Modified Silhouette Score with Generalized Mean and Trimmed Mean

MODIFIED SILHOUETTE SCORE WITH GENERALIZED MEAN AND TRIMMED MEAN

BY

YIRAN ZHANG, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF MATHEMATICS & STATISTICS

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Master of Science (2023) McMaster University

(Mathematics & Statistics) Hamilton, Ontario, Canada

TITLE:	Modified Silhouette Score with Generalized Mean and Trimmed Mean
AUTHOR:	Yiran Zhang B.Sc., (Mathematics and Statistics) McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Paul D. McNicholas
NUMBER OF PAGES:	x, 67

To my parents, my partner, and my friends.

Abstract

The silhouette score is a widely used technique to evaluate the quality of a clustering result. One of the current issues with the silhouette score is its sensitivity to outliers, which can lead to misleading interpretations. This problem is caused by the silhouette score using the arithmetic mean to calculate the average intra and inter-cluster distances.

To address this issue, three modified silhouette scores are presented: GenSil, TrimSil, and extended TrimSil, which replace the arithmetic mean with the generalized mean, the trimmed mean and a modified trimmed mean, respectively. Experiments on both simulated and real-world datasets show that GenSil is the most effective method, significantly reducing the impact of outliers and achieving high silhouette scores with negative parameter values. TrimSil also improves silhouette scores but performs worse than GenSil, while the extended TrimSil outperforms TrimSil but is still less effective than GenSil. To further aid in selecting the optimal number of clusters with these modified silhouette scores, a more straightforward visualization technique, the silhouette-parameter plot, is also introduced.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor, Dr. Paul McNicholas, for his professional support and encouragement throughout the entire year. Thank you for offering me tons of suggestions on my research, as well as the opportunity to connect with other outstanding researchers, which has greatly enriched my experience. You truly are the best supervisor.

I would also like to thank Dr. Shui Feng for guiding me to choose the right academic path, and Dr. Pratheepa Jeganathan for her advice on careers in the field of data science. I also want to express my appreciation for having them on my exam committee.

I am also grateful to The Classification Society for awarding my poster presentation at their 2023 meeting. This award boosted my confidence in my research.

Lastly, I would like to thank my mom and dad for their unwavering love and endless support. Special thanks to Xingyu Liao and Ruoxi Yang for bringing me constant joy and keeping me away from stress. Your presence in my life is invaluable.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Clustering analysis is an unsupervised learning technique that assigns data points into groups based on their underlying patterns (Everitt *et al.*, 2011). Assessing the accuracy of the clustering result is challenging because of the absence of prior knowledge of the true labels. Various validation indices have been proposed to evaluate the quality of clustering results. Some commonly used methods are discussed in Liu *et al.* (2010).

This thesis focuses on the silhouette score. Rousseeuw (1987) introduced silhouette score as a clustering evaluation method that measures the goodness of fit of individual data points to their assigned clusters. The silhouette score can be displayed graphically, providing a more direct interpretation of the results. However, Kaufman and Rousseeuw (1990) state that the silhouette scores lack robustness when encountering outliers, and they imply that this issue is caused by the use of mean during calculation. The presence of outliers affects the inter and intra-cluster distances, resulting in a misleading silhouette score. To tackle this problem, we introduce three modified silhouette scores, which reduce the impact of outliers by implementing the

generalized mean, the trimmed mean, and an extended trimmed mean. A graphical method associated with these three modified silhouette scores is also proposed to simplify the selection process of the number of clusters.

Chapter 2 provides an introduction to the background related to these modified silhouette scores. The detailed modification and visualization technique is explained in Chapter 3. Chapter 4 demonstrates an extensive simulation study, and the performance of all methods on real datasets is shown in Chapter 5. Lastly, a summary of the work described in this thesis is presented in Chapter 6.

Chapter 2

Background

2.1 Clustering

Clustering is a study of partitioning data points into distinct groups based on their similarities without any knowledge of the true labels (McNicholas, 2016a). The primary goal of clustering is to group objects with high similarity into the same cluster, while keeping dissimilar objects in separate clusters. Similarities are typically measured by the distances between observations. The Euclidean distance is the most commonly used distance metric, some alternatives are discussed in Pandit and Gupta (2011).

The majority of clustering algorithms can be categorized into three main groups: hierarchical clustering, partitioning clustering, and model-based clustering (Rokach and Maimon, 2005). Hierarchical clustering groups data through either successive merges or successive divisions. In agglomerative hierarchical clustering, each data point is initially considered as its own cluster, and then the most similar clusters are progressively merged until they eventually form a single cluster. In contrast, divisive

hierarchical clustering starts with a single cluster, and then recursively separates the most dissimilar objects until each object is in its own cluster. Partitioning clustering begins with an initial assignment of the data points, and then iteratively adjusts the membership of each point until reaches an optimization. K-means is an example of such clustering that minimizes the variance within each cluster. Model-based clustering relies more on statistical methods compared to the other two types. It is built on the assumption that the data points are generated by a finite number of mixture models, and the parameters of these models need to be estimated. While the mixture models can be based on any distribution, the Gaussian distribution is often preferred in model-based clustering.

Since clustering is an unsupervised learning technique that operates on datasets without any explicit labeling, computing a prediction accuracy is not applicable. Instead, internal validation methods are used to evaluate the cohesion and separation of clusters (Tan *et al.*, 2019). The silhouette score is a typical example of such internal validation, which measures the goodness of fit using the distances within each cluster and between clusters.

2.2 Silhouette Score

The silhouette score is a visualization-based evaluation technique. Suppose for an observation i, the cluster that i is assigned to is labeled as a, and its nearest neighbour cluster is b, then its individual silhouette score $s(i)$ is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}, \tag{2.1}$$

where $a(i)$ denotes the average intra-cluster distance of observation i, and its average inter-cluster distance is represented by $b(i)$.

Silhouette scores take a minimum value of -1 and a maximum of 1. A silhouette score of 1 indicates a perfect clustering, 0 shows that the observation is at the boundary of two clusters, and -1 means a complete misclassification. The overall silhouette score, which is the average of all individual silhouette scores, can be used to assess the quality of a clustering result. Kaufman and Rousseeuw (1990) recommend an interpretation for the silhouette scores: a score above 0.70 is considered strong, between 0.51 and 0.70 is reasonable, between 0.26 and 0.50 is relatively weak, and anything below 0.25 is viewed as incorrect.

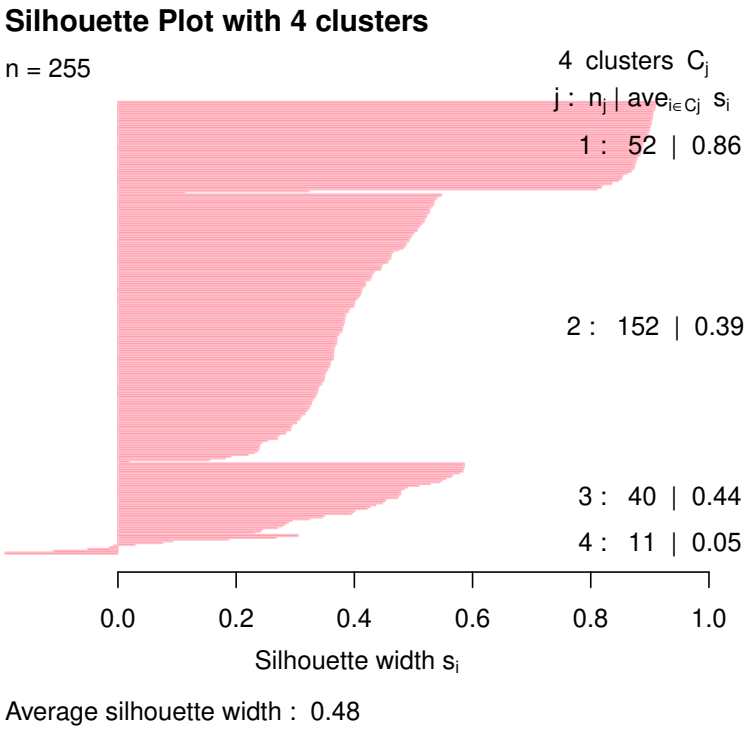

Figure 2.1: An example of silhouette plot for a clustering result with 4 clusters.

One of the most important features of the silhouette scores is their graphical representation. Figure 2.1 provides an example of a silhouette plot for a clustering result with 4 clusters. The silhouette plot consists of several groups of horizontal bars, where the width of each bar represents the silhouette score of the corresponding observation. Bars with negative silhouette scores are positioned in the opposite direction of the positive ones, as shown in the lower part of Figure 2.1, indicating potential misclassifications. The height of each group reflects the number of observations assigned to that particular cluster, and the labels on the right-hand side show the average silhouette score for each cluster. Additionally, the overall silhouette score is displayed at the bottom.

Silhouettes can be helpful in determining the optimal number of clusters. The number of clusters that achieves the highest overall silhouette score and produces the most visually natural silhouette plot is the optimal choice. Some examples are provided in Rousseeuw (1987), Shahapure and Nicholas (2020), and Dudek (2020).

In addition to its direct applications, the silhouette score has been explored for integration into clustering algorithms to enhance their performance. Van der Laan *et al.* (2003) propose two clustering algorithms, PAMSIL and PAMMEDSIL, which replace the loss function in partition around medoids clustering with silhouette score and simplified silhouette score, to identify small clusters effectively. Shutaywi and Kachouie (2021) use silhouette scores as a weighting function to aggregate multiple kernel k-means clustering results, aiming to achieve a less biased output. Batool and Hennig (2021) introduce two clustering techniques, OSil and FOSil, that identify clusters by maximizing the average silhouette score.

Despite the utility of the silhouette score in many aspects, it is affected by outliers.

The calculation of the silhouette score relies on the average intra and inter-cluster distances, $a(i)$ and $b(i)$. The existence of an outlier can increase the average distances and impact the silhouette score.

For simplicity, assume that $\max(a(i), b(i)) = b(i)$ in the silhouette score. Suppose that an outlier is in the same cluster as observation i, then the value of $a(i)$ rises to $a'(i)$, resulting in a lower silhouette score as shown below:

$$a(i) < a'(i) \quad \rightarrow \quad b(i) - a(i) > b(i) - a'(i) \quad \rightarrow \quad \frac{b(i) - a(i)}{b(i)} > \frac{b(i) - a'(i)}{b(i)}.$$

If the outlier exists in the nearest cluster of observation i, its impact on the silhouette score may differ depending on its specific location. Regardless of its position, the overall effect remains undesired.

Therefore, the presence of outliers deviates the silhouette score from its natural value, which can bring a negative impact on the clustering results.

2.3 K-Means Clustering

K-means (MacQueen, 1967; Lloyd, 1982) is a widely used clustering method. For a predefined number of clusters k, the k-means algorithm chooses k centroids randomly, calculates the distance between each data point and each centroid, and assigns them to the cluster associated with their nearest centroid. After the initial clustering, k-means updates the centroids to the new within-cluster mean, recalculates the distances, and reassigns the data points iteratively, until no further changes in the cluster assignments occur. In other words, k-means finds the clustering by minimizing the within-cluster variation.

K-means clustering is often chosen due to its computational efficiency and interpretability (Namratha and Prajwala, 2012). It has demonstrated successful applications across various fields (Govender and Sivakumar, 2020; Wang *et al.*, 2017). Nevertheless, it has been found that k-means tend to find spherical clusters with equal size (de Craen *et al.*, 2006).

In fact, k-means is equivalent to the classification expectation-maximization algorithm on a particular Gaussian mixture model, where the mixing proportion π_g is equal for all components, and the covariance matrix takes the form of $\boldsymbol{\Sigma}_g = \lambda \mathbf{I}$, with λ being a positive constant and $\mathbf{I}$ being an identity matrix (Celeux and Govaert, 1992). More details on the Gaussian mixture model are discussed in the next section.

Several studies have explored the extensions of k-means clustering to adapt various types of datasets. One example is partition around medoids or k-medoids clustering (Kaufman and Rousseeuw, 1987), which replaces the centroids with medoids to enhance robustness. Another example is the k-modes algorithm introduced by Huang (1997) as an extension of k-means for the categorical data. More modified k-means algorithms are discussed in Shukla and Naganna (2014).

2.4 Gaussian Mixture Models

The finite Gaussian mixture model is another powerful clustering technique that relies on the assumption of Gaussian distributions, and each cluster is considered as a component. Its density is defined as:

$$f(\mathbf{x} \mid \boldsymbol{\vartheta}) = \sum_{g=1}^{G} \pi_g \phi(\mathbf{x} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g), \tag{2.2}$$

where G is the total number of components and $\boldsymbol{\vartheta}$ is a vector of all parameters. π_g is the mixing proportion, representing the probability of the observation coming from component g, therefore it follows that $\pi_g > 0$ and $\sum_{g=1}^{G} \pi_g = 1$. The $\phi(\mathbf{x}|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ is the density function of component g, with a mean vector of $\boldsymbol{\mu}_g$ and a covariance matrix of $\boldsymbol{\Sigma}_g$ under a Gaussian distribution assumption.

Compared with k-means clustering, the Gaussian mixture model is more capable of identifying clusters of various shapes and sizes (Grün, 2019), therefore it captures more complex patterns in the data. An example of how Gaussian mixture models outperforms k-means in real datasets is Amruthnath and Gupta (2018).

The family of 14 Gaussian parsimonious clustering models (GPCMs; Banfield and Raftery, 1993; Celeux and Govaert, 1995) is introduced which puts constraints on the volume, shape, and orientation of the component covariance matrix, to fit different clustering situations. The component covariance matrix is expressed in the form of eigenvalue decomposition

$$\boldsymbol{\Sigma}_g = \lambda_g \mathbf{D}_g \mathbf{A}_g \mathbf{D}_g', \tag{2.3}$$

where λ_g is a constant, $\mathbf{D}_g$ is a matrix of the eigenvectors of $\boldsymbol{\Sigma}_g$, and $\mathbf{A}_g$ is a diagonal matrix with entries proportional to the eigenvalues of $\boldsymbol{\Sigma}_g$. Table 2.1 is the list of 14 models from the GPCM family.

Table 2.1: Models of the GPCM family.

Model	Volume	Shape	Orientation	$\boldsymbol{\Sigma}_g$
EII	Equal	Spherical	/	$\lambda\mathbf{I}$
VII	Variable	Spherical	/	$\lambda_g\mathbf{I}$
EEI	Equal	Equal	Axis-Aligned	$\lambda\mathbf{A}$
VEI	Variable	Equal	Axis-Aligned	$\lambda_g\mathbf{A}$
EVI	Equal	Variable	Axis-Aligned	$\lambda\mathbf{A}_g$
VVI	Variable	Variable	Axis-Aligned	$\lambda_g\mathbf{A}_g$
EEE	Equal	Equal	Equal	$\lambda\mathbf{DAD}'$
VEE	Variable	Equal	Equal	$\lambda_g\mathbf{DAD}'$
EVE	Equal	Variable	Equal	$\lambda\mathbf{DA}_g\mathbf{D}'$
EEV	Equal	Equal	Variable	$\lambda\mathbf{D}_g\mathbf{AD}_g'$
VVE	Variable	Variable	Equal	$\lambda_g\mathbf{DA}_g\mathbf{D}'$
VEV	Variable	Equal	Variable	$\lambda_g\mathbf{D}_g\mathbf{AD}_g'$
EVV	Equal	Variable	Variable	$\lambda\mathbf{D}_g\mathbf{A}_g\mathbf{D}_g'$
VVV	Variable	Variable	Variable	$\lambda_g\mathbf{D}_g\mathbf{A}_g\mathbf{D}_g'$

When choosing the most appropriate model from GPCMs, each model from the family is run with multiple values of G. The best model is selected via the Bayesian information criterion, also known as the BIC (Schwarz, 1978), calculated by

$$\text{BIC} = 2l(\hat{\boldsymbol{\vartheta}}) - \rho \log n, \tag{2.4}$$

where $\hat{\boldsymbol{\vartheta}}$ is the maximum likelihood estimate of $\boldsymbol{\vartheta}$, $l(\hat{\boldsymbol{\vartheta}})$ is the maximized log-likelihood, ρ is the number of free parameters, and n is the number of observations. The model

with the largest BIC is selected as the best model. The parameters of the model are usually estimated through the EM algorithm (Dempster *et al.*, 1977).

The GPCMs are found to be an effective technique for clustering, examples are the parsimonious generalized linear Gaussian cluster-weighted models proposed by Punzo and Ingrassia (2015) and the clustMD method for clustering mixed-type data presented by McParland and Gormley (2016). A detailed review of the GPCMs is included in McNicholas (2016b) and Gormley *et al.* (2023).

The GPCM family of models is implemented in many packages in R (R Core Team, 2023), this thesis will use the `mixture` package (Pocuca *et al.*, 2022).

2.5 Some Verification Indices

The accuracy and the adjusted Rand index (ARI) are used as references to verify the performance of the three modified silhouette scores in Chapter 5.

2.5.1 Misclassification Rate and Accuracy

The misclassification rate is often seen in classification problems where the true labels are given. For a dataset with n observations, the misclassification rate is calculated by

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i), \tag{2.5}$$

where the y_i and $\hat{y}_i$ are the true and predicted label for each observation, and $I(y_i \neq \hat{y}_i)$ is an indicator function that equals to 1 if $y_i \neq \hat{y}_i$, 0 otherwise (James *et al.*, 2013).

$1 -$ misclassification rate represents the accuracy. Alternatively, the accuracy can be calculated through a confusion matrix, where the diagonal elements represent the

number of observations correctly classified, and the off-diagonal elements represent the misclassified observations. Dividing the sum of diagonals by the total number of observations gives the accuracy. It is typically a percentage between 0 and 1, with higher values indicating higher accuracy.

2.5.2 Adjusted Rand Index

The Rand index (RI), introduced by Rand (1971), measures the similarity between two partitions by the proportion of pairwise agreements over the total amount of pairs. It can be expressed as:

$$\text{RI} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total number of pairs}}. \tag{2.6}$$

Here the True Positive represents the number of pairs that are assigned to the same cluster in both partitions, while the True Negative indicates the number of pairs that are assigned to different clusters in both partitions. The RI value ranges between 0 and 1, with a value of 1 implying perfect agreement between the two partitions.

The adjusted Rand index (ARI) is a modification of the Rand index that accounts for agreements by chance (Hubert and Arabie, 1985). Its general formula is given by:

$$\text{ARI} = \frac{\text{RI} - \text{Expected RI}}{\text{Maximum RI} - \text{Expected RI}}. \tag{2.7}$$

The ARI does not have a lower bound, but it is more desirable to have a positive value. The upper bound of the ARI is 1, indicating a perfect agreement between two partitions. The closer the ARI value is to 1, the more similar the two partitions are. The expected value of ARI equals 0 under random classification.

2.6 Generalized Mean

There are various methods of calculating the mean, such as the arithmetic mean and the quadratic mean. These means can be generalized into one formula, known as the generalized mean (Hardy *et al.*, 1952). For a set of non-negative numbers $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ and a parameter p, the generalized mean of this set is defined as

$$\mu_p(\mathbf{x}) = \left(\frac{1}{n} \sum_{j=1}^{n} x_j^p \right)^{\frac{1}{p}}. \tag{2.8}$$

It is a comprehensive measure of the mean, as other definitions of the mean can be achieved by adjusting the parameter p from $-\infty$ to ∞. Some special values for p include:

1. $p = -\infty$, then $\mu_{-\infty}(\mathbf{x}) = \min(x_1, \ldots, x_n)$, the minimum of $\mathbf{x}$.
2. $p = -1$, then $\mu_{-1}(\mathbf{x}) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{x_j}$, the harmonic mean.
3. $p = 0$, then $\mu_0(\mathbf{x})$ is defined as the geometric mean $\left(\prod_{j=1}^{n} x_j \right)^{\frac{1}{n}}$.
4. $p = 1$, then $\mu_1(\mathbf{x})$ is equivalent to the arithmetic mean $\frac{1}{n} \sum_{j=1}^{n} x_j$.
5. $p = \infty$, then $\mu_{\infty}(\mathbf{x}) = \max(x_1, \ldots, x_n)$, the maximum of $\mathbf{x}$.

Some important properties of the generalized mean are listed in Hardy *et al.* (1952). Among these properties, the following is the most useful for this thesis, and its proof is included in Appendix A.

Property 1. *For non-negative* $\mathbf{x}$ *and any real valued* p *and* q*, if* $p > q$*, then* $\mu_p(\mathbf{x}) \geq \mu_q(\mathbf{x})$*, with the equality holds only when the* $\mathbf{x}$ *are equal.*

This property directly implies that the generalized mean value decreases as the parameter p decreases.

The generalized mean is often considered as the alternative for the arithmetic mean to increase robustness against outliers. For example, Luukka and Leppälampi (2006) modify a fuzzy similarity-based classification technique with the generalized mean and successfully boost the prediction accuracy in some medical examples. Oh and Kwak (2016) implement the generalized mean in the principal component analysis and achieve a higher classification accuracy in image processing. Gou *et al.* (2019) replace the arithmetic mean with the generalized mean in the distance-weighted k-nearest neighbors, to avoid outliers dominating the classification result.

2.7 Trimmed Mean

The trimmed mean is a commonly used method that reduces the influence of extreme values when calculating the average. It removes a certain percentage of the highest and the lowest values from a set of numbers and computes the standard arithmetic mean of the remaining observations. The trimmed mean focuses more on the central observations, therefore is more robust to outliers (Fraiman and Muniz, 2001).

The trimming percentage, denoted as t, is the only parameter of the trimmed mean, which ranges from 0 to 0.5. When $t = 0$, no observation is removed, and hence the trimmed mean is equivalent to the original arithmetic mean. The trimmed mean is the median of the number set if $t = 0.5$. The floor value is taken if the number of objects to trim is not an integer.

A mathematical representation of the trimmed mean is as follows: for a set of

ordered real numbers $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, the trimmed mean is

$$\mu_t(\mathbf{x}) = \frac{1}{n - 2\lfloor tn \rfloor} \sum_{j=\lfloor tn \rfloor}^{n-\lfloor tn \rfloor} x_j. \tag{2.9}$$

Chapter 3

Methodology

3.1 Silhouette Score with Generalized Mean

The implementation of the generalized mean in the silhouette score is initially introduced by Lengyel and Botta-Dukát (2019), where they adopt the generalized mean to assist non-spherical clusters in achieving higher silhouette scores. In our study, the generalized mean is used to reduce the impact of outliers. There are no restrictions on which clustering algorithm should be associated with the silhouette score, this thesis focuses on k-means and GPCM clustering. Euclidean distance is used for all the distance metrics.

Algorithm 1 describes the calculation process of the modified silhouette score with generalized mean (GenSil). GenSil follows the same steps as the original silhouette score, with the difference that the arithmetic mean is substituted by the generalized mean when computing the average within-cluster distance $a(i)$ and the average between-cluster distance $b(i)$.

Algorithm 1 GenSil

Obtain a clustering result from k-means or GPCM.

for each observation i **do**

1. **Calculate a(i)**:

Compute the distance between i and each observation within its same cluster, and store these distances in a vector $\mathbf{x}$.

Calculate $a(i) = \mu_p(\mathbf{x}) = \left(\frac{1}{n}\sum_{j=1}^{n} x_j^p\right)^{\frac{1}{p}}$

2. **Calculate b(i)**:

Compute the distance between i and each observation in the nearest cluster to i, and store these distances in a vector $\mathbf{y}$.

Calculate $b(i) = \mu_p(\mathbf{y}) = \left(\frac{1}{m}\sum_{j=1}^{m} y_j^p\right)^{\frac{1}{p}}$

3. **Return** the individual GenSil score $s(i) = \frac{b(i)-a(i)}{\max(b(i),a(i))}$

end for

Calculate the overall GenSil score $\bar{s} = \frac{1}{n}\sum_{i=1}^{n} s(i)$.

According to Lengyel and Botta-Dukát (2019), the generalized mean with a low parameter value places greater emphasis on neighboring objects and reduces the impact from objects further away. Therefore, by choosing a small parameter value p, GenSil minimizes the influence of outliers and highlights the importance of nearby observations.

It is worth noting that as the parameter p approaches negative infinity, the generalized mean converges to the minimum value, which results in considering only the closest neighbor and disregarding information from other observations. Thus, there exists a trade-off between eliminating the impact of outlying values and considering the overall dataset when choosing an appropriate value for the parameter p.

3.2 Silhouette Score with Trimmed Mean

The silhouette score with trimmed mean (TrimSil) is another approach to enhance the robustness of the silhouette score. The implementation of the trimmed mean in the silhouette score follows a similar procedure described in Algorithm 1, but the mean functions in step 1 and step 2 are the trimmed mean, i.e., for each observation i,

$$a(i) = \mu_t(\mathbf{x}) = \frac{1}{n - 2\lfloor tn \rfloor} \sum_{j=\lfloor tn \rfloor}^{n-\lfloor tn \rfloor} x_j,$$
$$b(i) = \mu_t(\mathbf{y}) = \frac{1}{m - 2\lfloor tm \rfloor} \sum_{j=\lfloor tm \rfloor}^{n-\lfloor tm \rfloor} y_j.$$

As trimming percentage t gradually increases from 0 towards 0.5, the influence of large distances, which are primarily caused by outliers, is diminished. By emphasizing the central distance and disregarding the extreme values, the TrimSil score improves the original silhouette score. The reduction in the impact of outliers makes the TrimSil score a more stable approach compared to the original silhouette score.

A concern regarding the concept behind TrimSil is that, while it trims off the largest distances, it also removes an equal amount of the smallest distances due to the nature of a trimmed mean. As a result, the TrimSil score is computed based on observations within a moderate range of distances, which may not fully capture the importance of close-distance observations in the context of clustering. Since observations in close proximity often hold more relevance for identifying distinct clusters, TrimSil may only result in minor improvements in the silhouette score, or in some cases, even lead to an inferior score.

To address this issue, an extended version of TrimSil is introduced.

3.2.1 Extension of TrimSil

The extended version of TrimSil introduces a modified trimmed mean. Instead of trimming off both the highest and lowest values, the modified trimmed mean only removes the largest values and retains the small values. A mathematical notation of this modified trimmed mean is

$$\mu_t(\mathbf{x}) = \frac{1}{n - \lfloor tn \rfloor} \sum_{j=1}^{n-\lfloor tn \rfloor} x_j, \tag{3.1}$$

where $\mathbf{x}$ is a set of real numbers in ascending order, n is the total number of observations and t is the proportion of the largest values to be removed. The extended TrimSil effectively integrates the modified trimmed mean in the computation of $a(i)$ and $b(i)$. With this modified trimmed mean, the extended TrimSil is expected to achieve a better balance between reducing the impact of outliers and retaining important information from closely located observations.

Upon this modification in the trimmed mean, the lower bound of the parameter t remains 0, but the upper bound is extended from 0.5 to 1. When $t = 0$, the modified trimmed mean is the same as the arithmetic mean. When $t = 1$, the modified trimmed mean equals the smallest value in the number set.

The level of improvement obtained by the extended TrimSil is anticipated to be greater compared to TrimSil. The inclusion of small distances in the calculation takes the information from closely located observations into account, leading to a more accurate evaluation of the clustering result. Therefore, the extended TrimSil is

considered more reasonable than TrimSil for clustering assessment.

3.3 Parameter Selection

All of GenSil, TrimSil, and extended TrimSil require a pre-defined parameter. GenSil requires a value for p within the range of $-\infty$ to ∞. Similarly, TrimSil requires a trimming percentage t, which is a non-negative number that is less than or equal to 0.5, or 1 if using extended TrimSil.

Based on the definition of the generalized mean in the GenSil context, as p approaches negative infinity, the GenSil score continues increasing. Looking for the p that returns the highest GenSil score is not reasonable, because it will always be the negative infinity. The technique being used to determine the parameter value for GenSil in this thesis is that, run GenSil over a range of candidate values for p, plot the p against its corresponding GenSil score, and select the most appropriate one according to the graph. In practice, it is observed that this plot demonstrates a curved shape that looks like a logarithmic growth, or a reversed elbow shape, shown as an example in Figure 3.1. Initially, the GenSil score increases rapidly for the first few values of p, then tends to level out as p continues decreasing. This pattern indicates that the modified silhouette score rises significantly for the first few p, but the changes become negligible after p reaches a certain value. The p at the point where the line begins to level out, in other words, the elbow point, is selected as the parameter value for GenSil.

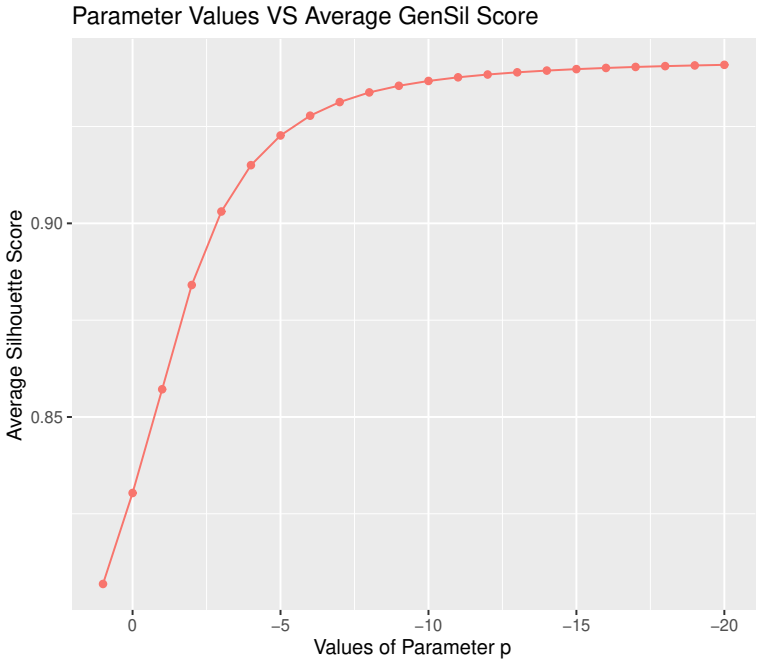

Figure 3.1: Example of the pattern of GenSil scores against different parameter values.

Selecting the trimming percentage for TrimSil and extended TrimSil is slightly more challenging. In theory, the parameters of TrimSil and extended TrimSil can be chosen following the same process as GenSil: plot the parameter values against the silhouette score and select the value at the elbow point. However, in practice, the reversed elbow shape is not always observed in the plot.

In some cases, the graph exhibits a linear trend, and hence the elbow point is no longer applicable. If this is the case, some external techniques need to be used to estimate the proportion of outliers in the dataset. For instance, the `findGrossOuts` function from the `oclust` package (Clark and McNicholas, 2022) finds the number of outliers in the dataset through DBSCAN, a scatter plot can visually identify potential outliers, and a boxplot can find data points beyond the whiskers. The proportion of outliers obtained from the data exploratory analysis is then chosen as the parameters

for TrimSil and extended TrimSil.

3.4 Silhouette-Parameter Plot

The traditional silhouette plot appears wider for the most appropriate number of clusters, k. The traditional visual approach to select optimal k involves generating multiple clustering results with different k, drawing each of their silhouette plots, and then choosing the one that appears the widest. This process is not complicated, however, it is inconvenient as it requires running the algorithms multiple times and comparing multiple plots, and the decision on the widest plot can also be subjective.

To overcome this issue, a new visualization technique, the silhouette-parameter plot, is introduced to facilitate selecting the optimal k. This plot needs to work with GenSil, TrimSil or extended TrimSil. The resulting graph is a collection of line graphs, where the modified silhouette scores are on the vertical axis, and the corresponding parameter values are on the horizontal axis, therefore it is called the silhouette-parameter plot. This graphical approach offers insights into how the modified silhouette scores vary in response to changes in the parameter values with different k.

Algorithm 2 explains the process of this visualization. The concept is to plot multiple line graphs over a large range of parameters for each k and display them on a single graph. The k value that corresponds to the highest line is then selected as the optimal choice. This approach provides a more efficient visualization for determining the optimal k.

Algorithm 2 Finding the optimal number of clusters

1. List a range of the number of clusters, k.

2. List a range of parameters: p if using GenSil, t if using TrimSil or extended TrimSil.

for k in the list from step 1 **do**

Calculate the overall modified silhouette score using each parameter in the list from step 2.

end for

Plot the overall modified silhouette score against the parameter.

Choose the k that returns the top line in the plot.

Figure 3.2 demonstrates an example of the output graph using GenSil. The horizontal axis is the p values ranging from 1 to -20, and the vertical axis is the average overall silhouette score achieved by GenSil using each p. In this graph, different colored lines represent different numbers of clusters. The line representing $k = 4$ locates at the top, indicating that grouping the data into 4 clusters returns the highest overall silhouette score. Similarly, the line for 6 clusters is the lowest in the plot, meaning a low overall silhouette score. Lines that are higher in the graph are preferred, therefore in this example, 4 is suggested as the optimal number of clusters. Additionally, observe that the line illustrates the reversed elbow shape as mentioned in Section 3.3. The silhouette score increases dramatically for the first few p values, then the following growth becomes relatively negligible. In this particular example, the value of $p = -5$, indicated by the vertical red line, is chosen to compute the GenSil score as it is at the elbow point where the growth rate levels out.

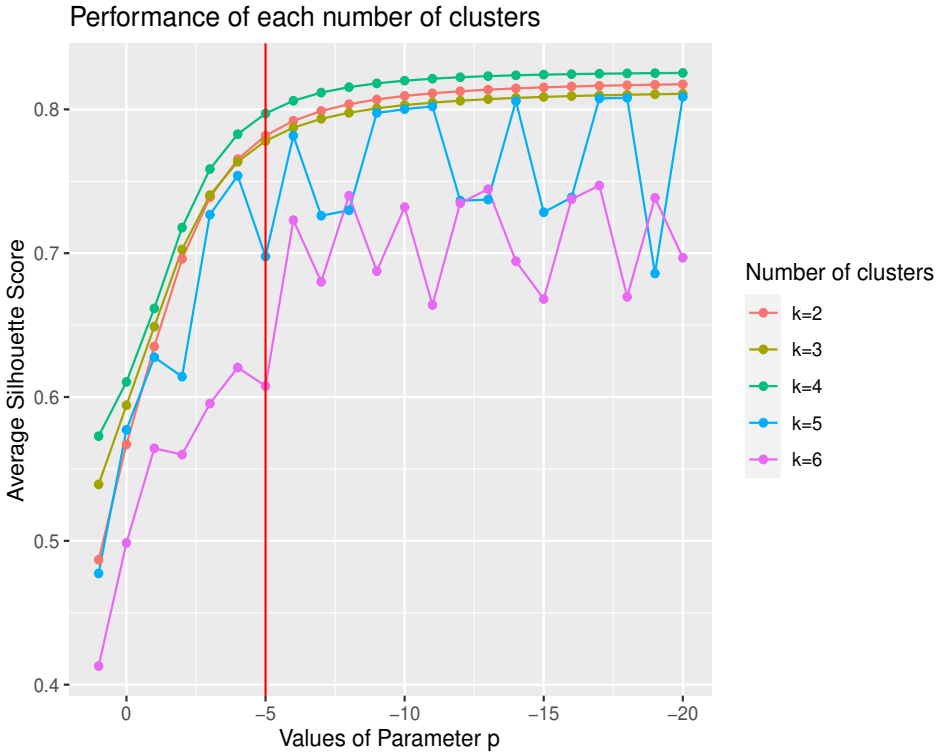

Figure 3.2: Example of selecting the optimal number of clusters and optimal parameter value using GenSil.

The silhouette-parameter plot effectively converts the silhouette plot into a line graph, allowing for the representation of multiple line graphs in a single plot. In contrast to the traditional silhouette plots, this new visualization is more straightforward and convenient because line graphs are easier to interpret and provide more objective insights. Furthermore, the silhouette-parameter plot can suggest a suitable parameter value, making the process of calculating the modified silhouette score more efficient.

Chapter 4

Simulation

4.1 Data Setting

To explore the factors influencing the performance of GenSil, TrimSil, and extended TrimSil, extensive clustering scenarios are simulated. For simplicity, two clusters are generated from two separate Gaussian distributions, consisting of a total of 100 observations. The mean, variance, and number of observations are manipulated to control the separation, compactness, and size of each Gaussian cluster. Furthermore, 5 extreme outliers are artificially added to each simulated dataset to evaluate the performance of each method with and without outliers. Figures 4.1, 4.2, and 4.3 provide visual representations of the simulated datasets. The red and green points form two clusters, and the grey points are the outliers.

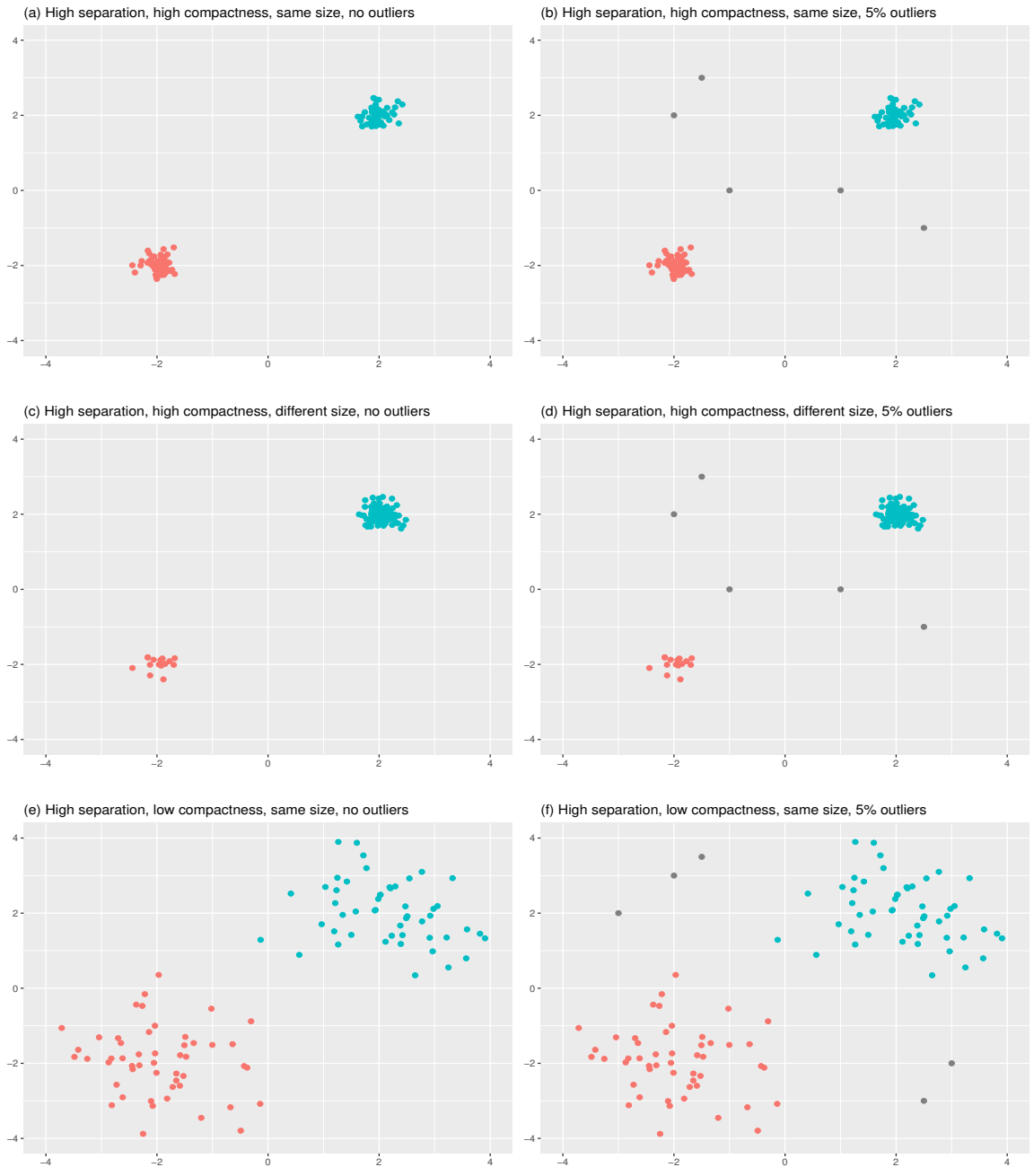

Figure 4.1: Simulation data settings (part 1).

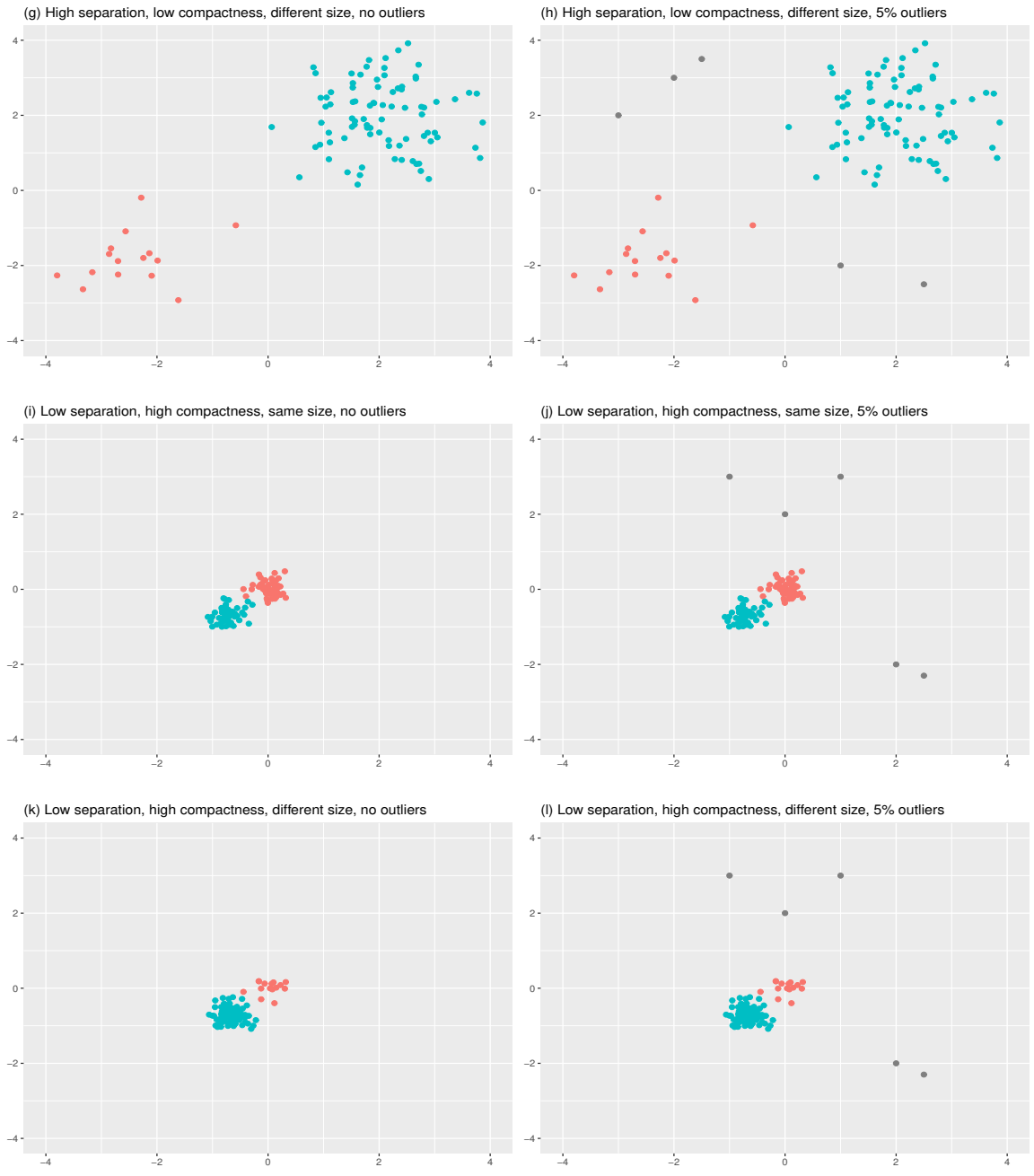

Figure 4.2: Simulation data settings (part 2).

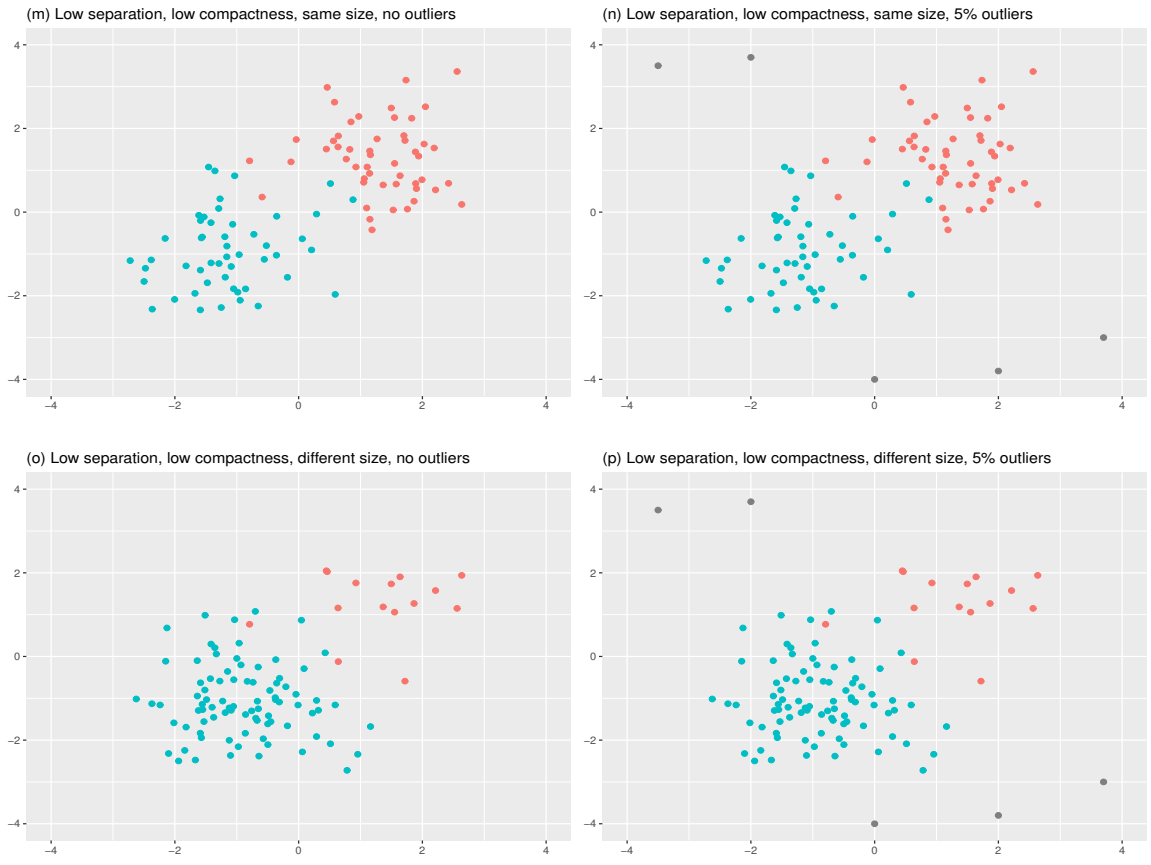

Figure 4.3: Simulation data settings (part 3).

For each simulated dataset, the performance of GenSil, TrimSil, and extended TrimSil is evaluated. The parameter for GenSil uses the elbow point value described in Section 3.3. For TrimSil and extended TrimSil, the trimming percentages are fixed at 5%, which is the proportion of outliers in the data. For consistency, 500 iterations are executed for each method, using both GPCM from the `mixture` package and k-means clustering from `stats` package in R, with a fixed number of clusters set to 2. The performance of each method is evaluated by their relative improvement, which is calculated as follows:

$$\frac{\text{new silhouette score} - \text{original silhouette score}}{\text{original silhouette score}} \times 100\%. \tag{4.1}$$

A positive relative improvement implies that the approach produces a higher silhouette score than the original, whereas a negative number suggests that the modified silhouette score is lower than the original silhouette score.

4.2 Results

4.2.1 Results of GenSil

Tables 4.1 and 4.2 present the results of GenSil in each cluster scenario under k-means and GPCM, respectively. The clustering scenarios, recorded in the first column, are described in the order of "Separation - Compactness - Cluster Size - Outliers".

GenSil demonstrates a higher relative improvement on clusters with low separation, low compactness, and a greater number of outliers. The cluster size, on the other hand, does not appear to have a significant impact on its performance.

In scenarios without outliers or with highly compact and well-separated clusters, GenSil performs similarly under both k-means and GPCM. However, when the observations within each cluster are dispersed and the clusters are close to each other, GenSil outperforms k-means more than GPCM.

Most of the relative improvements are positive, indicating that the GenSil score is higher than the original silhouette score. Only a single case returns a lower silhouette score when using k-means. This phenomenon is particularly observed when the

clusters are closely located, highly compact, of different sizes, and contain more outliers. Such a phenomenon is not observed when using GPCM. The current possible explanation for this is that the clustering algorithm wrongly assigns the outliers to their own cluster.

Nevertheless, GenSil proves to be a better alternative than the original silhouette score due to its ability to produce higher scores when having outliers in the dataset.

Table 4.1: Relative improvement of GenSil using k-means from 500 iterations.

Scenario	+ve output	-ve output	Relative Improvement
High - High - Same - No	500	0	4.37%
High - High - Same - Yes	500	0	7.17%
High - High - Different - No	500	0	4.51%
High - High - Different - Yes	500	0	6.02%
High - Low - same - No	500	0	24.3%
High - Low - Same - Yes	500	0	27.7%
High - Low - Different - No	500	0	25.3%
High - Low - Different - Yes	500	0	28.7%
Low - High - Same - No	500	0	32.1%
Low - High - Same - Yes	500	0	53.1%
Low - High - Different - No	500	0	33.7%
Low - High - Different - Yes	499	1	21.6%
Low - Low - Same - No	500	0	48.2%
Low - Low - Same - Yes	500	0	56.4%
Low - Low - Different - No	500	0	57.2%
Low - Low - Different - Yes	500	0	64.3%

Table 4.2: Relative improvement of GenSil using GPCM from 500 iterations.

Scenario	+ve output	-ve output	Relative Improvement
High - High - Same - No	500	0	4.39%
High - High - Same - Yes	500	0	7.62%
High - High - Different - No	500	0	4.52%
High - High - Different - Yes	500	0	8.87%
High - Low - same - No	500	0	24.3%
High - Low - Same - Yes	500	0	28.3%
High - Low - Different - No	500	0	25.3%
High - Low - Different - Yes	500	0	31.1%
Low - High - Same - No	500	0	32.4%
Low - High - Same - Yes	500	0	28.9%
Low - High - Different - No	500	0	34.5%
Low - High - Different - Yes	500	0	30.7%
Low - Low - Same - No	500	0	48.6%
Low - Low - Same - Yes	500	0	70.0%
Low - Low - Different - No	500	0	54.8%
Low - Low - Different - Yes	500	0	62.6%

4.2.2 Results of TrimSil

Tables 4.3 and 4.4 exhibit the relative improvements using TrimSil under k-means and GPCM, respectively.

Table 4.3: Relative improvement of TrimSil using k-means from 500 iterations.

Scenario	+ve output	-ve output	Relative Improvement
High - High - Same - No	500	0	0.0695%
High - High - Same - Yes	500	0	1.60%
High - High - Different - No	500	0	0.0681%
High - High - Different - Yes	500	0	1.42%
High - Low - same - No	500	0	0.396%
High - Low - Same - Yes	500	0	1.16%
High - Low - Different - No	500	0	0.390%
High - Low - Different - Yes	500	0	0.971%
Low - High - Same - No	500	0	0.494%
Low - High - Same - Yes	497	3	6.52%
Low - High - Different - No	500	0	0.516%
Low - High - Different - Yes	401	99	3.37%
Low - Low - Same - No	494	6	0.526%
Low - Low - Same - Yes	500	0	3.28%
Low - Low - Different - No	414	86	0.447%
Low - Low - Different - Yes	498	2	2.67%

Table 4.4: Relative improvement of TrimSil using GPCM from 500 iterations.

Scenario	+ve output	-ve output	Relative Improvement
High - High - Same - No	500	0	0.0685%
High - High - Same - Yes	500	0	1.12%
High - High - Different - No	500	0	0.0662%
High - High - Different - Yes	500	0	0.209%
High - Low - same - No	499	1	0.388%
High - Low - Same - Yes	500	0	1.13%
High - Low - Different - No	500	0	0.387%
High - Low - Different - Yes	498	2	0.778%
Low - High - Same - No	500	0	0.485%
Low - High - Same - Yes	442	58	1.18%
Low - High - Different - No	500	0	0.533%
Low - High - Different - Yes	266	234	-0.311%
Low - Low - Same - No	491	9	0.525%
Low - Low - Same - Yes	495	5	2.97%
Low - Low - Different - No	492	8	0.793%
Low - Low - Different - Yes	477	23	0.578%

The relative improvement of TrimSil is higher when the clusters have low separation and more outliers. When the clusters are well separated, TrimSil is found to have better performance on those with low compactness; but when the clusters have low separation, it performs better in those with high compactness. Similar to GenSil, the size of the clusters has little impact on TrimSil's performance.

TrimSil has the same level of performance in both k-means and GPCM when there are no outliers, or the clusters are highly compact and well separated. However, when the observations in each cluster are dispersed and the clusters are close, TrimSil's performance in GPCM is much worse than in k-means.

The amount of improvement achieved by TrimSil is relatively small compared to the ones from GenSil. Furthermore, TrimSil does not guarantee an improvement when the clusters have low separation. When using k-means, 5 cluster scenarios show negative relative improvements. Among these 5 scenarios, 2 of them (Low-High-Different-Yes and Low-Low-Different-No) have a great number of negative values (99 and 86 out of 500 respectively). The results obtained from GPCM are even worse. Negative outputs are observed in 8 scenarios. Particularly in the scenario with low separation, high compactness, different cluster sizes, and more outliers, half of the iterations produce a worse silhouette score than the original.

A plausible explanation for this phenomenon, as discussed in Section 3.2, is that the trimmed mean removes both the large and small distances, leading to a more significant impact, particularly in clusters with low separation and low compactness. Another potential cause is that the clustering algorithm being used does not accurately identify the optimal cluster assignments.

Therefore, it can be concluded that TrimSil is not an effective approach to raise the silhouette score, because it either results in a minor improvement or a lower score.

In addition to the tests above, the function `findGrossOuts` from the `oclust` package is used to assess the number of outliers within each simulated dataset. The trimming percentage is then chosen as the estimated proportion of outliers, and the

resulting relative improvement is computed. Table 4.5 illustrates the amount of relative improvement achieved using the t recommended by `oclust` for both k-means and GPCM. We observe that the percentages of outliers estimated by `oclust` align closely with the actual values, therefore it proves to be a reliable method to choose the trimming percentage.

Table 4.5: Relative improvement obtained by TrimSil using k-means and GPCM where t is estimated using oclust.

Scenario	% outliers estimated	k-means	GPCM
High - High - Same - No	0%	0%	0%
High - High - Same - Yes	3%	0.929%	0.562%
High - High - Different - No	0%	0%	0%
High - High - Different - Yes	3%	1.06%	0.0506%
High - Low - same - No	0%	0%	0%
High - Low - Same - Yes	3%	0.670%	0.670%
High - Low - Different - No	0%	0%	0%
High - Low - Different - Yes	4%	0.773%	0.773%
Low - High - Same - No	1%	0%	0%
Low - High - Same - Yes	5%	5.95%	1.13%
Low - High - Different - No	0%	0%	0%
Low - High - Different - Yes	5%	1.84%	0.780%
Low - Low - Same - No	0%	0%	0%
Low - Low - Same - Yes	5%	3.08%	3.22%
Low - Low - Different - No	3%	0.409%	0.390%
Low - Low - Different - Yes	4%	2.93%	0.384%

4.2.3 Results of Extended TrimSil

The relative improvement created by the extended TrimSil using k-means and GPCM is shown in Table 4.6 and 4.7.

Table 4.6: Relative improvement of extended TrimSil using k-means from 500 iterations.

Scenario	+ve output	-ve output	Relative Improvement
High - High - Same - No	500	0	0.269%
High - High - Same - Yes	500	0	2.25%
High - High - Different - No	500	0	0.323%
High - High - Different - Yes	500	0	2.22%
High - Low - same - No	500	0	1.15%
High - Low - Same - Yes	500	0	2.59%
High - Low - Different - No	500	0	1.30%
High - Low - Different - Yes	500	0	2.18%
Low - High - Same - No	500	0	1.35%
Low - High - Same - Yes	497	3	10.8%
Low - High - Different - No	500	0	1.52%
Low - High - Different - Yes	407	93	4.07%
Low - Low - Same - No	500	0	1.79%
Low - Low - Same - Yes	500	0	5.18%
Low - Low - Different - No	488	12	2.04%
Low - Low - Different - Yes	499	1	4.58%

Table 4.7: Relative improvement of extended TrimSil using GPCM from 500 iterations.

Scenario	+ve output	-ve output	Relative Improvement
High - High - Same - No	500	0	0.268%
High - High - Same - Yes	500	0	1.94%
High - High - Different - No	500	0	0.322%
High - High - Different - Yes	500	0	0.916%
High - Low - same - No	500	0	1.14%
High - Low - Same - Yes	500	0	2.54%
High - Low - Different - No	500	0	1.29%
High - Low - Different - Yes	500	0	1.93%
Low - High - Same - No	500	0	1.34%
Low - High - Same - Yes	460	40	2.62%
Low - High - Different - No	500	0	1.51%
Low - High - Different - Yes	481	19	1.53%
Low - Low - Same - No	500	0	1.88%
Low - Low - Same - Yes	497	3	5.32%
Low - Low - Different - No	486	14	1.74%
Low - Low - Different - Yes	499	1	2.35%

As an alternative for TrimSil, the extended TrimSil achieves higher improvement than TrimSil in all simulated scenarios. The specific scenarios and clustering algorithms that the extended TrimSil performs better in are the same as TrimSil.

Negative relative improvement still exists in a few cases for the extended TrimSil, but it is observed less frequently than TrimSil. The number of scenarios containing

negative output reduces to 4 when using k-means, and 5 when using GPCM. Moreover, the amount of negative outputs observed from extended TrimSil is significantly less than the ones from TrimSil.

The extended TrimSil is accepted as a better version of TrimSil, as it returns a greater amount of improvement. However, its relative improvement remains little, therefore it is still not as competitive as GenSil.

4.3 Silhouette-Parameter Plot

In this section, two simulations are carried out to demonstrate the performance of the silhouette-parameter plot. The aim is to assess the alignment of the silhouette-parameter plot with the traditional silhouette plot. The first simulation has clearly separated clusters, while the second has fuzzy clusters.

4.3.1 Well-Separated Clusters Simulation

The first simulated dataset comprises 5 well-separated clusters. It consists of 250 observations sampled from a Gaussian distribution and divided into 5 distinct clusters of equal sizes by adjusting their location. An additional 5 outliers are artificially added to the dataset. Figure 4.4 is the graph of this simulated dataset.

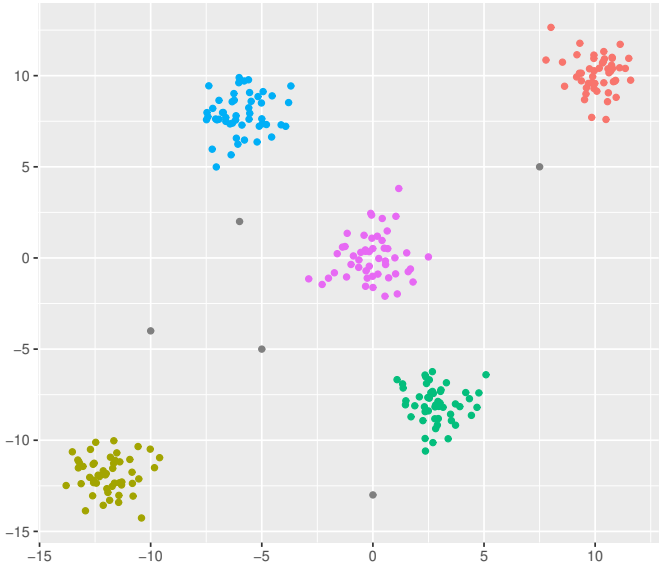

Figure 4.4: Simulated dataset with 5 well-separated clusters, 50 observations in each cluster, and 5 artificial outliers. Points with the same color belong to the same group, and the gray points are outliers.

Assuming no prior knowledge about the dataset, the k-means clustering is run multiple times with 2 to 7 centers, and their respective silhouette plots are generated. Figure 4.5 shows the silhouette plots of the clustering results obtained from k-means with 2 to 7 clusters. Based on the visual interpretation, we determine that 6 clusters represent the optimal number of clusters as its silhouette plot appears the darkest among all. Although the optimal k does not align with the true k, this result is reasonable considering the presence of outliers.

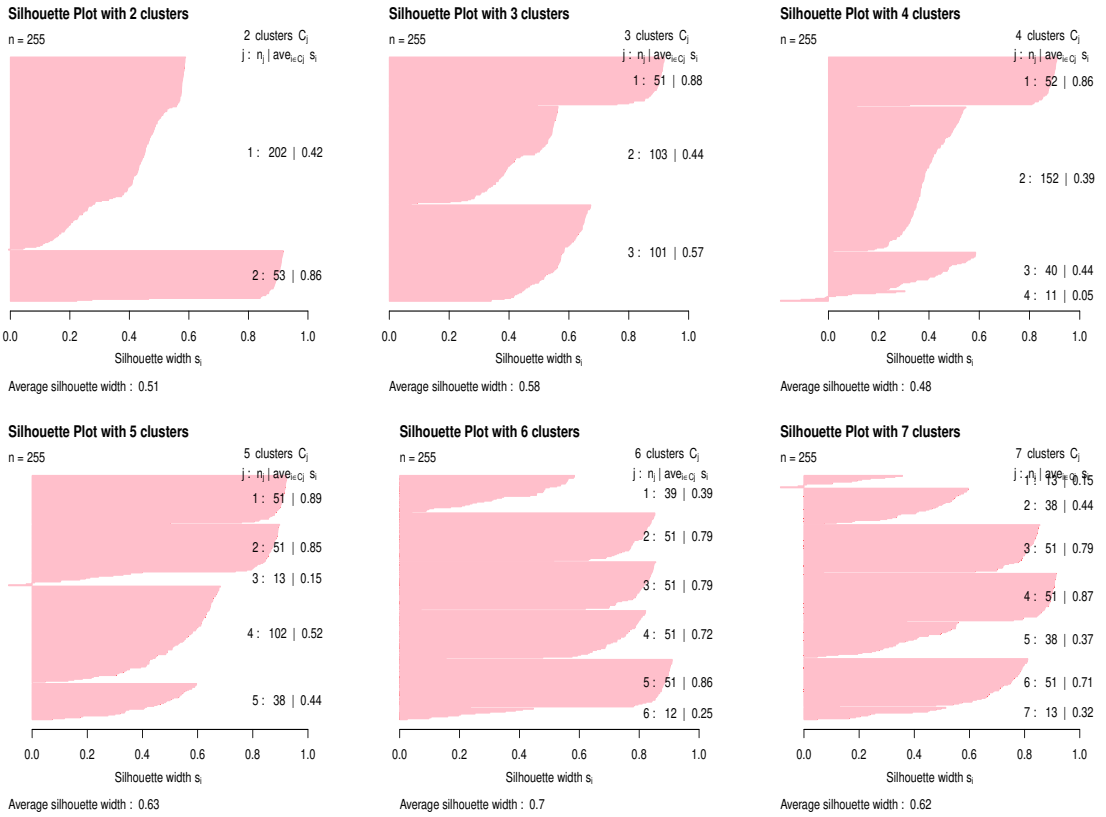

Figure 4.5: The traditional silhouette plots of the dataset with well-separated clusters, generated by k-means clustering with 2 to 7 clusters.

The silhouette-parameter plot is then generated from the same dataset, using k-means with the same number of clusters. The outcomes using GenSil, TrimSil, and the extended TrimSil are shown in Figure 4.6, 4.7, and 4.8, respectively.

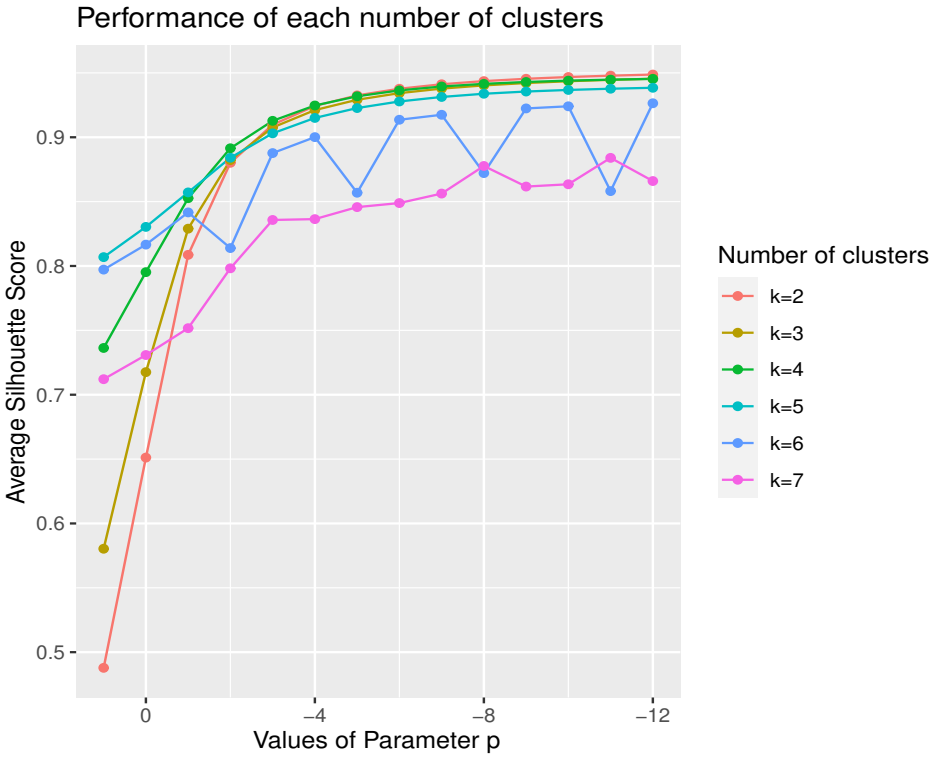

Figure 4.6: The silhouette-parameter plot of the dataset with well-separated clusters generated by GenSil, using 2 to 7 clusters.

Observe from Figure 4.6 that the lines corresponding to 2 to 5 clusters overlap each other as p decreases, and all of them appear to be relatively smooth, while the lines for 6 and 7 clusters fluctuate irregularly. Based on the description in Section 3.4, 2 to 5 clusters are all acceptable choices. This result does not follow our initial expectations and the result from the traditional method. The graph fails to distinguish the correct k from other k values that are smaller than it. But from another perspective, it effectively indicates the maximum reasonable value for k. The k whose line graph is not smooth is considered too large for the dataset. In this simulation, 5 is the largest acceptable k value, because once it exceeds 5, the line graph fluctuates. Additionally, this graph indicates that $p = -2$ is the optimal parameter because it is at the elbow

point for all lines.

Figure 4.7 presents a more desirable result obtained by TrimSil. Although the lines for 2 to 5 clusters are equally smooth, the line representing 5 clusters appears at the top. Therefore, a conclusion can be drawn easily from this graph that 5 clusters are the most ideal choice. However, the expected reversed elbow shape is not observed from this graph, providing no useful information on the choice of parameter t.

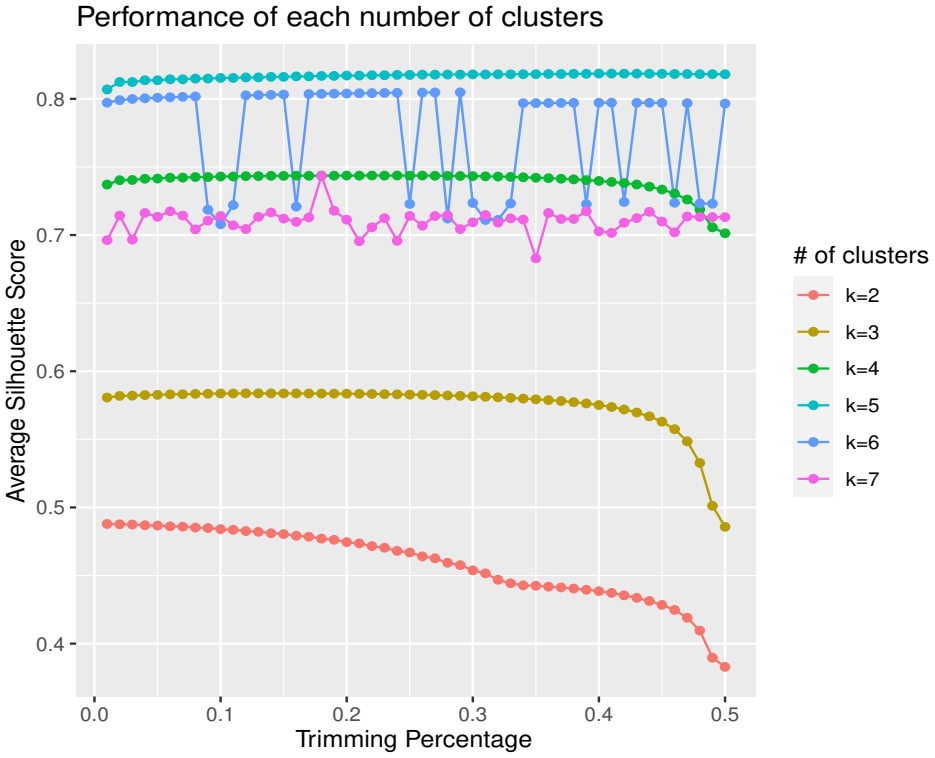

Figure 4.7: The silhouette-parameter plot of the dataset with well-separated clusters generated by TrimSil, using 2 to 7 clusters.

The resulting graph generated using extended TrimSil is also straightforward, as shown in Figure 4.8. The line corresponding to 5 clusters is the smoothest and highest among all the lines, indicating that this dataset is better partitioned into 5 clusters, which agrees with the result from TrimSil and the true labels. However, similar to

TrimSil, there is no clear indication of the optimal choice of the parameter from this graph.

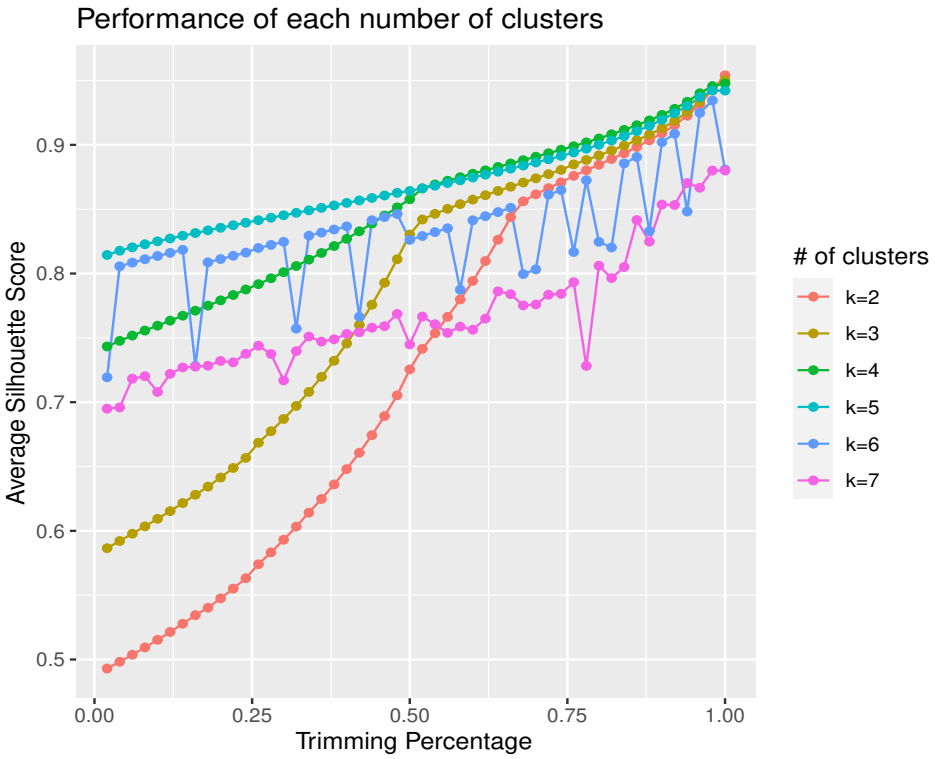

Figure 4.8: The silhouette-parameter plot of the dataset with well-separated clusters generated by extended TrimSil, using 2 to 7 clusters.

4.3.2 Fuzzy Clusters Simulation

The second simulated dataset also contains 250 observations divided into 5 clusters, and 5 outliers, as shown in Figure 4.9. Instead of having 5 well-separated clusters, this dataset is designed to have 2 clusters closely located to the other 2 clusters, and only 1 cluster far away. If the true labels are unknown, this dataset may give the impression of having three clusters of unequal sizes.

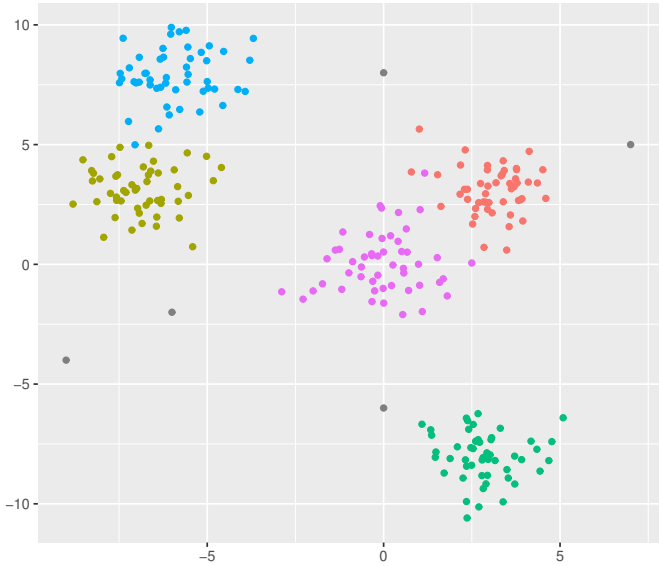

Figure 4.9: Simulated dataset with 5 fuzzy clusters, 50 observations in each cluster, and 5 artificial outliers. Points with the same color belong to the same group, and the gray points are outliers.

The silhouette plots of k-means with 2 to 7 centers are drawn in Figure 4.10. The graph for 4 clusters seems to have the widest silhouette width, and the graph for 5 clusters is also competitive. This suggests that partitioning this dataset into either 4 or 5 clusters is considered reasonable.

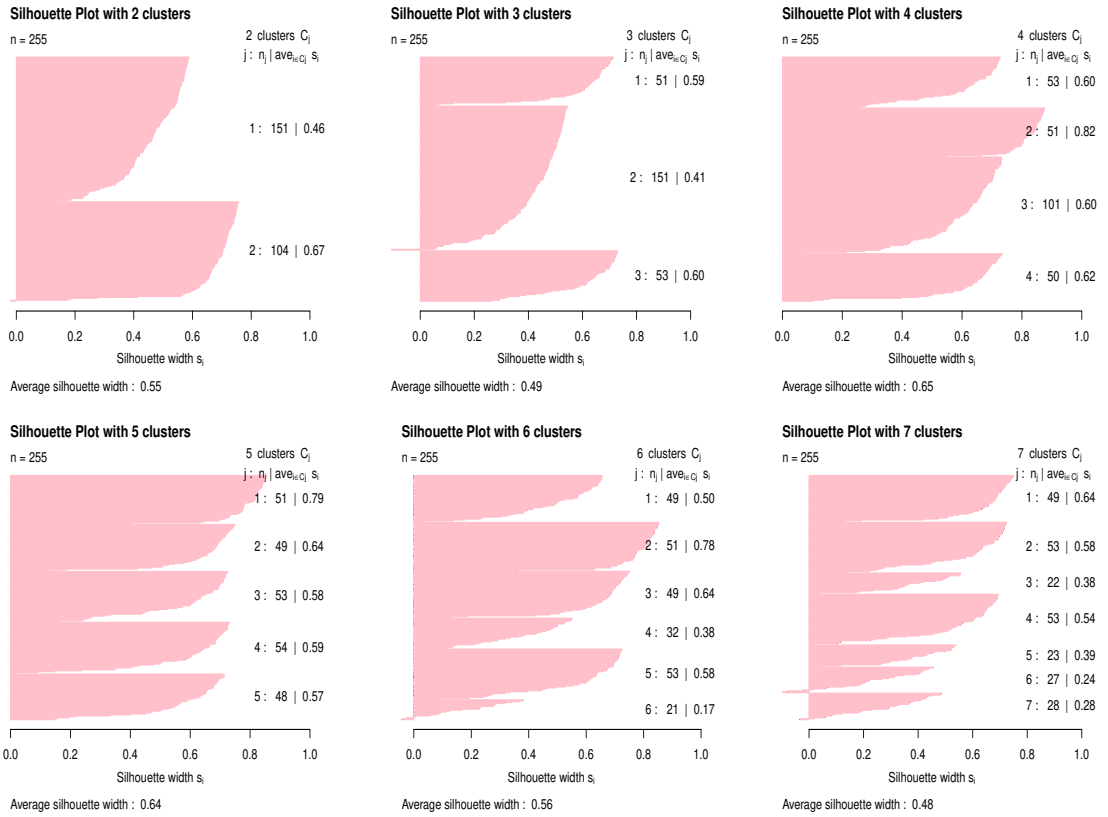

Figure 4.10: The traditional silhouette plots of the dataset with fuzzy clusters, generated by k-means clustering with 2 to 7 clusters.

Figure 4.11, 4.12, and 4.13 are the silhouette-parameter plots of this dataset using GenSil, TrimSil, and extended TrimSil.

When using with GenSil, the lines for 2 to 5 clusters appear to be smooth, and lines for 6 and 7 clusters fluctuate as shown in Figure 4.11. The lines for 2 and 3 clusters are closely located at the top, meanwhile, the lines for 4 and 5 clusters are lower. This silhouette-parameter plot suggests that either 2 or 3 clusters are the best k, 4 or 5 clusters are less desirable, and more than 5 clusters are not acceptable. This result does not align with the result from silhouette plots. Although this plot does not precisely identify closely located clusters, it effectively indicates that the maximum

acceptable value for k is 5 through the smoothness of the lines. Moreover, the elbow point for all lines occurs at $p = -3$, hence $p = -3$ is selected as the parameter for GenSil. Therefore, the silhouette-parameter plot when used with GenSil, is better to find the maximum acceptable value for k instead of the optimal k, and aid in determining the parameter p.

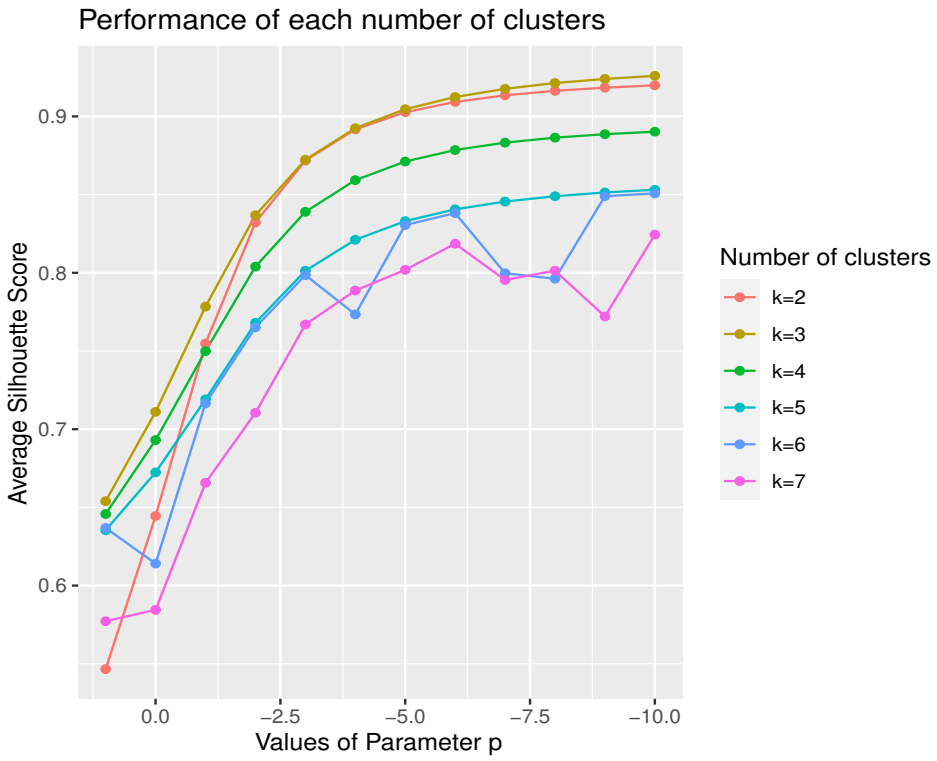

Figure 4.11: The silhouette-parameter plot of the dataset with fuzzy clusters generated by GenSil, using 2 to 7 clusters.

The graph generated by TrimSil is shown in Figure 4.12. The lines for 2 to 5 clusters are smooth, but the lines for 6 and 7 clusters are inconsistent and therefore not considered. The line for 3 clusters is located at the top among all lines, indicating that this dataset should be grouped into 3 clusters. Clustering into 4 or 5 clusters is also acceptable because their lines are just slightly below the line for 3 clusters.

This result aligns with the traditional silhouette plots for the most part. The lines are relatively flat rather than curved, and thus no conclusions about the parameter values can be drawn from this graph.

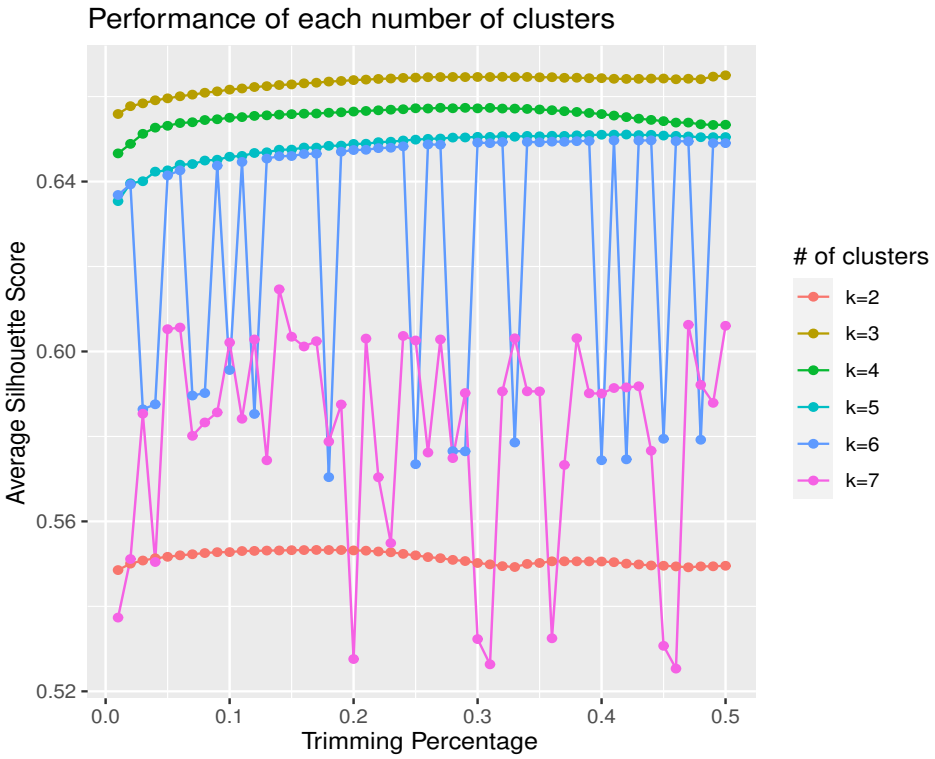

Figure 4.12: The silhouette-parameter plot of the dataset with fuzzy clusters generated by TrimSil, using 2 to 7 clusters.

Figure 4.13 shows the silhouette-parameter plot drawn using extended TrimSil. Similarly, only the lines corresponding to 2 to 5 clusters are smooth. Clustering into 3 groups is strongly supported, while 4 and 5 groups are equally good options. Clustering into 2 groups is not competitive until t reaches 0.5, meaning half of the observations are trimmed off. This result partly aligns with the traditional silhouette plots. Once again, the graph is inconclusive on the selection of the parameter t due to the location of the elbow point being unclear.

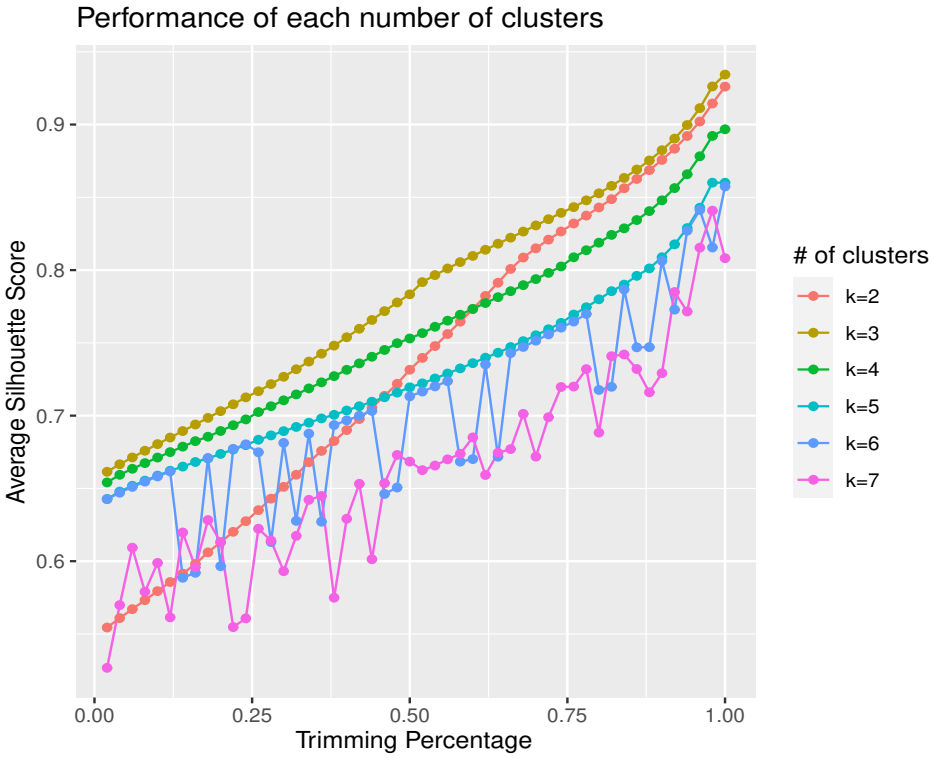

Figure 4.13: The silhouette-parameter plot of the dataset with fuzzy clusters generated by extended TrimSil, using 2 to 7 clusters.

In summary, based on the results from the two simulations above, the silhouette-parameter plot proves to be more effective and stable in identifying the maximum acceptable value for k, rather than the optimal value. Once the value of k exceeds this maximum, the line graph in the plot becomes non-smooth and easily distinguishable. TrimSil and extended TrimSil perform better with the silhouette-parameter plot compared to GenSil, as their suggested optimal k is closer to the true value, especially when the clusters are well-separated. However, it is worth noting that the silhouette-parameter plot only provides a suggested value for the parameter of GenSil, while the parameters for TrimSil and extended TrimSil need to be determined through other techniques.

Chapter 5

Application

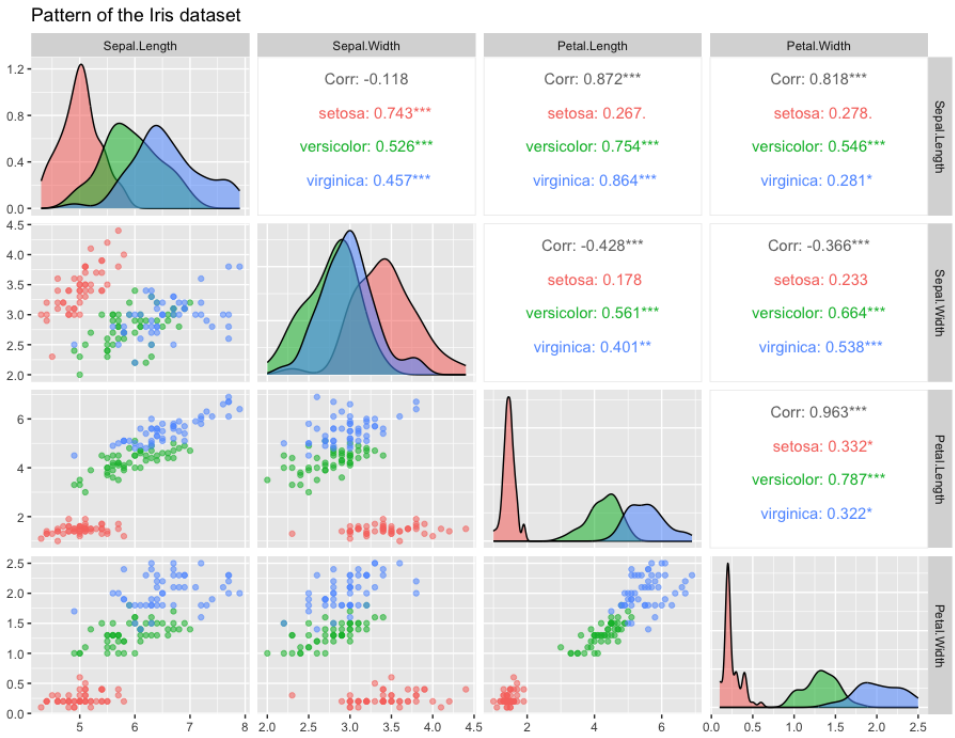

Figure 5.1: Pairs plot of the iris dataset.

The iris dataset, collected by Fisher (1936), consists of numeric measurements of sepal length, sepal width, petal length and petal width for 150 iris flowers. These samples are evenly distributed over three species: Setosa, Versicolor and Virginica. A small number of outliers are observed from Setosa and Virginica according to the pairs plot in Figure 5.1.

The true labels of the iris dataset are removed to create the condition for clustering. Note that the patterns from Versicolor and Virginica are overlapping, therefore a fuzzy partition between these two groups is expected. The silhouette-parameter plots using all three methods are drawn to determine the most ideal or the maximum number of clusters for the iris dataset. The resulting graphs are shown in Figures 5.2, 5.3, 5.4. The line corresponding to 2 is at the top in all three plots, followed by the line of 3 clusters. The results suggest that 2 clusters are a better choice for the iris dataset. However, considering the prior knowledge that the true number of clusters is 3 and the conclusion from Section 4.3, the performance of all three methods will be evaluated using both 3 and 2 clusters. $p = -8$ is selected as the parameter for GenSil as it is located at the elbow point. Due to the absence of the curve shape in the plots associated with TrimSil and extended TrimSil, both an exploratory analysis and `oclust` is run to estimate the proportion of outliers in the iris dataset. Both results agree that approximately 4 out of 150 observations are considered as outlying values, which takes about 3% of the entire dataset. Therefore $t = 0.03$ for both TrimSil and extended TrimSil.

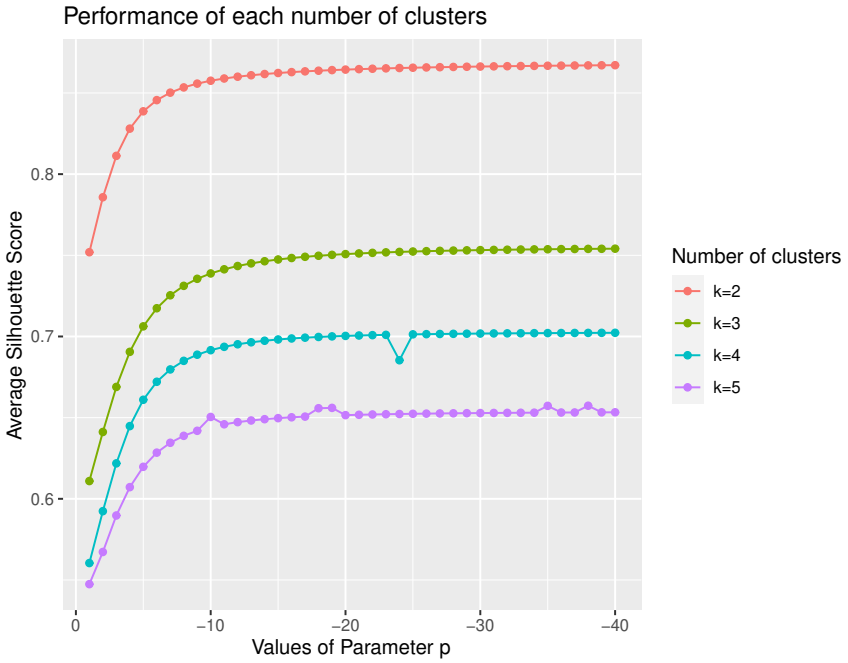

Figure 5.2: Silhouette-parameter plot associated with GenSil.

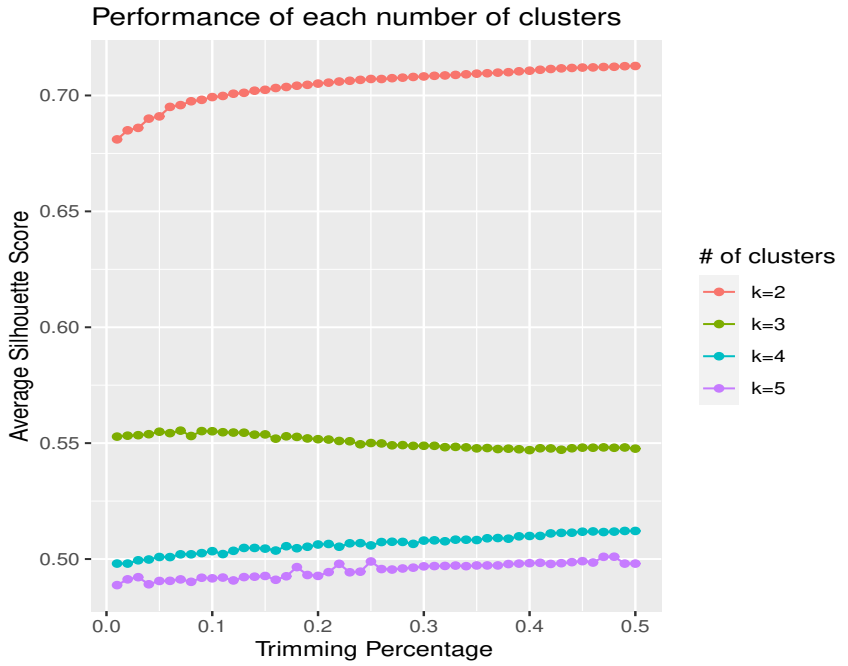

Figure 5.3: Silhouette-parameter plot associated with TrimSil.

Figure 5.4: Silhouette-parameter plot associated with extended TrimSil.

Firstly, the iris dataset is clustered into 3 groups using both k-means and GPCM. The clustering result from k-means has a silhouette score of 0.5528, which indicates a fair quality. However, when comparing the predicted labels from k-means with the true labels, it is found that 89.3% of the observations are classified correctly, with an ARI of 0.73. These verification indices suggest that the clustering performs satisfactorily, and hence this clustering result deserves a higher silhouette score. The results obtained from GPCM are similar: an intermediate silhouette score of 0.5012, but a high accuracy of 96.7% and a high ARI of 0.90. To further enhance the silhouette score, GenSil, TrimSil, and extended TrimSil are applied.

Table 5.1 provides a comparison among the original silhouette score, GenSil score, TrimSil score, and extended TrimSil score for the iris dataset using 3 clusters. All

methods return a higher silhouette score than the original. However, GenSil demonstrates a more significant improvement while TrimSil and extended TrimSil make little changes. The extended TrimSil score is slightly higher than the TrimSil score. Overall, GenSil yields more favorable silhouette scores for the iris dataset, whether applying k-means or GPCM clustering.

Table 5.1: Four silhouette scores of the iris dataset using k-means and GPCM (3 clusters)

	Original Silhouette Score	GenSil	TrimSil	extended TrimSil
K-Means	0.5528	0.7312	0.5534	0.5573
GPCM	0.5012	0.7048	0.5030	0.5078

The performance of partitioning the iris dataset into 2 groups is also assessed. However, the accuracy and the ARI are not applicable in this case as the number of clusters does not align with the true labels. Therefore, only the silhouette scores are considered. The silhouette scores obtained by k-means and GPCM are 0.6810 and 0.6867, respectively. Although these scores are higher compared to the 3 clusters scenario, they have not yet reached satisfactory. The GenSil, TrimSil and extended TrimSil scores are presented in Table 5.2.

Table 5.2: Four silhouette scores of the iris dataset using k-means and GPCM (2 clusters)

	Original Silhouette Score	GenSil	TrimSil	extended TrimSil
K-Means	0.6810	0.8534	0.6860	0.6925
GPCM	0.6867	0.8901	0.6898	0.6958

All methods produce a higher silhouette score than the original score, and the

scores obtained from k-means are almost identical to the ones from GPCM. Among the three modified silhouette scores, GenSil stands out by achieving the highest score and demonstrating the most improvement over TrimSil and extended TrimSil. As expected, extended TrimSil obtains a slightly higher score than TrimSil.

To summarize, the ranking of these three modified silhouette scores based on their performance is as follows: GenSil, extended TrimSil, and TrimSil. As a result, GenSil is recommended over the original or the other modified silhouette scores for the iris dataset. In addition to that, the silhouette-parameter plot suggests that partitioning the iris dataset into 2 clusters is better.

Chapter 6

Conclusion

GenSil, TrimSil, and extended TrimSil were introduced to reduce the impact of outliers on the calculation of the silhouette score. These methods replaced the arithmetic mean in the original silhouette score with the generalized mean, the trimmed mean and a modified trimmed mean, respectively. GenSil minimized the impact of outliers by adopting a negative p value, whereas TrimSil reduced the influence of outliers by discarding the extreme distances, and the extended TrimSil improved TrimSil by retaining the small distances while trimming.

A visualization technique, the silhouette-parameter plot, was proposed to help with the selection of the optimal number of clusters and parameter values for GenSil, TrimSil, and extended TrimSil. It improved the existing method by simplifying the process and demonstrating the relationship between the silhouette scores and the parameters from various cluster numbers in a single graph.

The performance of all methods was investigated through an extensive simulation study on 16 different cluster settings. The findings indicated that all three methods exhibited improved performance when the clusters were closely located and contained

a higher number of outliers. GenSil demonstrated better results in scenarios where the clusters were dispersed, whereas the effect of compactness on TrimSil and extended TrimSil was dependent on the level of separation between clusters. The size of the clusters did not significantly influence the performance of either GenSil or (extended) TrimSil. Additionally, the choice between k-means and GPCM did not affect the performance of all methods when the clusters were clearly separated or contained no outliers. However, all methods tended to perform better with k-means when the separation and compactness decreased. Furthermore, GenSil consistently generated an improved silhouette score, while TrimSil and extended TrimSil occasionally returned a lower score, particularly in closely located clusters.

Finally, the effectiveness of these modified silhouette scores was demonstrated on the iris dataset. The iris dataset was considered to have high classification accuracy and ARI, but a low silhouette score due to the presence of outliers. GenSil, TrimSil, and extended TrimSil had been proven to be effective in enhancing the silhouette scores, while GenSil increased the silhouette score more significantly than TrimSil and extended TrimSil. The silhouette-parameter plot suggested that its optimal number of clusters was 2.

In conclusion, when evaluating the quality of clustering results for datasets containing outliers or clusters with low separation, all of these three modified silhouette scores can be considered as a substitution for the original silhouette score. While GenSil demonstrates the greatest amount of improvement, TrimSil and extended TrimSil are found to be more useful in practice. These two methods align better with the silhouette-parameter plot to assist in the identification of the optimal number of clusters.

Appendix A

Proof of Property 1

Proof. The proof of Property 1 is built on Jensen's Inequality (Jensen, 1906):

If $\phi(x)$ is a convex function, then $\phi\left(\frac{1}{n}\sum_{j=1}^{n} x_j\right) \leq \frac{1}{n}\sum_{j=1}^{n}\phi(x_j)$.

If $\phi(x)$ is a concave function, then $\phi\left(\frac{1}{n}\sum_{j=1}^{n} x_j\right) \geq \frac{1}{n}\sum_{j=1}^{n}\phi(x_j)$.

To prove Property 1, the following three cases are considered:

Case 1: $p > 0 > q$

To prove $\mu_p \geq \mu_0$, i.e.,

$$\left(\frac{1}{n}\sum_{j=1}^{n} x_j^p\right)^{\frac{1}{p}} \geq \left(\prod_{j=1}^{n} x_j\right)^{\frac{1}{n}} \tag{A.1}$$

Take the natural log on both sides and get

$$\frac{1}{p}\log\left(\frac{1}{n}\sum_{j=1}^{n} x_j^p\right) \geq \frac{1}{n}\log\left(\prod_{j=1}^{n} x_j\right) = \frac{1}{n}\sum_{j=1}^{n}\log\left(x_j\right) \tag{A.2}$$

Given $p > 0$, multiply p on both sides and get

$$\log\left(\frac{1}{n}\sum_{j=1}^{n} x_j^p\right) \geq \frac{p}{n}\sum_{j=1}^{n}\log\left(x_j\right) = \frac{1}{n}\sum_{j=1}^{n}\log\left(x_j^p\right) \tag{A.3}$$

A.3 must hold to prove A.1. Knowing that log is a concave function, from Jensen's Inequality, $\log\left(\frac{1}{n}\sum_{j=1}^{n} x_j^p\right) \geq \frac{1}{n}\sum_{j=1}^{n}\log\left(x_j^p\right)$ is true. Therefore $\mu_p(\mathbf{x}) \geq \mu_0(\mathbf{x})$ for $p > 0$.

Given $0 > q$, $\mu_0(\mathbf{x}) \geq \mu_q(\mathbf{x})$ can be proved following the same procedure as above. Combining the two results above, we conclude that $\mu_p(\mathbf{x}) \geq \mu_0(\mathbf{x}) \geq \mu_q(\mathbf{x})$ for $p > 0 > q$.

Case 2: $p > q > 0$

To prove $\mu_p(\mathbf{x}) \geq \mu_q(\mathbf{x})$, i.e.,

$$\left(\frac{1}{n}\sum_{j=1}^{n} x_j^p\right)^{\frac{1}{p}} \geq \left(\frac{1}{n}\sum_{j=1}^{n} x_j^q\right)^{\frac{1}{q}} \tag{A.4}$$

Take both sides to the power of p

$$\frac{1}{n}\sum_{j=1}^{n} x_j^p \geq \left(\frac{1}{n}\sum_{j=1}^{n} x_j^q\right)^{\frac{p}{q}} \tag{A.5}$$

Note that $p > q > 0$, therefore $\frac{p}{q} > 1$, and hence $x^{\frac{p}{q}}$ is a convex function (for non-negative x). By Jensen's inequality, the following holds

$$\left(\frac{1}{n}\sum_{j=1}^{n} x_j^q\right)^{\frac{p}{q}} \leq \frac{1}{n}\sum_{j=1}^{n}\left(x_j^q\right)^{\frac{p}{q}} = \frac{1}{n}\sum_{j=1}^{n} x_j^p \tag{A.6}$$

A.5 holds, therefore proves that $\mu_p(\mathbf{x}) \geq \mu_q(\mathbf{x})$ for $p > q > 0$.

Case 3: $0 > p > q$

Similar to case 2, but the inequality sign flips as we take the power of p on both sides due to the fact that p is negative.

$$\frac{1}{n}\sum_{j=1}^{n} x_j^p \leq \left(\frac{1}{n}\sum_{j=1}^{n} x_j^q\right)^{\frac{p}{q}} \tag{A.7}$$

Given $0 > p > q$, we know that $0 < \frac{p}{q} < 1$, and $x^{\frac{p}{q}}$ is a concave function for non-negative x. Then based on Jensen's inequality, the inequality below is true:

$$\left(\frac{1}{n}\sum_{j=1}^{n} x_j^q\right)^{\frac{p}{q}} \geq \frac{1}{n}\sum_{j=1}^{n} \left(x_j^q\right)^{\frac{p}{q}} = \frac{1}{n}\sum_{j=1}^{n} x_j^p \tag{A.8}$$

A.7 is true, therefore $\mu_p(\mathbf{x}) \geq \mu_q(\mathbf{x})$ for $0 > p > q$.

This property can also be proved alternatively by taking the first derivative with respect to p and showing that it is non-negative.

Case 1: $p < 0$

$$\mu(p) = \left(\frac{1}{n}\sum_{j=1}^{n} x_j^p\right)^{\frac{1}{p}} = e^{\frac{1}{p}(\log\sum_{j=1}^{n} x_j^p - \log(n))}$$

$$\begin{aligned}\frac{\partial}{\partial p}\mu_p &= e^{\frac{1}{p}(\log\sum_{j=1}^{n} x_j^p - \log(n))}[-\frac{1}{p^2}\log\sum_{j=1}^{n} x_j^p + \frac{1}{p}\frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p} + \frac{1}{p^2}\log(n)] \\ &= e^{\frac{1}{p}(\log\sum_{j=1}^{n} x_j^p - \log(n))}[-\frac{1}{p^2}\log\sum_{j=1}^{n} x_j^p + \frac{1}{p}\frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p}] + e^{\frac{1}{p}(\log\sum_{j=1}^{n} x_j^p - \log(n))}\frac{1}{p^2}\log(n)\end{aligned}$$

The term $e^{\frac{1}{p}(\log\sum_{j=1}^{n} x_j^p - \log(n))}$ is an exponential function, therefore it is positive for all values of p. Given that $n \geq 1$, it is known that $\log(n) \geq 0$. Therefore, it is left to

show that $-\frac{1}{p^2}\log\sum_{j=1}^{n} x_j^p + \frac{1}{p}\frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p} \geq 0$.

$$
\begin{aligned}
-\frac{1}{p^2}\log\sum_{j=1}^{n} x_j^p + \frac{1}{p}\frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p} &\geq 0 \\
-\frac{1}{p^2}[\log\sum_{j=1}^{n} x_j^p - p\frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p}] &\geq 0 \\
\log\sum_{j=1}^{n} x_j^p - p\frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p} &\leq 0
\end{aligned}
$$

Given that $\log(x)$ is a concave function, therefore by Jensen's Inequality, $\log\sum_{j=1}^{n} x_j^p \geq \sum_{j=1}^{n}\log(x_j^p) = p\sum_{j=1}^{n}\log(x_j)$. Substitute this inequality into the previous formula and get:

$$
\begin{aligned}
p\sum_{j=1}^{n}\log(x_j) - p\frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p} &\leq 0 \\
\sum_{j=1}^{n}\log(x_j) - \frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p} &\geq 0 \\
\sum_{j=1}^{n}\log(x_j)\sum_{j=1}^{n} x_j^p - \sum_{j=1}^{n} x_j^p \log(x_j) &\geq 0 \\
\sum_{j=1}^{n}\log(x_j)\sum_{j=1}^{n} x_j^p &\geq \sum_{j=1}^{n} x_j^p \log(x_j)
\end{aligned}
$$

The above inequality is true by the Cauchy–Schwarz inequality. Therefore the first derivative of $\mu(p)$ is greater or equal to 0.

Case 2: $p = 0$

$$\mu(p) = (\prod_{j=1}^{n} x_j)^{\frac{1}{n}}$$
$$\frac{\partial}{\partial p}\mu_p = \frac{\partial}{\partial p}(\prod_{j=1}^{n} x_j)^{\frac{1}{n}} = 0$$

Therefore $\frac{\partial}{\partial p}\mu_p \geq 0$ holds.

Case 3: $p > 0$

This case can be proved followed by a similar approach as in case 1, where we want to show that $-\frac{1}{p^2}\log\sum_{j=1}^{n} x_j^p + \frac{1}{p}\frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p} \geq 0$.

$$-\frac{1}{p^2}\log\sum_{j=1}^{n} x_j^p + \frac{1}{p}\frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p} \geq 0$$
$$\frac{1}{p^2}[\log\sum_{j=1}^{n} x_j^p - p\frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p}] \geq 0$$
$$\log\sum_{j=1}^{n} x_j^p - p\frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p} \geq p\sum_{j=1}^{n}\log(x_j) - p\frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p} \geq 0$$
$$\sum_{j=1}^{n}\log(x_j) - \frac{\sum_{j=1}^{n} x_j^p \log(x_j)}{\sum_{j=1}^{n} x_j^p} \geq 0$$
$$\sum_{j=1}^{n}\log(x_j)\sum_{j=1}^{n} x_j^p - \sum_{j=1}^{n} x_j^p \log(x_j) \geq 0$$
$$\sum_{j=1}^{n}\log(x_j)\sum_{j=1}^{n} x_j^p \geq \sum_{j=1}^{n} x_j^p \log(x_j)$$

Hence, the first derivative of $\mu(p)$ is non-negative with respect to p, which implies that the generalized mean is monotonic increasing. □

Bibliography

Amruthnath, N. and Gupta, T. (2018). Fault class prediction in unsupervised learning using model-based clustering approach. In *2018 International Conference on Information and Computer Technologies (ICICT)*, pages 5–12.

Banfield, J. D. and Raftery, A. E. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821.

Batool, F. and Hennig, C. (2021). Clustering with the average silhouette width. *Computational Statistics and Data Analysis*, **158**, 107190.

Celeux, G. and Govaert, G. (1992). A classification em algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, **14**(3), 315–332.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, **28**(5), 781–793.

Clark, K. M. and McNicholas, P. D. (2022). *oclust: Gaussian Model-Based Clustering with Outliers*. R package version 0.2.0.

de Craen, S., Commandeur, J. J. F., Frank, L. E., and Heiser, W. J. (2006). Effects of group size and lack of sphericity on the recovery of clusters in k-means cluster analysis. *Multivariate Behavioral Research*, **41**(2), 127–145. PMID: 26782907.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, **39**(1), 1–38.

Dudek, A. (2020). Silhouette index as clustering evaluation tool. In *Classification and Data Analysis*, pages 19–33, Cham. Springer International Publishing.

Everitt, B., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster Analysis*. Wiley.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, **7**(2), 179–188.

Fraiman, R. and Muniz, G. (2001). Trimmed means for functional data. *Test*, **10**, 419–440.

Gormley, I. C., Murphy, T. B., and Raftery, A. E. (2023). Model-based clustering. *Annual Review of Statistics and Its Application*, **10**(1), 573–595.

Gou, J., Ma, H., Ou, W., Zeng, S., Rao, Y., and Yang, H. (2019). A generalized mean distance-based k-nearest neighbor classifier. *Expert Systems with Applications*, **115**, 356–372.

Govender, P. and Sivakumar, V. (2020). Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). *Atmospheric Pollution Research*, **11**(1), 40–56.

Grün, B. (2019). Model-based clustering. In *Handbook of mixture analysis*, pages 157–192. CRC Press, Taylor & Francis Group.

Hardy, G. H., Littlewood, J. E., and Pólya, G. (1952). *Inequalities*. Cambridge university press.

Huang, Z. (1997). A fast clustering algorithm to cluster very large categorical data sets in data mining. *Dmkd*, **3**(8), 34–39.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, **2**(1), 193–218.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.

Jensen, J. L. W. V. (1906). On convex functions and inequalities between mean values. *Acta mathematica*, **30**(1), 175–193.

Kaufman, L. and Rousseeuw, P. J. (1987). Clustering by means of medoids. In *Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland*, volume 31.

Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction To Cluster Analysis*. Wiley, New York.

Lengyel, A. and Botta-Dukát, Z. (2019). Silhouette width using generalized mean—a flexible method for assessing clustering efficiency. *Ecology and Evolution*, **9**(23), 13231–13243.

Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining*, pages 911–916.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, **28**(2), 129–137.

Luukka, P. and Leppälampi, T. (2006). Similarity classifier with generalized mean applied to medical data. *Computers in Biology and Medicine*, **36**(9), 1026–1040.

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

McNicholas, P. D. (2016a). *Mixture model-based classification*. CRC Press, Boca Raton.

McNicholas, P. D. (2016b). Model-based clustering. *Journal of Classification*, **33**(3), 331–373.

McParland, D. and Gormley, I. C. (2016). Model based clustering for mixed data: clustmd. *Advances in Data Analysis and Classification*, **10**(2), 155–169.

Namratha, M. and Prajwala, T. (2012). A comprehensive overview of clustering algorithms in pattern recognition. *IOSR Journal of Computer Engineering*, **4**(6), 23–30.

Oh, J. and Kwak, N. (2016). Generalized mean for robust principal component analysis. *Pattern Recognition*, **54**, 116–127.

Pandit, S. and Gupta, S. (2011). A comparative study on distance measuring approaches for clustering. *International journal of research in computer science*, **2**(1), 29–31.

Pocuca, N., Browne, R. P., and McNicholas, P. D. (2022). *mixture: Mixture Models for Clustering and Classification*. R package version 2.0.5.

Punzo, A. and Ingrassia, S. (2015). Parsimonious generalized linear gaussian cluster-weighted models. In *Advances in Statistical Models for Data Analysis*, pages 201–209, Cham. Springer International Publishing.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

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

Rokach, L. and Maimon, O. (2005). Clustering methods. *Data mining and knowledge discovery handbook*, pages 321–352.

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.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, **6**(2), 461–464.

Shahapure, K. R. and Nicholas, C. (2020). Cluster quality analysis using silhouette score. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 747–748.

Shukla, S. and Naganna, S. (2014). A review on k-means data clustering approach. *International Journal of Information & Computation Technology*, **4**(17), 1847–1860.

Shutaywi, M. and Kachouie, N. N. (2021). Silhouette analysis for performance evaluation in machine learning with applications to clustering. *Entropy*, **23**(6).

Tan, P.-N., Steinbach, M., Karpatne, A., and Kumar, V. (2019). *Introduction to Data Mining*. Pearson, NY, NY, second edition. edition.

Van der Laan, M., Pollard, K., and Bryan, J. (2003). A new partitioning around medoids algorithm. *Journal of Statistical Computation and Simulation*, **73**(8), 575–584.

Wang, F., Franco-Penya, H.-H., Kelleher, J. D., Pugh, J., and Ross, R. (2017). An analysis of the application of simplified silhouette to the evaluation of k-means clustering validity. In *Machine Learning and Data Mining in Pattern Recognition: 13th International Conference, MLDM 2017, New York, NY, USA, July 15-20, 2017, Proceedings 13*, pages 291–305. Springer.