1	0
Model-based	0	0	0	0	0	2	2	4	7	10
Spectral	10	15	0	0	0	0	0	0	0	0
BI										
kmeans	0	0	0	0	9	15	1	0	0	0
PAM	0	0	0	0	0	0	25	0	0	0
Single	0	0	0	0	0	0	25	0	0	0
Complete	0	0	0	0	0	0	25	0	0	0
Average	0	0	0	0	0	0	25	0	0	0
Ward	0	0	0	0	0	0	25	0	0	0
McQuitty	0	0	0	0	0	0	25	0	0	0
Model-based	0	14	7	3	0	1	0	0	0	0
Spectral	0	3	0	3	8	9	0	0	0	2
CVNN										
Single	0	0	0	0	0	0	25	0	0	0
Complete	0	0	0	0	0	0	25	0	0	0
Average	0	0	0	0	0	0	25	0	0	0
Ward	0	0	0	0	0	0	25	0	0	0
McQuitty	0	0	0	0	0	0	25	0	0	0
kmeans	0	0	0	2	5	6	12	0	0	0
PAM	0	0	0	0	0	0	25	0	0	0
Model-based	0	1	2	2	1	5	14	0	0	0
Spectral	25	0	0	0	0	0	0	0	0	0
BIC										
Model-based	1	0	0	1	0	4	15	0	4	0
PAMSIL	0	0	0	0	0	0	25	0	0	0

	ASW									
kmeans	0	0	0	0	4	6	13	2	0	0
PAM	0	0	0	0	0	0	25	0	0	0
Single	0	0	0	0	0	0	25	0	0	0
Complete	0	0	0	0	0	0	25	0	0	0
Average	0	0	0	0	0	0	25	0	0	0
Ward	0	0	0	0	0	0	25	0	0	0
McQuitty	0	0	0	0	0	0	25	0	0	0
Model-based	0	0	0	1	0	4	14	0	4	2
Spectral	20	5	0	0	0	0	0	0	0	0
	OSil									
kmeans	0	0	0	0	0	0	20	4	1	0
PAM	0	0	0	0	0	0	25	0	0	0
Single	0	0	0	0	0	0	25	0	0	0
Complete	0	0	0	0	0	0	25	0	0	0
Average	0	0	0	0	0	0	25	0	0	0
Ward	0	0	0	0	0	0	25	0	0	0
McQuitty	0	0	0	0	0	0	25	0	0	0
Model-based	0	0	0	0	0	5	14	0	4	2
Spectral	0	24	1	0	0	0	0	0	0	0

Table C.21: Frequency table of indication of cluster estimation at correct level for all the indices in combination with all the methods for Model 1-10

Models	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
No.of dims.	2	2	2	2	2	5	10	500	1000	60	
No. of clusters	2	3	4	5	6	5	7	10	3	7	Overall
CH											
kmeans	18	2	0	4	4	7	5	2	17	12	74
PAM	14	1	0	0	0	25	1	2	25	25	93
Single	3	6	0	2	3	0	1	18	25	25	83
Complete	3	0	0	1	0	0	1	0	25	25	55
Average	6	1	0	11	0	3	0	10	25	25	81
Ward	6	0	0	0	2	25	1	0	25	25	84
McQuitty	2	0	0	3	0	0	1	0	25	25	56
Model-based	6	4	1	7	7	25	10	0	25	14	99
Spectral	19	8	1	9	8	7	3	6	25	2	78
											703
H											
kmeans	5	4	7	2	2	0	1	0	0	0	21
PAM	0	0	0	0	0	0	0	25	0	0	25
Single	0	0	1	3	2	0	0	24	0	0	30
Complete	4	0	2	0	2	3	0	25	0	0	36
Average	3	2	1	3	2	7	1	23	0	0	42
Ward	0	0	0	0	0	0	0	25	0	0	25
McQuitty	2	2	2	1	3	4	1	25	0	0	40
Model-based	8	5	5	3	1	3	3	25	0	0	53
Spectral	4	5	4	1	3	4	1	0	0	0	22
											294
Gamma											
kmeans	1	0	1	16	4	0	0	0	17	13	52
PAM	0	0	0	20	0	0	0	25	25	25	95

Single	0	0	1	1	2	0	0	25	25	25	79
Complete	0	0	0	14	0	0	0	25	25	25	89
Average	0	0	0	7	1	0	0	25	25	25	83
Ward	0	0	0	16	0	0	0	25	25	25	91
McQuitty	0	0	0	8	0	0	0	25	25	25	83
Model-based	0	0	0	13	0	0	0	25	25	14	77
Spectral	0	1	0	10	4	0	2	8	25	0	50
											699
C											
kmeans	1	0	1	17	5	0	0	0	17	13	54
PAM	0	0	0	18	0	0	0	25	25	25	93
Single	0	0	1	2	2	0	0	25	25	25	80
Complete	0	0	0	10	1	0	0	25	25	25	86
Average	0	0	0	6	3	0	0	25	25	25	84
Ward	0	0	0	14	0	0	0	25	25	25	64
McQuitty	0	0	0	7	3	0	0	25	25	25	88
Model-based	0	0	0	14	0	0	1	25	25	14	79
Spectral	0	0	0	11	5	0	2	8	25	0	51
											679
KL											
kmeans	8	3	0	2	5	0	0	0	10	7	35
PAM	7	4	4	4	7	0	0	0	4	0	30
Single	7	5	4	6	6	0	1	0	6	0	45
Complete	3	7	6	5	0	0	0	0	0	0	21
Average	5	2	3	1	7	0	0	0	0	0	18
Ward	6	2	5	1	5	0	0	0	4	0	23
McQuitty	0	6	6	0	2	0	1	0	5	0	20
Model-based	2	6	1	2	1	0	0	0	9	4	21
Spectral	2	2	3	6	5	0	1	0	8	5	16
											229
Gap											
kmeans	20	9	4	10	1	3	1	0	17	4	69
PAM	15	8	0	10	0	4	0	0	25	21	83
Single	3	3	0	2	0	4	0	0	25	24	61
Complete	14	5	3	16	0	3	0	0	25	25	92
Average	18	11	1	22	0	0	0	0	24	25	101
Ward	13	9	0	21	11	0	0	0	25	23	107
McQuitty	13	8	5	14	0	0	5	0	24	25	94
											607
Jump											
$p/2$	6	8	1	25	0	25	0	0	25	0	90
$p/3$	22	24	3	25	0	25	0	0	25	0	124
$p/4$	2	20	8	25	0	8	0	0	0	0	63
$p/5$	0	1	11	25	0	0	0	0	0	0	37
$p/6$	0	0	2	25	0	0	0	0	0	0	27
$p/7$	0	0	0	4	0	0	0	0	0	0	4
											345
PS											
kmeans	24	25	1	1	0	0	0	0	10	0	61
PAM	21	0	1	25	25	25	0	25	21	25	168
Single	12	2	9	0	0	14	7	25	0	0	69
Complete	0	0	0	20	0	0	0	25	0	25	70
Average	4	0	3	22	0	10	0	25	0	14	78
Ward	5	0	1	25	0	25	0	25	3	25	109

McQuitty	2	0	2	21	0	7	0	25	0	15	72
Model-based	25	25	20	24	0	25	0	25	0	2	146
Spectral	18	1	0	0	0	0	0	0	25	0	44
817											
	BI										
kmeans	25	22	0	5	1	0	1	0	2	1	47
PAM	6	0	4	24	6	0	0	0	3	25	68
Single	11	2	3	1	3	0	0	0	0	25	45
Complete	0	0	0	19	0	0	0	5	25	25	74
Average	0	0	0	21	0	0	0	0	9	25	55
Ward	0	0	0	23	0	0	0	6	17	25	71
McQuitty	0	0	1	17	0	0	0	7	13	25	63
Model-based	25	25	7	17	0	0	0	0	15	0	89
Spectral	5	1	0	1	0	0	3	0	20	0	30
542											
	CVNN										
kmeans	9	21	7	17	3	3	0	0	17	25	102
PAM	11	21	16	25	0	8	0	0	25	25	131
Single	11	4	0	2	0	0	0	0	25	25	67
Complete	12	8	5	21	1	0	0	0	25	25	97
Average	3	20	1	23	0	3	0	0	25	25	100
Ward	10	21	16	23	5	15	0	0	25	12	127
McQuitty	17	8	5	20	0	0	0	0	25	25	100
Model-based	8	24	19	25	0	15	0	0	25	14	130
Spectral	17	3	4	7	3	0	1	0	0	0	36
890											
	BIC										
Model-based	25	25	10	11	1	25	0	0	25	15	151
PAMSIL	20	7	2	25	25	0	0	25	25	25	154
	ASW										
kmeans	21	13	1	17	9	0	0	2	17	13	93
PAM	20	12	2	25	24	0	0	25	25	25	158
Single	4	5	0	2	0	0	0	25	25	25	86
Complete	7	1	0	20	0	0	0	25	25	25	103
Average	13	11	1	24	0	0	0	25	25	25	124
Ward	19	8	2	25	11	0	0	25	25	25	140
McQuitty	7	2	1	20	0	0	0	25	25	25	105
Model-based	20	13	3	24	0	0	0	25	25	14	124
Spectral	24	12	1	14	10	0	0	7	25	0	93
1026											
	OSil										
kmeans	21	12	2	17	9	3	0	2	17	20	103
PAM	20	9	2	25	25	0	0	25	25	25	156
Single	7	3	0	2	0	0	0	25	25	25	87
Complete	19	6	0	21	0	0	0	25	25	25	121
Average	17	9	1	24	0	0	0	25	25	25	126
Ward	20	10	2	25	12	0	0	25	25	25	139
McQuitty	15	7	1	22	0	0	0	25	25	25	120
Model-based	21	9	3	25	0	0	0	25	25	14	122
Spectral	22	7	1	14	6	0	0	12	25	0	87
1061											

Appendix D

Numerical example for Richness proof for ASW

In this appendix the richness property for the ASW index is proved using a numerical example. Several cases were considered in numerical examples before getting the final (two) categories that are presented as the cases in Figure 5.2. The **Example 1** below is structures into three parts. The **Example 1** represents a case from Figure 5.2 defined as: Cases for \mathcal{C} — Case 1: No singleton permitted — Cases for \mathcal{C}' — **Part 1** No singleton permitted — (a), and **Part 2** at least one singleton in \mathcal{C}' .

Part 3 represents a case where \mathcal{C}' was constructed by making the multiple moves between points at a time to observe the affect on ASW value. Several other unique possibilities were considered (not presented here) and the cases were identified that are most general and can cover the primary philosophy for richness proof for ASW. The cases were developed that are generalization for all the cases and covers all the unique arguments needed to prove richness. These cases are presented in Figure 5.2.

Example 1: As an example consider a small data set having 8 objects for clustering. Let the indices of the objects are: $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$. Define the distance as $d_{x_i \sim_C x_j}(x_i, x_j) = 1$ and $d_{x_i \approx_C x_j}(x_i, x_j) = 2$. Let say we have a 3 clusters¹ clustering of 8 data points as $C_1 : \{x_1, x_2\}$, $C_2 : \{x_3, x_4, x_5\}$, $C_3 : \{x_6, x_7, x_8\}$, where C_1 , C_2 and C_3 represents cluster 1, 2, and 3 respectively. Call this clustering by \mathcal{C}_k . We now compute ASW for this clustering.

Let we represent $S_i(\mathcal{C}_k, d)$ and $\bar{S}_i(\mathcal{C}_k, d)$ by a short hand notation as $S(i)$ and \bar{S} respectively for the examples only. For \mathcal{C}_k , all $a(i) = 1$, $b(i) = 2$, and $\bar{S}(\mathcal{C}_k, d) = 0.5$, such that $\bar{S} = 0.5$

¹The number of clusters can be any we assume 3 to make a simple example but big enough to capture the general situation. Obviously data set can also be of any size.

Table D.1 Distance of every object to all other object with regard to cluster membership of the objects for clustering \mathcal{C}_k . The distance d is used for calculations.

	$\{x_1 \ x_2\}$	$\{x_3 \ x_4 \ x_5\}$	$\{x_6 \ x_7 \ x_8\}$
x_1	0		
x_2	1	0	
x_3	2	2	0
x_4	2	2	1
x_5	2	2	1
x_6	2	2	2
x_7	2	2	2
x_8	2	2	2

All the possibilities for \mathcal{C}'_k are discussed below as parts now.

Part 1: As a first possibility to \mathcal{C}'_k assume that object x_3 is moved from C_2 to C_1 . The new clustering is now $\mathcal{C}'_k = \{C_1, C_2, C_3\} = \{\{x_1, x_2, x_3\}, \{x_4, x_5\}, \{x_6, x_7, x_8\}\}$. We compute its ASW as below.

$$\begin{aligned} a(x_1) &= \frac{1+2}{2} = 1.5 = a(x_2), & b(x_1) &= \min\left(\frac{2+2}{2}, \frac{2+2+2}{3}\right) = 2 = b(x_2). \\ a(x_3) &= \frac{2+2}{2} = 2, & b(x_3) &= \min\left(\frac{1+1}{2}, \frac{2+2+2}{3}\right) = 1. \\ a(x_4) &= \frac{1}{2} = 0.5 = a(x_5), & b(x_4) &= \min\left(\frac{2+2+1}{3}, \frac{2+2+2}{3}\right) = 1.67 = b(x_5). \\ a(x_6) &= \frac{1+1}{2} = 1 = a(G) = a(x_8). & b(x_6) &= \min\left(\frac{2+2+2}{3}, \frac{2+2}{2}\right) = 2 = b(x_7) = b(x_8). \end{aligned}$$

$$\begin{aligned} S(x_1) &= \frac{b(x_1) - a(x_1)}{\max(b(x_1), a(x_1))} = \frac{2-1.5}{2} = 0.25 = S(x_2), & S(x_4) &= \frac{1.67-0.5}{1.67} = 0.70 = S(x_5), \\ S(x_3) &= \frac{1-2}{2} = -0.5, & S(x_6) &= \frac{2-1}{2} = 0.5 = S(x_7) = S(x_8). \end{aligned}$$

The ASW for \mathcal{C}'_k is given as:

$$\bar{S} = \frac{2(0.25) - 0.5 + 2(0.70) + 3(0.5)}{8} = 0.3625.$$

Part 2: As a second possibility for \mathcal{C}'_k assume that object x_2 is moved to C_2 such that there is a single point cluster C_1 in the clustering say \mathcal{C}'_k . The clustering can be written as: $\{x_1\}, \{x_2, x_3, x_4, x_5\}, \{x_6, x_7, x_8\}$.

$$\begin{aligned} a(x_1) &= \text{undefined}, & b(x_1) &= \min\left(\frac{1+2+2+2}{4}, \frac{2+2+2}{3}\right) = 1.75, \\ a(x_2) &= \frac{2+2+2}{3} = 2, & b(x_2) &= \min\left(\frac{1}{2}, \frac{2+2+2}{3}\right) = 0.5, \\ a(x_3) &= \frac{2+1+1}{3} = 1.33 = a(x_4) = a(x_5), & b(x_3) &= \min\left(\frac{2}{2}, \frac{2+2+2}{3}\right) = 2 = b(x_4) = b(x_5), \\ a(x_6) &= \frac{1+1}{2} = 1 = a(x_7) = a(x_8), & b(x_6) &= \min\left(\frac{1}{1}, \frac{2+2+2+2}{4}\right) = 2 = b(x_7) = b(x_8). \\ S(x_1) &= 0, & S(x_3) &= \frac{2-1.33}{2} = 0.335 = S(x_4) = S(x_5), \\ S(x_2) &= \frac{0.5-2}{2} = -0.75, & S(x_5) &= \frac{2-1}{2} = 0.5 = S(x_7) = S(x_8). \end{aligned}$$

Such that the ASW for \mathcal{C}'_k is given as:

$$\bar{S} = \frac{-0.75 + 3(0.335) + 3(0.5)}{8} = 0.2194.$$

Note that as the single point clusters will increase ASW will further reduce. For instance if we move object x_2 from C_1 , x_3 and x_4 from C_2 to cluster C_3 from clustering \mathcal{C}_k we get $\bar{S} = 0.06$. The calculation for this is shown below as Part 3.

Part 3: In this part a case is developed by making multiple moves between points to construct \mathcal{C}' to observe the affect on ASW value. The moves are two observations that forms a cluster in \mathcal{C} are now in separate cluster in \mathcal{C}' such that while separating them one of them moves to cluster which has more points and other define singleton cluster. More than one singleton clusters were defined. The other singleton cluster is defined by the observations coming from a cluster in \mathcal{C} which has more than two points. The new clustering is now $\mathcal{C}'_k = \{\{x_1\}, \{x_5\}, \{x_2, x_3, x_4, x_6, x_7, x_8\}\}$.

$$\begin{aligned} a(x_1) &= a(x_5) = \text{undefined}, & b(x_1) &= \min\left(\frac{2}{1}, \frac{1+2+2+2+2+2}{6}\right) = 1.83. \\ a(x_2) &= \frac{2+2+2+2+2}{5} = 2, & b(x_2) &= \min\left(\frac{1}{2}, \frac{2}{1}\right) = 0.5, \\ a(x_3) &= \frac{2+1+2+2+2}{5} = 1.8 = a(x_4), & b(x_3) &= \min\left(\frac{2}{1}, \frac{1}{1}\right) = 1 = b(x_4). \\ a(x_6) &= \frac{2+2+\cancel{2}+1+1}{5} = 1.6 = a(x_7) = a(x_8), & b(x_5) &= \min\left(\frac{2}{1}, \frac{2+1+1+2+2+2}{6}\right) = , \\ & & b(x_6) &= b(x_7) = b(x_8) = \min\left(\frac{2}{1}, \frac{2}{1}\right) = 2. \end{aligned}$$

$$\begin{aligned} S(x_1) &= S(x_5) = 0, & S(x_3) &= \frac{1-1.8}{1.8} = -0.44 = S(x_4), \\ S(x_2) &= \frac{0.5-2}{2} = 0.75, & S(x_6) &= \frac{2-1.6}{2} = 0.2 = S(x_7) = S(x_8). \end{aligned}$$

The ASW for \mathcal{C}'_k is given as:

$$\bar{S} = \frac{0.75 + 2(-0.44) + 3(0.2)}{8} = 0.0588.$$

Bibliography

- Ackerman, M. (2012). Towards theoretical foundations of clustering.
- Ackerman, M. and S. Ben-David (2008). Which data sets are clusterable?-a theoretical study of clusterability. *preprint*.
- Ackerman, M., S. Ben-David, and D. Loker (2010a). Characterization of linkage-based clustering. In *COLT*, pp. 270–281.
- Ackerman, M., S. Ben-David, and D. Loker (2010b). Towards property-based classification of clustering paradigms. In *Advances in Neural Information Processing Systems*, pp. 10–18.
- Aloise, D., A. Deshpande, P. Hansen, and P. Popat (2009). Np-hardness of euclidean sum-of-squares clustering. *Machine learning* 75(2), 245–248.
- Alon, U., N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences* 96(12), 6745–6750.
- Arbelaitz, O., I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition* 46(1), 243–256.
- Arthur, D. and S. Vassilvitskii (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035. Society for Industrial and Applied Mathematics.
- Baker, F. B. and L. J. Hubert (1975). Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association* 70(349), 31–38.
- Ball, G. H. and D. J. Hall (1967). A clustering technique for summarizing multivariate data. *Behavioral Science* 12(2), 153–155.
- Bandyopadhyay, S., A. Mukhopadhyay, and U. Maulik (2007). An improved algorithm for clustering gene expression data. *Bioinformatics* 23(21), 2859–2865.

- Banfield, J. D. and A. E. Raftery (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, 803–821.
- Baraldi, A. and P. Blonda (1999). A survey of fuzzy clustering algorithms for pattern recognition. i. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 29(6), 778–785.
- Becker, R., J. Chambers, and A. Wilks (1988). The new s language (pacific grove, ca: Wadsworth & brooks/cole). *BeckerThe New S Language1988*.
- Ben-David, S. and M. Ackerman (2009). Measures of clustering quality: A working set of axioms for clustering. In *Advances in neural information processing systems*, pp. 121–128.
- Berkhin, P. (2006). A survey of clustering data mining techniques: recent advances in clustering. *Grouping Multidimensional Data* 25, 71.
- Bernard, E., P. Naveau, M. Vrac, and O. Mestre (2013). Clustering of maxima: Spatial dependencies among heavy rainfall in france. *Journal of Climate* 26(20), 7929–7937.
- Biase, F. H., X. Cao, and S. Zhong (2014). Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell rna sequencing. *Genome research* 24(11), 1787–1796.
- Binder, D. A. (1978). Bayesian cluster analysis. *Biometrika* 65(1), 31–38.
- Bock, H. H. (1996). Probabilistic models in cluster analysis. *Computational Statistics & Data Analysis* 23(1), 5–28.
- Bock, H.-H. (2008). Origins and extensions of the k-means algorithm in cluster analysis. *Electronic Journal for History of Probability and Statistics* 4(2), Article–14.
- Bolshakova, N. and F. Azuaje (2003). Cluster validation techniques for genome expression data. *Signal Processing* 83(4), 825–833.
- Bowcock, A. M., A. Ruiz-Linares, J. Tomfohrde, E. Minch, J. R. Kidd, and L. L. Cavalli-Sforza (1994). High resolution of human evolutionary trees with polymorphic microsatellites. *Nature* 368(6470), 455–457.
- Brun, M., C. Sima, J. Hua, J. Lowey, B. Carroll, E. Suh, and E. R. Dougherty (2007). Model-based evaluation of clustering validation measures. *Pattern Recognition* 40(3), 807–824.
- Butler, A., P. Hoffman, P. Smibert, E. Papalexi, and R. Satija (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*.

- Caliński, T. and J. Harabasz (1974). A dendrite method for cluster analysis. *Communications in Statistics-Theory and Methods* 3(1), 1–27.
- Campello, R. J. and E. R. Hruschka (2006). A fuzzy extension of the silhouette width criterion for cluster analysis. *Fuzzy Sets and Systems* 157(21), 2858–2875.
- Carlsson, G. and F. Mémoli (2010). Characterization, stability and convergence of hierarchical clustering methods. *Journal of Machine Learning Research* 11(Apr), 1425–1470.
- Carlsson, G. and F. Mémoli (2013). Classifying clustering schemes. *Foundations of Computational Mathematics* 13(2), 221–252.
- Celebi, M. E., H. A. Kingravi, and P. A. Vela (2013). A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications* 40(1), 200–210.
- Cleuz, G. and G. Govaert (1995). Gaussian parsimonious clustering models. *Pattern Recognition* 28(5), 781–793.
- Chen, G., S. A. Jaradat, N. Banerjee, T. S. Tanaka, M. S. Ko, and M. Q. Zhang (2002). Evaluation and comparison of clustering algorithms in analyzing es cell gene expression data. *Statistica Sinica* (1), 241–262.
- Chi, E. C. and K. Lange (2015). Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics* 24(4), 994–1013.
- Cho, Y.-J., A. Tsherniak, P. Tamayo, S. Santagata, A. Ligon, H. Greulich, R. Berhoukim, V. Amani, L. Goumnerova, C. G. Eberhart, et al. (2010). Integrative genomic analysis of medulloblastoma identifies a molecular subgroup that drives poor clinical outcome. *Journal of Clinical Oncology* 29(11), 1424–1430.
- Chuang, K.-S., H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen (2006). Fuzzy c-means clustering with spatial information for image segmentation. *Computerized Medical Imaging and Graphics* 30(1), 9–15.
- Chung, F. R. (1997). *Spectral graph theory*, Volume 92. Rhode Island: American Mathematical Society Providence.
- Cooley, D., P. Naveau, and P. Poncet (2006). Variograms for spatial max-stable random fields. In *Dependence in probability and statistics*, pp. 373–390. Springer.
- Cooley, R., B. Mobasher, and J. Srivastava (1997). Web mining: Information and pattern discovery on the world wide web. In *Tools with Artificial Intelligence, 1997. In Proceedings of Ninth IEEE International Conference on Tools with Artificial Intelligence*, pp. 558–567. IEEE.

- Cordeiro De Amorim, R. and P. Komisarczuk (2012). On initializations for the minkowski weighted k-means. *Advances in Intelligent Data Analysis XI*.
- Coretto, P and C. Hennig (2016). Robust improper maximum likelihood: tuning, computation, and a comparison with other methods for robust gaussian clustering. *Journal of the American Statistical Association* 111(516), 1648–1659.
- Correa-Morris, J. (2013). An indication of unification for different clustering approaches. *Pattern Recognition* 46(9), 2548–2561.
- Craddock, R. C., G. A. James, P. E. Holtzheimer, X. P. Hu, and H. S. Mayberg (2012). A whole brain fmri atlas generated via spatially constrained spectral clustering. *Human Brain Mapping* 33(8), 1914–1928.
- Cuesta-Albertos, J., A. Gordaliza, C. Matrán, et al. (1997). Trimmed k -means: An attempt to robustify quantizers. *The Annals of Statistics* 25(2), 553–576.
- Dasgupta, A. and A. E. Raftery (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association* 93(441), 294–302.
- Davies, D. L. and D. W. Bouldin (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2), 224–227.
- Davis, L. (1991). Handbook of genetic algorithms.
- Day, W. H. and H. Edelsbrunner (1984). Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification* 1(1), 7–24.
- de Amorim, R. C. and C. Hennig (2015). Recovering the number of clusters in data sets with noise features using feature rescaling factors. *Information Sciences* 324, 126–145.
- de Amorim, R. C. and P. Komisarczuk (2012). On initializations for the minkowski weighted k-means. In *International Symposium on Intelligent Data Analysis*, pp. 45–55. Springer.
- Defays, D. (1977). An efficient algorithm for a complete link method. *The Computer Journal* 20(4), 364–366.
- Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1–38.
- Deza, M. M. and E. Deza (2009). *Encyclopedia of distances*. Springer.

- Dhillon, I. S., Y. Guan, and B. Kulis (2004). *A unified view of kernel k-means, spectral clustering and graph cuts*. Citeseer.
- Ding, C. H., X. He, H. Zha, M. Gu, and H. D. Simon (2001). A min-max cut algorithm for graph partitioning and data clustering. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on Data Mining*, pp. 107–114. IEEE.
- Dortet-Bernadet, J.-L. and N. Wicker (2007). Model-based clustering on the unit sphere with an illustration using gene expression profiles. *Biostatistics* 9(1), 66–80.
- Dunn, G. and B. S. Everitt (2004). *An introduction to mathematical taxonomy*. Courier Corporation.
- Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics* 4(1), 95–104.
- Eddelbuettel, D., R. François, J. Allaire, J. Chambers, D. Bates, and K. Ushey (2011). Rcpp: Seamless r and c++ integration. *Journal of Statistical Software* 40(8), 1–18.
- Edwards, A. W. and L. L. Cavalli-Sforza (1965). A method for cluster analysis. *Biometrics*, 362–375.
- Eisen, M. B., P. T. Spellman, P. O. Brown, and D. Botstein (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences* 95(25), 14863–14868.
- Ester, M., H.-P. Kriegel, J. Sander, X. Xu, et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, Volume 96, pp. 226–231.
- Everitt, B. S., S. Landau, M. Leese, and D. Stahl (2011). *Cluster analysis* (5 ed.). Chichester, UK: Wiley.
- Fang, Y. and J. Wang (2012). Selection of the number of clusters via the bootstrap method. *Computational Statistics & Data Analysis* 56(3), 468–477.
- Fayyad, U. M., C. Reina, and P. S. Bradley (1998). Initialization of iterative refinement clustering algorithms. In *KDD*, pp. 194–198.
- Filippone, M., F. Camastra, F. Masulli, and S. Rovetta (2008). A survey of kernel and spectral methods for clustering. *Pattern Recognition* 41(1), 176–190.
- Firdaus, S. and M. A. Uddin (2015). A survey on clustering algorithms and complexity analysis. *International Journal of Computer Science Issues (IJCSI)* 12(2), 62.

- Fisher, L. and J. W. V. Ness (1971). Admissible clustering procedures. *Biometrika* 58(1), 91–104.
- Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics* 21, 768–769.
- Fraley, C. and A. E. Raftery (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal* 41(8), 578–588.
- Fraley, C. and A. E. Raftery (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association* 97(458), 611–631.
- Franck, P., E. Cameron, G. Good, J.-Y. RASPLUS, and B. Oldroyd (2004). Nest architecture and genetic differentiation in a species complex of australian stingless bees. *Molecular Ecology* 13(8), 2317–2331.
- Fred, A. L. and J. M. Leitão (2003). A new cluster isolation criterion based on dissimilarity increments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(8), 944–958.
- Fridlyand, J. and S. Dudoit (2001). Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method. Technical report, 600, Department of Statistics, UC Berkeley.
- Friedman, J., T. Hastie, and R. Tibshirani (2001). *The elements of statistical learning*, Volume 1. Springer series in statistics New York.
- Fujita, A., D. Y. Takahashi, and A. G. Patriota (2014). A non-parametric method to estimate the number of clusters. *Computational Statistics & Data Analysis* 73, 27–39.
- Ganesan, T. B. and R. Sukanesh (2008). Segmentation of brain mr images using fuzzy clustering method with silhouette method. *Journal of Engineering and Applied Sciences* 3, 792–795.
- Garey, M., D. Johnson, and H. Witsenhausen (1982). The complexity of the generalized lloyd-max problem (corresp.). *IEEE Transactions on Information Theory* 28(2), 255–256.
- Ghahramani, Z. and M. I. Jordan (1995). Learning from incomplete data.
- Gionis, A., H. Mannila, and P. Tsaparas (2007). Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data* 1(1), 4.
- Goolam, M., A. Scialdone, S. J. Graham, I. C. Macaulay, A. Jedrusik, A. Hupalowska, T. Voet, J. C. Marioni, and M. Zernicka-Goetz (2016). Heterogeneity in oct4 and sox2 targets biases cell fate in 4-cell mouse embryos. *Cell* 165(1), 61–74.

- Gordon, A. (1982). Classification: Methods for the exploratory analysis of multivariate data. *Routledge Chapman & Hall*.
- Graham, R. L. and P. Hell (1985). On the history of the minimum spanning tree problem. *Annals of the History of Computing* 7(1), 43–57.
- Guha, S., R. Rastogi, and K. Shim (1998). Cure: an efficient clustering algorithm for large databases. In *ACM SIGMOD Record*, Volume 27, pp. 73–84. ACM.
- Guha, S., R. Rastogi, and K. Shim (1999). Rock: A robust clustering algorithm for categorical attributes. In *Proceedings of 15th International Conference on Data Engineering*, pp. 512–521.
- Hagen, L. and A. B. Kahng (1992). New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems* 11(9), 1074–1085.
- Hahsler, M. and M. Piekenbrock (2018). *dbSCAN: Density Based Clustering of Applications with Noise (DBSCAN) and Related Algorithms*. R package version 1.1-3.
- Halkidi, M., Y. Batistakis, and M. Vazirgiannis (2001). On clustering validation techniques. *Journal of Intelligent Information Systems* 17(2-3), 107–145.
- Halkidi, M. and M. Vazirgiannis (2001). Clustering validity assessment: Finding the optimal partitioning of a data set. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on Data Mining*, pp. 187–194. IEEE.
- Halkidi, M. and M. Vazirgiannis (2008). A density-based cluster validity approach using multi-representatives. *Pattern Recognition Letters* 29(6), 773–786.
- Halkidi, M., M. Vazirgiannis, and Y. Batistakis (2000). Quality scheme assessment in the clustering process. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 265–276. Springer.
- Han, J., J. Pei, and M. Kamber (2011). *Data mining: concepts and techniques*. Elsevier.
- Handl, J. and J. Knowles (2007). An evolutionary approach to multiobjective clustering. *IEEE transactions on Evolutionary Computation* 11(1), 56–76.
- Handl, J., J. Knowles, and D. B. Kell (2005). Computational cluster validation in post-genomic data analysis. *Bioinformatics* 21(15), 3201–3212.
- Hartigan, J. A. (1975). *Clustering algorithms*. New York, USA: John Wiley and Sons.

- Hartigan, J. A. and M. A. Wong (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28(1), 100–108.
- Hausdorf, B. (2011). Progress toward a general species concept. *Evolution* 65(4), 923–931.
- Heller, R., D. Stanley, D. Yekutieli, N. Rubin, and Y. Benjamini (2006). Cluster-based analysis of fmri data. *NeuroImage* 33(2), 599–608.
- Hennig, C. (2014). How many bee species? a case study in determining the number of clusters. In *Data Analysis, Machine Learning and Knowledge Discovery*, pp. 41–49. Springer.
- Hennig, C. (2015a). *fpc: Flexible Procedures for Clustering*. R package version 2.1-10.
- Hennig, C. (2015b). What are the true clusters? *Pattern Recognition Letters* 64, 53–62.
- Hennig, C. (2017). Cluster validation by measurement of clustering characteristics relevant to the user. *arXiv preprint arXiv:1703.09282*.
- Hennig, C. and B. Hausdorf (2015). *prabclus: Functions for Clustering of Presence-Absence, Abundance and Multilocus Genetic Data*. R package version 2.2-6.
- Hennig, C., M. Meila, F. Murtagh, and R. Rocci (2015). *Handbook of cluster analysis*. Boca Rataon, USA: CRC Press.
- Hinneburg, A. and H.-H. Gabriel (2007). Denclue 2.0: Fast clustering based on kernel density estimation. In *International symposium on intelligent data analysis*, pp. 70–80. Springer.
- Hruschka, E. R., R. J. Campello, A. A. Freitas, et al. (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 39(2), 133–155.
- Hruschka, E. R. and N. F. Ebecken (2003). A genetic algorithm for cluster analysis. *Intelligent Data Analysis* 7(1), 15–25.
- Huang, Z. (1997). A fast clustering algorithm to cluster very large categorical data sets in data mining. Volume 3, pp. 34–39. Citeseer.
- Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery* 2(3), 283–304.
- Hubert, L. and P. Arabie (1985). Comparing partitions. *Journal of Classification* 2(1), 193–218.

- Hubert, L. and J. Schultz (1976). Quadratic assignment as a general data analysis strategy. *British Journal of Mathematical and Statistical Psychology* 29(2), 190–241.
- Hwang, B., J. H. Lee, and D. Bang (2018). Single-cell rna sequencing technologies and bioinformatics pipelines. *Experimental & molecular medicine* 50(8), 96.
- Ignaccolo, R., S. Ghigo, and E. Giovenali (2008). Analysis of air quality monitoring networks by functional clustering. *Environmetrics* 19(7), 672–686.
- Jain, A. K. and R. C. Dubes (1988a). *Algorithms for clustering data*. Prentice-Hall, Inc.
- Jain, A. K. and R. C. Dubes (1988b). Cluster validity. In *Algorithms for clustering data*, Chapter 4, pp. 143–222. Prentice-Hall, Inc.
- Jain, A. K. and M. H. Law (2005). Data clustering: A user's dilemma. In *International conference on pattern recognition and machine intelligence*, pp. 1–10. Springer.
- Jain, A. K., M. N. Murty, and P. J. Flynn (1999). Data clustering: a review. *ACM Computing Surveys (CSUR)* 31(3), 264–323.
- Jardine, N. and R. Sibson (1968). The construction of hierachic and non-hierachic classifications. *The Computer Journal* 11(2), 177–184.
- Jeffreys, H. (1935). Some tests of significance, treated by the theory of probability. In *Proceedings of the Cambridge Philosophical Society*, Volume 31, pp. 203–222.
- Jiang, D., C. Tang, and A. Zhang (2004). Cluster analysis for gene expression data: a survey. *IEEE Transactions on knowledge and data engineering* 16(11), 1370–1386.
- Kalpakis, K., D. Gada, and V. Puttagunta (2001). Distance measures for effective clustering of arima time-series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on Data Mining*, pp. 273–280. IEEE.
- Kannan, S. (2008). A new segmentation system for brain mr images based on fuzzy techniques. *Applied Soft Computing* 8(4), 1599–1606.
- Kannan, S., S. Ramathilagam, A. Sathya, and R. Pandiyarajan (2010). Effective fuzzy c-means based kernel function in segmenting medical images. *Computers in Biology and Medicine* 40(6), 572–579.
- Karypis, G., E.-H. Han, and V. Kumar (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer* 32(8), 68–75.
- Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association* 90(430), 773–795.

- Kaufman, L. and P. Rousseeuw (1987). *Clustering by means of medoids*. Amsterdam: North-Holland.
- Kaufman, L. and P. J. Rousseeuw (1990). *Finding groups in data: an introduction to cluster analysis*, Volume 344. John Wiley & Sons.
- Kennedy, G. C., H. Matsuzaki, S. Dong, W.-m. Liu, J. Huang, G. Liu, X. Su, M. Cao, W. Chen, J. Zhang, et al. (2003). Large-scale genotyping of complex dna. *Nature Biotechnology* 21(10), 1233–1237.
- Kiselev, V. Y., T. S. Andrews, and M. Hemberg (2019). Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 1.
- Kiselev, V. Y., K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, K. N. Natarajan, W. Reik, M. Barahona, A. R. Green, and M. Hemberg (2017). Sc3 - consensus clustering of single-cell rna-seq data. *Nature Methods*.
- Kleinberg, J. M. (2003). An impossibility theorem for clustering. In *Advances in neural information processing systems*, pp. 463–470.
- Kolodziejczyk, A. A., J. K. Kim, J. C. Tsang, T. Ilicic, J. Henriksson, K. N. Natarajan, A. C. Tuck, X. Gao, M. Bühler, P. Liu, et al. (2015). Single cell rna-sequencing of pluripotent states unlocks modular transcriptional variation. *Cell stem cell* 17(4), 471–485.
- Krzanowski, W. J. and Y. Lai (1988). A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics*, 23–34.
- Lance, G. N. and W. T. Williams (1967). A general theory of classificatory sorting strategies: 1. hierarchical systems. *The computer journal* 9(4), 373–380.
- Lei, Y., J. C. Bezdek, S. Romano, N. X. Vinh, J. Chan, and J. Bailey (2017). Ground truth bias in external cluster validity indices. *Pattern Recognition* 65(1), 58–70.
- Leisch, F. and E. Dimitriadou (2010). *mlbench: Machine Learning Benchmark Problems*. R package version 2.1-1.
- Leskovec, J., K. J. Lang, and M. Mahoney (2010). Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pp. 631–640. ACM.
- Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern Recognition* 38(11), 1857–1874.
- Liu, W.-m., X. Di, G. Yang, H. Matsuzaki, J. Huang, R. Mei, T. B. Ryder, T. A. Webster, S. Dong, G. Liu, et al. (2003). Algorithms for large-scale genotyping microarrays. *Bioinformatics* 19(18), 2397–2403.

- Liu, Y., Z. Li, H. Xiong, X. Gao, J. Wu, S. Wu, et al. (2013). Understanding and enhancement of internal clustering validation measures. *IEEE Transactions on Cybernetics* 43(3), 982–994.
- Lleti, R., M. C. Ortiz, L. A. Sarabia, and M. S. Sánchez (2004). Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes. *Analytica Chimica Acta* 515(1), 87–100.
- Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory* 28(2), 129–137.
- Lovmar, L., A. Ahlford, M. Jonsson, and A.-C. Syvänen (2005). Silhouette scores for assessment of SNP genotype clusters. *BMC Genomics* 6(1), 35.
- Lucasius, C. B., A. D. Dane, and G. Kateman (1993). On k-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasibility and comparison. *Analytica Chimica Acta* 282(3), 647–669.
- Lun, A. and D. Risso (2017). *SingleCellExperiment: S4 Classes for Single Cell Data*. R package version 1.0.0.
- Lun, A. T., D. J. McCarthy, and J. C. Marioni (2016a). A step-by-step workflow for low-level analysis of single-cell RNA-seq data with bioconductor. *F1000Research* 5.
- Lun, A. T. L., D. J. McCarthy, and J. C. Marioni (2016b). A step-by-step workflow for low-level analysis of single-cell RNA-seq data with bioconductor. *F1000Res.* 5, 2122.
- MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Volume 1, pp. 281–297. Oakland, CA, USA.
- Maechler, M., P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik (2017). *cluster: Cluster Analysis Basics and Extensions*. R package version 2.0.6.
- Mahajan, M., P. Nimbhorkar, and K. Varadarajan (2009). The planar k-means problem is np-hard. In *International Workshop on Algorithms and Computation*, pp. 274–285. Springer.
- Malliaros, F. D. and M. Vazirgiannis (2013). Clustering and community detection in directed networks: A survey. *Physics Reports* 533(4), 95–142.
- Mardis, E. R. (2008). The impact of next-generation sequencing technology on genetics. *Trends in genetics* 24(3), 133–141.

- Margareta Ackerman, Shai Ben-David, S. B. and D. Loker (2012). Weighted clustering. In *Proceeding of 26th AAAI Conference on Artificial Intelligence*.
- Maulik, U. and S. Bandyopadhyay (2000). Genetic algorithm-based clustering technique. *Pattern recognition* 33(9), 1455–1465.
- McCarthy, D. J., K. R. Campbell, A. T. L. Lun, and Q. F. Wills (2017). Scater: pre-processing, quality control, normalisation and visualisation of single-cell rna-seq data in r. *Bioinformatics* 14 Jan.
- McQuitty, L. L. (1957). Elementary linkage analysis for isolating orthogonal and oblique types and typal relevancies. *Educational and Psychological Measurement* 17(2), 207–229.
- McQuitty, L. L. (1966). Similarity analysis by reciprocal pairs for discrete and continuous data. *Educational and Psychological measurement* 26(4), 825–831.
- Meila, M. (2015). Spectral clustering. In C. Hennig, M. Meila, F. Murtagh, and R. Rocci (Eds.), *Handbook of cluster analysis*, Chapter 7, pp. 125–141. Boca Rataon, USA: CRC Press.
- Meila, M. and J. Shi (2000). Learning segmentation by random walks. In *NIPS*, Volume 14.
- Menardi, G. (2011). Density-based silhouette diagnostics for clustering methods. *Statistics and Computing* 21(3), 295–308.
- Merendino, S. and M. E. Celebi (2013). A simulated annealing clustering algorithm based on center perturbation using gaussian mutation. In *The Twenty-Sixth International FLAIRS Conference*.
- Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics* 21(6), 1087–1092.
- Milligan, G. W. (1981). A monte carlo study of thirty internal criterion measures for cluster analysis. *Psychometrika* 46(2), 187–199.
- Milligan, G. W. and M. C. Cooper (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika* 50(2), 159–179.
- Mimmack, G. M., S. J. Mason, and J. S. Galpin (2001). Choice of distance matrices in cluster analysis: Defining regions. *Journal of climate* 14(12), 2790–2797.

- Mohar, B., Y. Alavi, G. Chartrand, and O. Oellermann (1991). The laplacian spectrum of graphs. *Graph Theory, Combinatorics, and Applications* 2(871-898), 12.
- Moulavi, D., P. A. Jaskowiak, R. J. Campello, A. Zimek, and J. Sander (2014). Density-based clustering validation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pp. 839–847. SIAM.
- Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The computer journal* 26(4), 354–359.
- Neumann, J., D. Cramon, and G. Lohmann (2008). Model-based clustering of meta-analytic functional imaging data. *Human Brain Mapping* 29(2), 177–192.
- Newton, S. C., S. Pemmaraju, and S. Mitra (1992). Adaptive fuzzy leader clustering of complex data sets in pattern recognition. *IEEE Transactions on Neural Networks* 3(5), 794–800.
- Ng, A. Y., M. I. Jordan, and Y. Weiss (2001). On spectral clustering: Analysis and an algorithm. In *NIPS*, Volume 14, pp. 849–856.
- Ng, R. T. and J. Han (1994). Efficient and effective clustering methods for spatial data mining. In *VLDB 94 proceedings of the 20th international conference on very large data bases*, pp. 144–155. Morgan Kaufmann Publishers San Francisco, USA.
- Nguyen, T., A. Bhatti, A. Khosravi, S. Haggag, D. Creighton, and S. Nahavandi (2015). Automatic spike sorting by unsupervised clustering with diffusion maps and silhouettes. *Neurocomputing* 153, 199–210.
- Park, H.-S. and C.-H. Jun (2009). A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications* 36(2), 3336–3341.
- Patterson, N., A. L. Price, and D. Reich (2006). Population structure and eigenanalysis. *PLoS genetics* 2(12), e190.
- Pollard, D. et al. (1981). Strong consistency of k -means clustering. *The Annals of Statistics* 9(1), 135–140.
- Puzicha, J., T. Hofmann, and J. M. Buhmann (2000). A theory of proximity based clustering: Structure detection by optimization. *Pattern Recognition* 33(4), 617–634.
- R Core Team (2015). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association* 66(336), 846–850.

- Rani, S. and G. Sikka (2012). Recent techniques of clustering of time series data: a survey. *International Journal of Computer Applications* 52(15).
- Recupero, D. R. (2007). A new unsupervised method for document clustering by using wordnet lexical and conceptual relations. *Information Retrieval* 10(6), 563–579.
- Reynolds, A. P., G. Richards, B. de la Iglesia, and V. J. Rayward-Smith (2006). Clustering rules: a comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms* 5(4), 475–504.
- Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20, 53–65.
- Rubin, J. (1967). Optimal classification into groups: an approach for solving the taxonomy problem. *Journal of Theoretical Biology* 15(1), 103–144.
- Ruspini, E. H. (1969). A new approach to clustering. *Information and Control* 15(1), 22–32.
- Saitta, S., B. Raphael, and I. F. Smith (2007). A bounded index for cluster validity. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 174–187. Springer.
- Schaeffer, S. E. (2007). Graph clustering. *Computer science review* 1(1), 27–64.
- Schubert, E. and P. J. Rousseeuw (2018). Faster k-medoids clustering: Improving the pam, clara, and clarans algorithms. *arXiv preprint arXiv:1810.05691*.
- Scrucca, L., M. Fop, T. B. Murphy, and A. E. Raftery (2017). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal* 8(1), 205–233.
- Scrucca, L. and A. E. Raftery (2015). Improved initialisation of model-based clustering using gaussian hierarchical partitions. *Advances in Data Analysis and Classification* 9(4), 447–460.
- Sculley, D. (2010). Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pp. 1177–1178. ACM.
- Shapiro, E., T. Biezuner, and S. Linnarsson (2013). Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nature Reviews Genetics* 14(9), 618.
- Sheng, W. and X. Liu (2006). A genetic k-medoids clustering algorithm. *Journal of Heuristics* 12(6), 447–466.

- Shi, J. and J. Malik (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8), 888–905.
- Sibson, R. (1973). Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal* 16(1), 30–34.
- Slonim, N., E. Aharoni, and K. Crammer (2013). Hartigan’s k-means versus lloyd’s k-means—is it time for a change? In *IJCAI*, pp. 1677–1684.
- Sokal, R. R. and C. D. Michener (1958). A statistical method for evaluating systematic relationships. *University Kansas Science Bulletin* 38(22), 1409–1438.
- Sørensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biologiske Skrifter* 5(4), 1–34.
- Srivastava, J., R. Cooley, M. Deshpande, and P.-N. Tan (2000). Web usage mining: Discovery and applications of usage patterns from web data. *Acm Sigkdd Explorations Newsletter* 1(2), 12–23.
- Steinhaeuser, K., N. Chawla, and A. Ganguly (2011). Comparing predictive power in climate data: Clustering matters. *Advances in Spatial and Temporal Databases*, 39–55.
- Steinhaus, H. (1956). Sur la division des corp materiels en parties. *Bulletin de L’academie Polonaise des Sciences* 4(12), 801–804.
- Strehl, A. and J. Ghosh (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research* 3(Dec), 583–617.
- Sturn, A., J. Quackenbush, and Z. Trajanoski (2002). Genesis: cluster analysis of microarray data. *Bioinformatics* 18(1), 207–208.
- Sugar, C. and G. M. James (2003a). Documentation for the r-code to implement the jump methodology in “finding the number of clusters in a data set: An information theoretic approach”. *Marshall School of Business, University of California*.
- Sugar, C. A. and G. M. James (2003b). Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association* 98(463), 750–763.
- Tang, F., C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. B. Tuch, A. Siddiqui, et al. (2009). mrna-seq whole-transcriptome analysis of a single cell. *Nature methods* 6(5), 377.

- Thirion, B., G. Varoquaux, E. Dohmatob, and J.-B. Poline (2014). Which fmri clustering gives good brain parcellations? *Frontiers in Neuroscience* 8.
- Tibshirani, R. and G. Walther (2005). Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics* 14(3), 511–528.
- Tibshirani, R., G. Walther, and T. Hastie (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63(2), 411–423.
- Tramacere, A., D. Paraficz, P. Dubath, J.-P. Kneib, and F. Courbin (2016). Asterism-application of topometric clustering algorithms in automatic galaxy detection and classification. *arXiv preprint arXiv:1609.06728*.
- Troyanskaya, O., M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman (2001). Missing value estimation methods for dna microarrays. *Bioinformatics* 17(6), 520–525.
- Ultsch, A. (2005). Clustering with som: U*c. In *Proceeding of Workshop on Self-Organizing Maps, Paris, France*, 75–82.
- Uludag, U., A. Ross, and A. Jain (2004). Biometric template selection and update: a case study in fingerprints. *Pattern Recognition* 37(7), 1533–1542.
- Vallejos, C. A., D. Risso, A. Scialdone, S. Dudoit, and J. C. Marioni (2017). Normalizing single-cell rna sequencing data: challenges and opportunities. *Nature methods* 14(6), 565.
- Van der Laan, M., K. Pollard, and J. Bryan (2003). A new partitioning around medoids algorithm. *Journal of Statistical Computation and Simulation* 73(8), 575–584.
- Van't Veer, L. J., H. Dai, M. J. Van De Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. Van Der Kooy, M. J. Marton, A. T. Witteveen, et al. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature* 415(6871), 530–536.
- Villani, A.-C., R. Satija, G. Reynolds, S. Sarkizova, K. Shekhar, J. Fletcher, M. Griesbeck, A. Butler, S. Zheng, S. Lazo, et al. (2017). Single-cell rna-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science* 356(6335), eaah4573.
- Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing* 17(4), 395–416.
- Von Luxburg, U., R. C. Williamson, and I. Guyon (2012). Clustering: Science or art? In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pp. 65–79.

- Wagstaff, K. (2004). Clustering with missing values: No imputation required. In *Classification, Clustering and Data Mining Applications. Proceedings of the Meeting of the International Federation of Classification Societies (IFCS), Illinois Institute of Technology, Chicago, USA*, pp. 649–658.
- Walesiak, M. and A. Dudek (2017). *clusterSim: Searching for Optimal Clustering Procedure for a Data Set*. R package version 0.47-1.
- Walker, A. J. (1974). New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters* 10(8), 127–128.
- Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association* 58(301), 236–244.
- Willett, P. (1988). Recent trends in hierachic document clustering: a critical review. *Information Processing & Management* 24(5), 577–597.
- Wright, W. E. (1973). A formalization of cluster analysis. *Pattern Recognition* 5(3), 273–282.
- Xiong, H. and Z. Li (2013). Clustering validation measures. In C. Aggarwal and C. Reddy (Eds.), *Data Clustering Algorithm and Applications*, Chapter 23, pp. 571–606. Boca Raton, USA: CRC Press.
- Yan, L., M. Yang, H. Guo, L. Yang, J. Wu, R. Li, P. Liu, Y. Lian, X. Zheng, J. Yan, et al. (2013). Single-cell rna-seq profiling of human preimplantation embryos and embryonic stem cells. *Nature structural & molecular biology* 20(9), 1131.
- Ye, Z., S. Hu, and J. Yu (2008). Adaptive clustering algorithm for community detection in complex networks. *Physical Review E* 78(4), 046115.
- Yeung, K. Y., C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo (2001). Model-based clustering and data transformations for gene expression data. *Bioinformatics* 17(10), 977–987.
- Zadeh, R. B. and S. Ben-David (2009). A uniqueness theorem for clustering. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pp. 639–646. AUAI Press.
- Zeileis, A., K. Hornik, A. Smola, and A. Karatzoglou (2004). kernlab-an s4 package for kernel methods in r. *Journal of Statistical Software* 11(9), 1–20.
- Zeisel, A., A. B. Muñoz-Manchado, S. Codeluppi, P. Lönnerberg, G. La Manno, A. Juréus, S. Marques, H. Munguba, L. He, C. Betsholtz, et al. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science* 347(6226), 1138–1142.

- Zelnik-Manor, L. and P. Perona (2004). Self-tuning spectral clustering. In *NIPS*, Volume 17, pp. 16.
- Zhang, T., R. Ramakrishnan, and M. Livny (1996). Birch: an efficient data clustering method for very large databases. In *ACM Sigmod Record*, Volume 25, pp. 103–114. ACM.
- Zhong, C., D. Miao, and R. Wang (2010). A graph-theoretical clustering method based on two rounds of minimum spanning trees. *Pattern Recognition* 43(3), 752–766.
- Zhou, F., F. De la Torre, and J. K. Hodgins (2008). Aligned cluster analysis for temporal segmentation of human motion. In *Automatic Face & Gesture Recognition. 2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1–7. IEEE.