CLUSTERABILITY, MODEL SELECTION AND EVALUATION

A Dissertation Presented

by

KAIXUN HUA

Submitted to the Office of Graduate Studies,
University of Massachusetts Boston,
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2019

Computer Science Program

CLUSTERABILITY, MODEL SELECTION AND EVALUATION

A Dissertation Presented

by

KAIXUN HUA

Approved as to style and content by:

Dan A. Simovici, Professor
Chairperson of Committee

Marc Pomplun, Professor
Member

Ping Chen, Associate Professor
Member

Wei Ding, Associate Professor
Member

Dan A. Simovici, Program Director
Computer Science Program

Peter Fejer, Chairperson
Computer Science Department

ABSTRACT

CLUSTERABILITY, MODEL SELECTION AND EVALUATION

MAY 2019

KAIXUN HUA
B.Sc., SHANGHAI JIAO TONG UNIVERSITY
M.Eng., CORNELL UNIVERSITY
M.Sc., UNIVERSITY OF MASSACHUSETTS BOSTON
Ph.D., UNIVERSITY OF MASSACHUSETTS BOSTON

Directed by: Professor Dan A. Simovici, Professor

Clustering is a central topic in unsupervised learning and has a wide variety of applications. However, the increasing needs of clustering massive datasets and the high cost of running clustering algorithms poses difficult problems for users, while to select the best clustering model with a suitable number of clusters is also a primary focus. In this thesis, we mainly focus on determining whether a data set is clusterable, and what is the natural number of clusters of in a dataset.

First, we approach data clusterability from an ultrametric-based perspective. A novel approach to determine the ultrametricity of a dataset is proposed via a special type of matrix product and via this measure, we can evaluate the clusterability of it. Then, we show that our method of matrix product on the distance matrix will finally generate a sub-dominant ultrametric distance space of the original dataset. In addition, if a dataset has a unimodal or poorly constructed structure, its ultrametricity will be lower than other datasets with the same cardinality. We also show that by promoting the clusterability of a dataset, a poor clustering algorithm will perform better on the same dataset.

Secondly, we present a technique grounded in information theory for determining the natural number of clusters existent in a data set. Our approach involves a bi-criterial optimization that makes use of the entropy and the cohesion of a partition. Additionally, the experimental results are validated by using two quite distinct clustering methods: the k-means algorithm and Ward hierarchical clustering and their contour curves. We also show that by modifying the parameter, our approach can handle dataset with heavily imbalanced clustering structure, which is further complicated in practice.

ACKNOWLEDGMENTS

It is my great honor to convey my appreciation to everyone who assistant me during the path of pursuing my PhD degree. It is my pleasure to work with so many excellent colleagues in UMass Boston.

I would like to first deliver my sincere gratitude to my adviser, Prof. Dan. A. Simovici. It is my fortune to have such a responsible professor as my mentor. I thank him for his continuous support and advising on enhancing my quality for the discovery and conquering of research problems. His solid background in mathematics and machine learning can always strengthen my work until the success. His enthusiasm for research also infected me, fostering my passion for continuous exploration in the field of data mining. I can still remember the time that he revised my writing step by step or the period that we discussed problems and triggered many innovative ideas. The experience I gathered from him intrigued me to devote my life into the academic career and motivated me to involve myself into scientific research.

I should also express my appreciations to Prof. Wei Ding and Prof Ping Chen, who gave me a lot of support and encouragement on my study and school work. Thanks to them for sharing their academic and research experience to me. Also, I am glad they became members of my dissertation committee.

I took Prof. Marc Pomplun's computer vision class when I first entered UMass Boston. I thank him for his clear and precise explanations on the application of machine learning in computer vision. His modesty and his affinity for research influenced me a lot. And also, I thank him for being one of my dissertation committee member.

I would like to express my great thankfulness to all of my friends and colleagues: Tong Wang, Yong Zhuang, Yahui Di, Jipeng Qiang, Rosanne Vetro, Amir Zarring-

halam, Yi Ren, Shaohua Jia, Siyuan Gong, Matthew Almeida, and many others. I appreciate their support and friendship during my PhD study. In particular, many thanks to my friend and colleague Joshua Yee, who helped me with the proofreading.

I can not thank my parents (Yinhua Hua, Xinbao Zhang) and my wife's family enough. Without their long-standing patience, comprehension and encouragement, I wouldn't be able to focus deeply into my research and get through the study process. Finally, I express my greatest appreciation to my significant other, Ning Ning You, with her persistent understanding and consideration. I thank her for accompanying me on this journey, through both the good and the tough times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1
INTRODUCTION

1.1 Background

Clustering is the prototypical unsupervised learning activity which consists of identifying cohesive and well-differentiated groups of records in data. It aims to partition a set of objects such that similar objects will be assigned to the same group while those that are dissimilar will be placed in different groups [40].

However, this definition is not entirely satisfactory. It is still hard to define exactly when two objects are similar and to what degree. The general challenges of clustering are specified in [40]:

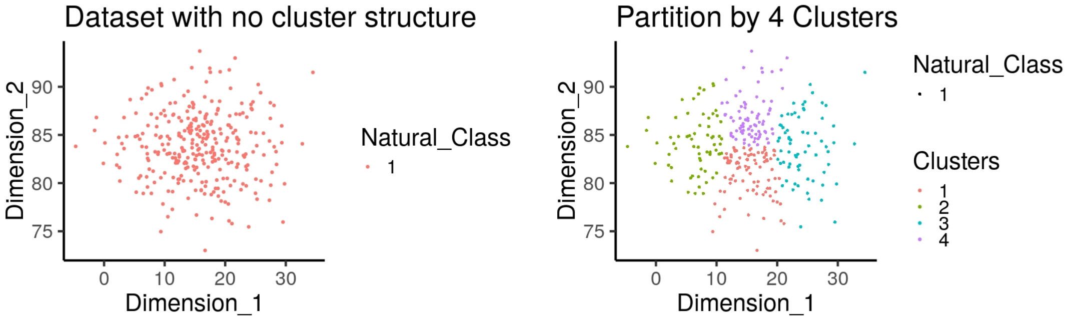
- What is a cluster?
- What features should be used?
- Should the data be normalized?
- Does the data contain any outliers?
- How do we define the pair-wise similarity?
- How many clusters are present in the data?
- Which clustering method should be used?
- Does the data have a natural clustering structure?
- Are the discovered clusters and partition valid?

In this thesis, our work mainly focuses on solving the questions related to clustering tendency and cluster validity. These two topics bear importance at very different times in the clustering process. Clustering tendency is important before generating partitions, while cluster validity is considered after running the clustering algorithms.

Clustering Tendency Generally, before executing the clustering task, it is worthwhile to analyze the underlying structure of the data set. Running a clustering algorithm is expensive, especially with the rapid increase of data size recently. This motivates the users to asses the “goodness” of the data set before running the clustering algorithm. Here, the “goodness” refers to how easily the data set can be clustered; more technically, we seek to determine how well the feature (or representation) separates the data set into clusters.

Despite the utility of checking clustering tendency of a data set, there is still no clear definition for measuring the “goodness” of a data set. Without a measure of “goodness”, we risk attempting to cluster a dataset with no underlying clustering structures. Any current clustering algorithm will still produce a result, even though that result will fail to accurately represent the data. These problems can create not only a waste of resources and time but also misleading results for potential clients. Figure 1.1a and Figure 1.1b give an example of “null” situation for clustering. This data is generated from one Gaussian Distribution and has no cluster structure in it. However, if we apply k-means with $k = 4$ to it, a clustering result with four clusters still be produced, even though these clusters are meaningless for understanding the data set.

Cluster Validity Even if we can determine there is a clustering structure in a data set, we still face new problems after running the clustering algorithms. It is still necessary to justify whether the produced clustering really is the best clustering representation of the data set.



(a) *A dataset with no cluster structure* (b) *Partition by k-means with 4 clusters.*

Figure 1.1: A Dataset without any cluster structures can still be partitioned by k-means clustering algorithm.

Several issues arise in evaluating the "goodness" of a clustering result. One such issue is deciding if clusters produced by the algorithm really reflect some intuitive grouping of the data set. It is entirely possible for an algorithm to make a clustering that makes sense numerically but has no practical use.

Different clustering algorithms will probably give different clustering results for the same data set. Even for a single clustering algorithm, changing its parameter will create a significant variance on the final clustering result. In these cases, it is necessary to consider which algorithm and which parameters produced the most suitable partition.

In particular, most clustering algorithms need a parameter k that specifies the number of clusters to detect. Determining the best choice of k for such algorithms is a long-standing and challenging problem that has attracted a great number of investigators. Determining this k is intimately related to determining the "natural" clustering structure in a data set, as discussed above.

The choice of optimal parameters is made even more difficult by the fact that many algorithms require a dissimilarity/similarity measure on a data set. Given the tremendous number of choices for dissimilarity/similarity measures, it is even harder to select the most suitable one. Currently, there is still no universal method to help the users to make this choice.

Figure 1.2 gives an example of a data set generated from five Gaussian Distribution. In this data set, k-means clustering algorithm is run on it and we can see that given different parameter of k, the final clustering results will fit the data set with the corresponding number of groups.

As we mentioned above, many of the problems for clustering analysis mainly come from the ambiguity of the definition of similarity (or dissimilarity) between objects. Because of this ambiguity, the clustering results may be affected by the usage of different types of clustering algorithm. In addition, the choice of parameters of the clustering algorithm may also influence the results.

There are several types of similarity or dissimilarity for defining the closeness between two objects. A general definition of dissimilarity is as follows:

Definition 1.1.1. *A dissimilarity on a set S is a mapping $d : S \times S \longrightarrow \mathbb{R}$ such that*

(i) $d(x, y) \geqslant 0$ *and* $d(x, y) = 0$ *if and only if* $x = y$*;*

(ii) $d(x, y) = d(y, x)$*;*

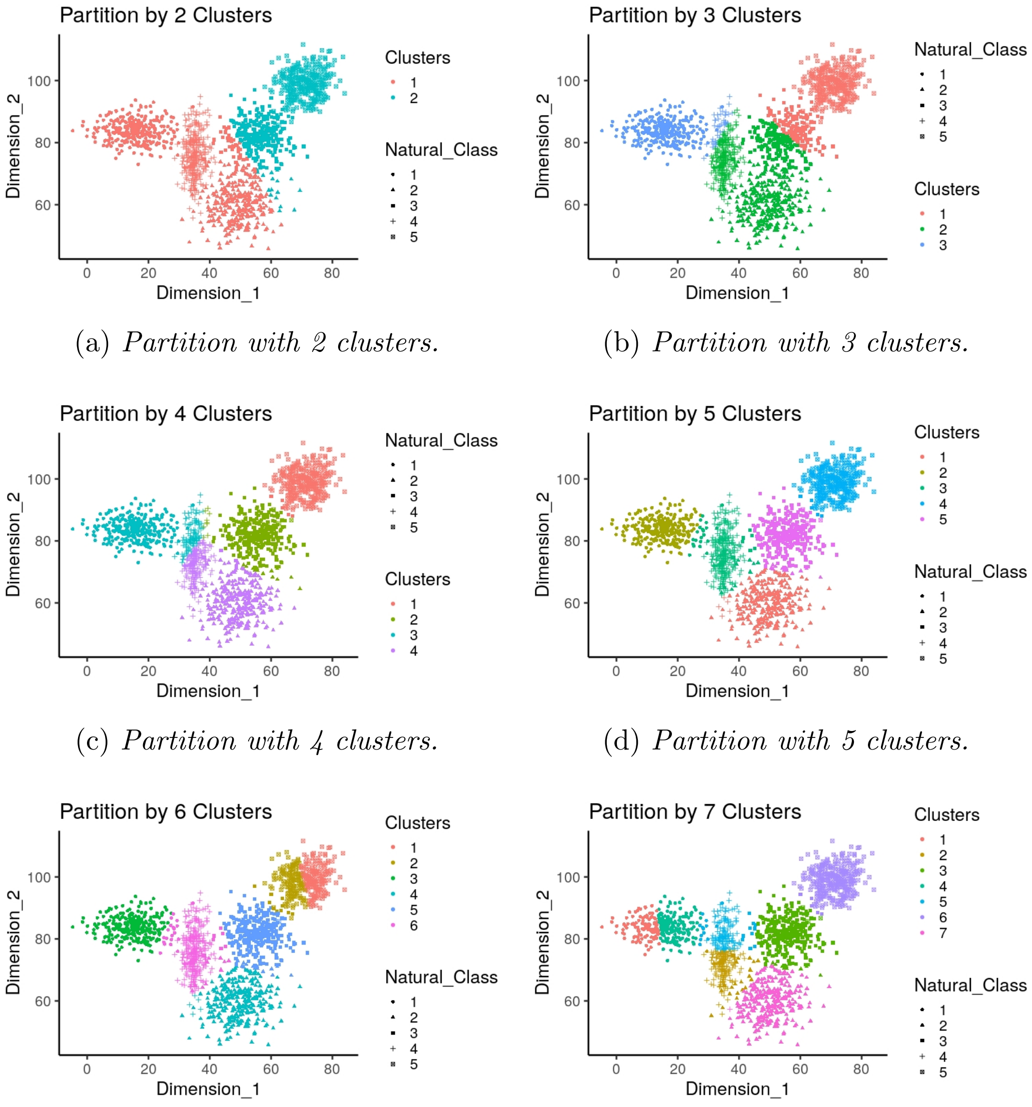
A dissimilarity on S that satisfies the triangular inequality

$$d(x, y) \leqslant d(x, z) + d(z, y)$$

for every $x, y, z \in S$ is a metric and we usually term $d(x, y)$ as the distance between points x and y.

The most commonly used distance is the Minkowski distance. Let two vectors $\mathbf{x} = (x_1, x_2, \dots x_n), \mathbf{y} = (y_1, y_2, \dots y_n)$, we can have the definition of Minkowski distance of order p between $\mathbf{x}, \mathbf{y}$ as follow:

Definition 1.1.2. $d_p(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_p = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}$



(a) *Partition with 2 clusters.*

(b) *Partition with 3 clusters.*

(c) *Partition with 4 clusters.*

(d) *Partition with 5 clusters.*

(e) *Partition with 6 clusters.*

(f) *Partition with 7 clusters.*

Figure 1.2: A Dataset with five Gaussian distributed classes which is separated by k-means algorithm for different values of k. If we select a suitable k, we can have a better clustering result than others. However, even if a wrong value of k is selected, the algorithm still returns a clustering result.

With a different choice of p, d_p can be corresponding to the different type of distance. For instance, $p = 1$ corresponds to the Manhattan distance while $p = 2$ will generate the Euclidean distance.

One compelling case is the Chebyshev distance, which can be acquired when p reaches infinity:

$$\lim_{p\to\infty}(\sum_{i=1}^{n}|x_i - y_i|^p)^{\frac{1}{p}} = \max_{i=1}^{n}|x_i - y_i|$$

All metrics above did not consider the weight of different variables. In their definitions, each variable has the same importance. In practice, some variables may probably more significant than others. Therefore, we can introduce the relationship between each variable and forms the Mahalanobis distance.

Suppose we have a covariance matrix S, then the distance between $\mathbf{x}$ and $\mathbf{y}$ is defined as follow:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{|\mathbf{x} - \mathbf{y}|^T S|\mathbf{x} - \mathbf{y}|}$$

If we set S as the identity matrix, then the Mahalanobis distance will become the Euclidean distance. If we let the covariance matrix as an arbitrary diagonal matrix, then we can weight each dimension with different values on the diagonal of S.

Besides the arbitrary choice of metrics on a dataset, the selection of distinct clustering algorithms will also lead to vastly different clustering results. There are several types of clustering algorithms, and each clustering algorithm performs differently depending on the shape of the data set of interest. Usually, partitional-based clustering is commonly used in practice, which is best for data sets for which we expect spherical-shaped clusters. The most commonly used algorithm of this type is k-means clustering algorithm. This algorithm requires a parameter k that specifies the number of clusters. The final result of k-means clustering is to achieve "close enough for points within-cluster and far enough for points between-cluster".

Initially, users need to select k random points as the center points of the preliminary clusters, then for each point in the data set, assign it to the cluster whose center is closest. Then based on the preliminary clustering, we recalculate the locations of center points and reassign the data points to the clusters corresponding to the new closest centers. We repeat this process until the locations are stabilized and the final assignment forms the final clustering result. Depending on the definition of "closeness" or the dissimilarity between points, k-means clustering algorithm will generate various clustering results.

Another popular type of clustering algorithm is the hierarchical clustering, which exists two forms: agglomerative and divisive. The first type declares each single point as a cluster, then combines "close" clusters until every point is contained in one single cluster. Finally, by keeping track the process of combination of clusters, a tree structure of the data set can be established. The divisive type executes the same task but in a reverse way. It starts from one single cluster which contains all points, and it separates the cluster into smaller pieces until every cluster only has one single point. Similarly to before, we can also create a tree structure that visually describes the division process. In both case, we refer to the tree as dendrogram.

The method for determining which clusters are close is known as "linkage". Each choice of linkage creates a unique hierarchical clustering algorithm. The commonly used linkages are "single-linkage", "complete-linkage", "average-linkage", "Ward-linkage", etc. Compared to the partitional-based clustering algorithms, hierarchical clustering methods can competently handle non-spherical shaped clusters and have no parameter k that requires the users to select the number of clusters before running the algorithm. Users can easily get a partition of the data set by choosing a stopping point in the agglomerative or divisive process of the algorithm.

1.2 Our Contribution

In this thesis, we aim to address issues in clustering tendency and cluster validity. We mainly utilize the concept of ultrametric and partitional generalized entropy to solve the problems.

We first provide the reader with an introduction to the basic knowledge and concepts related to ultrametrics. We also introduce the notion of ultrametricity, which measures how close a dissimilarity is to an ultrametric. The idea of β-entropy and the definition of a metric on a partition space based on β-entropy are also illustrated.

We separate the thesis into two parts. In the first, we consider the problem of clustering tendency. We define the concept of clusterability and show that it checks whether a dataset has a significant "natural" cluster structure. In particular, there are two findings of note:

- By increasing the clusterability of the dataset, we can improve the clustering results from some conventional clustering algorithms.
- With a given radius r, an r-spherical clustering can be performed on the dataset. This clustering result enables the users to find the natural number of clusters and detect the corresponding outliers.

In the second part, we will apply the concept of multi-objective optimization to define a dual-criteria for detecting the number of clusters. Based on the monotonicity of the function of the sum of squared error(sse) and the anti-monotonicity of partitional β-entropy with respect to the number of clusters k, we can form a compromise between these two criteria and achieve a reasonable k for the dataset. We also explore the performance of our method on a dataset with imbalanced cluster distribution. We conclude that by varying the value of β(mainly decreasing it), we can achieve a better result for detecting the number of clusters on a dataset with imbalanced clusters. Finally, we propose a method to validate the result of our dual-criteria method. This method makes use of the distance between partitions generated by different algorithms. Ostensibly, if a natural clustering structure exists in data, two clustering algorithms should produce similar clustering results. Therefore, the distance between such clustering results will be minimal.

CHAPTER 2

PRELIMINARIES

2.1 Ultrametric

If the stronger inequality

$$d(x, y) \leqslant \max\{d(x, z), d(z, y)\}$$

is satisfied instead of the traditional triangular inequality, d is said to be an *ultrametric* and the pair (S, d) is an *ultrametric space.*

Definition 2.1.1. *A* closed sphere *in (S, d) is a set $B[x, r]$ defined by*

$$B[x, r] = \{y \in S \mid d(x, y) \leqslant r\}.$$

When (S, d) is an ultrametric space two spheres having the same radius r in (S, d) are either disjoint or coincide [74].

Definition 2.1.2. *the collection of closed spheres of radius r in S, $\mathcal{C}_r = \{B[x, r] \mid r \in S\}$ is a partition of S; we refer to this partition as an* r-spheric clustering *of (S, d).*

In an ultrametric space (S, d) every triangle is isosceles. Indeed, let $T = (x, y, z)$ be a triplet of points in S and let $d(x, y)$ be the least distance between the points of T. Since $d(x, z) \leqslant \max\{d(x, y), d(y, z)\} = d(y, z)$ and $d(y, z) \leqslant \max\{d(y, x), d(x, z)\} = d(x, z)$, it follows that $d(x, z) = d(y, z)$, so T is isosceles; the two longest sides of this triangle are equal.

It is interesting to note that every r-spheric clustering in an ultrametric space is a perfect clustering [4]. This means that all of its in-cluster distances are smaller than all of its between-cluster distances. Indeed, if x, y belong to the same cluster $B[u, r]$ then $d(x, y) \leqslant r$. If $x \in B[u, r]$ and $y \in B[v, r]$, where $B[u, r] \cap B[v, r] = \emptyset$, then $d(v, x) > r$, $d(y, v) \leqslant r$ and this implies $d(x, y) = d(x, v) > r$ because the triangle (x, y, v) is isosceles and $d(y, v)$ is not the longest side of this triangle.

Example 2.1.3. Let $S = \{x_i \mid 1 \leqslant i \leqslant 8\}$ and let (S, d) be the ultrametric space, where the ultrametric d is defined by the following table:

$d(x_i, x_j)$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	4	4	10	10	16	16	16
x_2	4	0	4	10	10	16	16	16
x_3	4	4	0	10	10	16	16	16
x_4	10	10	10	0	6	16	16	16
x_5	10	10	10	6	0	16	16	16
x_6	16	16	16	16	16	0	4	4
x_7	16	16	16	16	16	4	0	4
x_8	16	16	16	16	16	4	4	0

The closed spheres of this spaces are:

$$\begin{aligned} B[x_i, r] &= \begin{cases} \{x_i\} & \text{for } r < 4, \\ \{x_1, x_2, x_3\} & \text{for } 4 \leqslant r < 10, \\ \{x_1, x_2, x_3, x_4, x_5\} & \text{for } 10 \leqslant r < 16, \\ S & \text{for } r = 16, \end{cases} \\ &\text{for } 1 \leqslant i \leqslant 3, \\ B[x_i, r] &= \begin{cases} \{x_i\} & \text{for } r < 6, \\ \{x_4, x_6\} & \text{for } 6 \leqslant r < 16, \\ S & \text{for } r = 16, \end{cases} \\ &\text{for } 4 \leqslant i \leqslant 5, \\ B[x_i, r] &= \begin{cases} \{x_i\} & \text{for } r < 4, \\ \{x_6, x_7, x_8\} & \text{for } 4 \leqslant r < 16, \\ S & \text{for } r = 16, \end{cases} \\ &\text{for } 6 \leqslant i \leqslant 8. \end{aligned}$$

$\square$

Based on the properties of spheric clusterings mentioned above meaningful such clusterings can be produced in linear time in the number of objects. For the ultrametric space mentioned in Example 2.1.3, the closed spheres of radius 6 produce the clustering

$$\{x_1, x_2, x_3\}, \{x_4, x_5, \}, \{x_6, x_7, x_8\}.$$

If a dissimilarity defined on a data set is close to an ultrametric it is natural to assume that the data set is clusterable. We assess the closeness between a dissimilarity d and a special ultrametric known as the *subdominant ultrametric* of d using a matrix approach.

Let S be a set. Define a partial order "$\leqslant$" on the set of definite dissimilarities $\mathcal{D}_S$ by $d \leqslant d'$ if $d(x,y) \leqslant d'(x,y)$ for every $x, y \in S$. It is easy to verify that $(\mathcal{D}_S, \leqslant)$ is a poset.

The set $\mathcal{U}_S$ of ultrametrics on S is a subset of $\mathcal{D}_S$.

Theorem 2.1.4. *Let $\{d_i \in \mathcal{U}_S \mid i \in I\}$ be a collection of ultrametrics on the set S. Then, the mapping $d : S \times S \longrightarrow \mathbb{R}_{\geqslant 0}$ defined as*

$$d(x,y) = \sup\{d_i(x,y) \mid i \in I\}$$

is an ultrametric on S.

Proof. We need to verify only that $d(x,y)$ satisfies the ultrametric inequality $d(x,y) \leqslant \max\{d(x,z), d(z,y)\}$ for $x, y, z \in S$. Since each mapping d_i is an ultrametric, for $x, y, z \in S$ we have

$$\begin{aligned} d_i(x,y) &\leqslant \max\{d_i(x,z), d_i(z,y)\} \\ &\leqslant \max\{d(x,z), d(z,y)\} \end{aligned}$$

for every $i \in I$. Therefore,

$$\begin{aligned} d(x,y) &= \sup\{d_i(x,y) \mid i \in I\} \\ &\leqslant \max\{d(x,z), d(z,y)\}, \end{aligned}$$

hence d is an ultrametric on S. □

Theorem 2.1.5. *Let d be a dissimilarity on a set S and let U_d be the set of ultrametrics $U_d = \{e \in \mathcal{U}_S \mid e \leqslant d\}$. The set U_d has a largest element in the poset $(\mathcal{U}_S, \leqslant)$.*

Proof. The set U_d is nonempty because the zero dissimilarity d_0 given by $d_0(x, y) = 0$ for every $x, y \in S$ is an ultrametric and $d_0 \leqslant d$.

Since the set $\{e(x, y) \mid e \in U_d\}$ has $d(x, y)$ as an upper bound, it is possible to define the mapping $e_1 : S^2 \longrightarrow \mathbb{R}_{\geq 0}$ as $e_1(x, y) = \sup\{e(x, y) \mid e \in U_d\}$ for $x, y \in S$. It is clear that $e \leqslant e_1$ for every ultrametric e. We claim that e_1 is an ultrametric on S.

We prove only that e_1 satisfies the ultrametric inequality. Suppose that there exist $x, y, z \in S$ such that e_1 violates the ultrametric inequality; that is,

$$\max\{e_1(x, z), e_1(z, y)\} < e_1(x, y).$$

This is equivalent to

$$\begin{aligned} & \sup\{e(x, y) \mid e \in U_d\} \\ & \quad > \max\{\sup\{e(x, z) \mid e \in U_d\}, \\ & \qquad \sup\{e(z, y) \mid e \in U_d\}\}. \end{aligned}$$

Thus, there exists $\hat{e} \in U_d$ such that

$$\hat{e}(x, y) > \sup\{e(x, z) \mid e \in U_d\}$$

and

$$\hat{e}(x, y) > \sup\{e(z, y) \mid e \in U_d\}.$$

In particular, $\hat{e}(x, y) > \hat{e}(x, z)$ and $\hat{e}(x, y) > \hat{e}(z, y)$, which contradicts the fact that $\hat{e}$ is an ultrametric. $\square$

The ultrametric defined by Theorem 2.1.5 is known as the *maximal subdominant ultrametric for the dissimilarity d.*

The situation is not symmetric with respect to the infimum of a set of ultrametrics because, in general, the infimum of a set of ultrametrics is not necessarily an ultrametric.

2.2 The Metric Space of Partitions of a Finite Set

Properties of generalized entropy defined on partition lattices were explored in [74]. Unless stated otherwise all sets are supposed to be finite.

Definition 2.2.1. *A partition of a set S is a non-empty collection of pairwise disjoint and non-empty subsets of S referred to as* blocks, *$\pi = \{B_1, \ldots, B_n\}$ such that $\bigcup_{i=1}^{n} B_i = S$. The set of partitions of a set S is denoted by $\mathsf{PART}(S)$; the set of partitions of S having n blocks is denoted by $\mathsf{PART}_n(S)$.*

Definition 2.2.2. *If $\pi, \sigma \in \mathsf{PART}(S)$, we write $\pi \leqslant \sigma$ if every block of σ is a union of blocks of π. The relation "$\leqslant$" is a* partial order *on $\mathsf{PART}(S)$ having $\iota_S = \{\{x\} \mid x \in S\}$ as its least element and $\omega_S = \{S\}$ as its largest element, so $\iota_S \leqslant \pi \leqslant \omega_S$ for $\pi \in \mathsf{PART}(S)$.*

The partially ordered set $(S, \leqslant)$ is a *lattice*, where $\pi \wedge \sigma = \{B_i \cap C_j \mid B_i \in \pi, C_j \in \sigma \text{ and } B_i \cap C_j \neq \emptyset\}$. The other lattice operation, $\pi \vee \sigma$ has a more complicated description that can be found, for example, in [74].

The partition σ covers the partition π (denoted by $\pi \prec \sigma$) if $\pi \leqslant \sigma$ and there is no partition τ distinct from π and σ such that $\pi \leqslant \tau \leqslant \sigma$. It is known (see [17]) that $\pi \prec \sigma$ if and only if σ is obtained from π by fusing two blocks of π. Of course, if $\pi \leqslant \sigma$, there exists a chain of partitions $\tau_0, \tau_1, \ldots, \tau_n$ such that $\pi = \tau_0$, $\tau_i \prec \tau_{i+1}$ for $0 \leqslant i \leqslant n-1$ and $\tau_n = \sigma$.

If $\pi = \{B_1, \ldots, B_n\} \in \mathsf{PART}(S)$ and $C \subseteq S$, the *trace of π on C* is the partition $\pi_C \in \mathsf{PART}(C)$ given by $\pi_C = \{B_i \cap C \mid B_i \in \pi \text{ and } B_i \cap C \neq \emptyset\}$. Note that we have $\pi \leqslant \sigma$ if and only if $\sigma_B = \omega_B$ for every block B of π.

Definition 2.2.3. *If $\pi = \{B_1, \ldots, B_n\}$ is a partition of a set S and $\beta > 0$, then its* β-entropy *(introduced in [24, 38]), H_β, is given by:*

$$H_\beta(\pi) = \frac{1}{1 - 2^{1-\beta}} \left(1 - \sum_{i=1}^{n} \left(\frac{|B_i|}{|S|}\right)^{\beta}\right). \tag{2.1}$$

It is immediate that $H_\beta(\omega_S) = 0$.

Note that for $\beta = 2$, we obtain the well-known *Gini* index

$$H_2(\pi) = 2\left(1 - \sum_{i=1}^{n} \left(\frac{|B_i|}{|S|}\right)^{2}\right).$$

and when $\beta \to 1$,

$$\lim_{\beta \to 1} H_\beta = -\sum_{i=1}^{n} \frac{|B_i|}{|S|} \log \frac{|B_i|}{|S|}$$

as it can be verified immediately by applying l'Hôpital rule.

Thus, the Shannon entropy denoted by H is a limit case of the generalized entropy. Although in most cases we use the Shannon entropy, the β-entropy is important for determining the number of clusters for imbalanced data sets.

Definition 2.2.4. *Let $h_\beta : [0,1] \longrightarrow \mathbb{R}$. Define*

$$h_\beta(x) = \frac{x - x^\beta}{1 - 2^{1-\beta}},$$

where $\beta > 0$ and $\beta \neq 1$.

Theorem 2.2.5. *h_β is a concave function for $\beta > 0$ and $\beta \neq 1$.*

Proof. Since $h''_\beta(x) = \frac{-\beta(\beta-1)x^{\beta-2}}{1-2^{1-\beta}}$, it follows that $h''_\beta(x) \leqslant 0$ because $1 - 2^{1-\beta} \geqslant 0$ when $\beta \geqslant 1$, and $1 - 2^{1-\beta} \leqslant 0$ when $\beta \leqslant 1$. Therefore, h_β is a concave function for $\beta > 0$ and $\beta \neq 1$. □

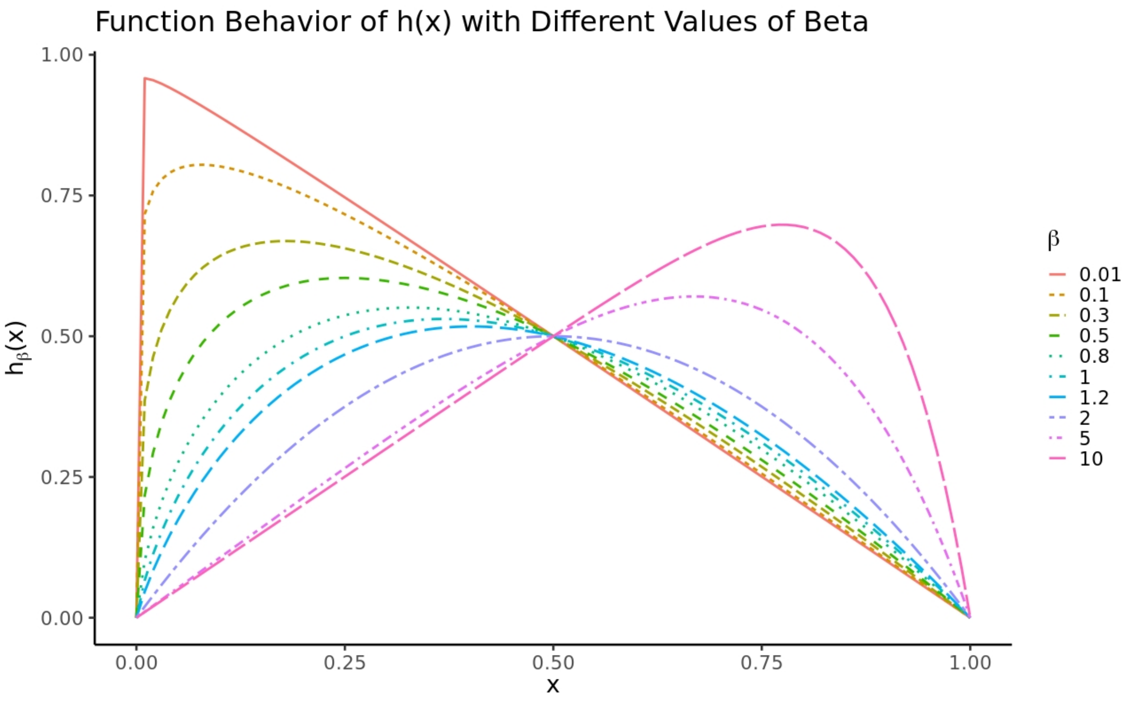


Figure 2.1: *Behavior of Function $h_\beta(x)$ with different β*

We have $h_\beta\left(\frac{1}{2}\right) = \frac{1}{2}$; the maximum of h_β on the $[0,1]$ interval is achieved at $x_\beta = \beta^{-\frac{1}{\beta-1}}$ and equals $\frac{\beta-1}{1-2^{1-\beta}}\beta^{-\frac{\beta}{\beta-1}}$. The behavior of function with different β is shown in Figure 2.1. The function h_β is subadditive for every $\beta \in (0,1) \cup (1,\infty)$, that is,

$$h_\beta(x+y) \leqslant h_\beta(x) + h_\beta(y)$$

for $x, y \in [0,1]$. Observe that $\lim_{\beta\to 1} h_\beta(x) = x \log_2 \frac{1}{x}$.

Since

$$\begin{aligned} H_\beta(\pi) &= \frac{1}{1-2^{1-\beta}}\left(1 - \sum_{i=1}^{n}\left(\frac{|B_i|}{|S|}\right)^\beta\right) \\ &= \sum_{i=1}^{n} h_\beta\left(\frac{|B_i|}{|S|}\right), \end{aligned}$$

the concavity of h_β implies that the maximum value of $H_\beta(\pi)$ is achieved when $|B_1| = \cdots = |B_n|$ and is equal to $\frac{1}{1-2^{1-\beta}}\left(1 - n^{1-\beta}\right)$. Thus, the maximal value of $H_\beta(\pi)$ is obtained when $\pi = \iota_S$ and it is equal to $\frac{1}{1-2^{1-\beta}}\left(1 - |S|^{1-\beta}\right)$. Note that the minimal value of $H_\beta(\pi)$ is achieved when $\pi = \omega_S$, $H_\beta(\omega_S) = 0$ and $H_\beta(\pi) = 0$ implies $\pi = \omega_S$.

Let $\{S_1, \ldots, S_n\}$ be a partition of the set S and let $\pi_1, \ldots, \pi_n$ be n partitions such that $\pi_i \in \mathsf{PART}(S_i)$ for $1 \leqslant i \leqslant n$. Define the partition $\pi_1 + \cdots + \pi_n$ as the partition of S that consists of all blocks of $\pi_1, \ldots, \pi_n$. Then,

$$\begin{aligned} H_\beta(\pi_1 + \ldots + \pi_n) &= H_\beta(\{S_1, \ldots, S_n\}) \\ &\quad + \sum_{i=1}^{n} \left(\frac{|S_i|}{|S|}\right)^\beta H_\beta(\pi_i). \end{aligned}$$

If $\pi = \{B_1, \ldots, B_m\}$ and let $\sigma = \{C_1, \ldots, C_n\}$ are two partitions in $\mathsf{PART}(S)$, then

$$\begin{aligned} H_\beta(\pi \wedge \sigma) &= H_\beta(\sigma) + \sum_{j=1}^{m} \left(\frac{|C_j|}{|S|}\right)^\beta H_\beta(\pi_{C_j}) \\ &= H_\beta(\pi) + \sum_{i=1}^{n} \left(\frac{|B_i|}{|S|}\right)^\beta H_\beta(\sigma_{B_i}). \end{aligned}$$

Definition 2.2.6. *The* conditional β-entropy $H_\beta(\pi|\sigma)$ *is defined as*

$$H_\beta(\pi|\sigma) = H_\beta(\pi \wedge \sigma) - H_\beta(\sigma).$$

The β-entropy is anti-monotonic, that is, for $\beta \in \mathbb{R}_{>0} - \{1\}$ and $\pi, \sigma \in \mathsf{PART}(S)$, $\pi \leqslant \sigma$ implies $H_\beta(\sigma) \leqslant H_\beta(\pi)$. The conditional β-entropy $H_\beta(\pi|\sigma)$ is anti-monotonic in its first argument and monotonic in its second, that is $\pi_1 \leqslant \pi_2$ implies $H_\beta(\pi_1|\sigma) \geqslant H_\beta(\pi_2|\sigma)$ and $\sigma_1 \leqslant \sigma_2$ implies $H_\beta(\pi|\sigma_1) \geqslant H_\beta(\pi|\sigma_2)$.

Theorem 2.2.7. *the function $d_\beta : \mathsf{PART}(S) \times \mathsf{PART}(S) \longrightarrow \mathbb{R}$ defined by*

$$d_\beta(\pi, \sigma) = H_\beta(\pi|\sigma) + H_\beta(\sigma|\pi)$$

is a metric on $\mathsf{PART}(S)$.

The result is obtained and proved in [75] (which is a generalization of a result of [25]). This function will be used to evaluate distance between clusterings regarded as sets of objects.

CHAPTER 3

DATA ULTRAMETRICITY AND CLUSTERABILITY

In this chapter, we introduce two definitions of ultrametricity and utilize the second one to formalize the definition of clusterability. Then we test the validity of our definition on both synthetic and real data sets.

3.1 Introduction

A data set is clusterable if such groups exist; however, due to the variety in data distributions and the inadequate formalization of certain basic notions of clustering, determining data clusterability before applying specific clustering algorithms is a difficult task.

Evaluating data clusterability before the application of clustering algorithms can be very helpful because clustering algorithms are expensive. However, many such evaluations are impractical because they are NP-hard, as shown in [3]. Other notions define data as clusterable when the minimum between-cluster separation is greater than the maximum intra-cluster distance [28], or when each element is closer to all elements in its cluster than to all other data [12].

Several approaches exist in assessing data clusterability. The main hypothesis of [1] is that clusterability can be inferred from an one-dimensional view of pairwise distances between objects. Namely, clusterability is linked to the multimodality of the histogram of inter-object dissimilarities. The basic assumption is that "the presence of multiple modes in the set of pairwise dissimilarities indicates that the original data is clusterable." Multimodality is evaluated using the *Dip* and *Silverman* statistical multimodality tests, an approach that is computationally efficient.

Alternative approaches to data clusterability are linked to the feasibility of producing a clustering; a corollary of this assumption is that “data that are hard to cluster do not have a meaningful clustering structure” [23]. Other approaches to clusterability are identified based on clustering quality measures, and on loss function optimization [3, 15, 2, 14, 12, 16].

We propose a novel approach that relates data clusterability to the extent to which the dissimilarity defined on the data set relate to a special ultrametric defined on the set.

This chapter is structured as follows. In Section 3.2, we introduced a definition of ultrametricity by measuring the power of distance value. After it, a faster version of calculating this ultrametricity is also given in Section 3.3. The next section(Section 3.4) shows a special matrix product on matrices with non-negative elements that allow an efficient computation of the subdominant ultrametric. We utilize the number of product to reach ultrametric as the other measure of ultrametricity. In Section 3.5, resort to the concept of ultrametricity in previous section, a measure of clusterability that is based on the iterative properties of the dissimilarity matrix is defined. We provide experimental evidence on the effectiveness of the proposed measure through several experiments on small artificial data sets in Section 3.6. Finally, we present our conclusions and future plans in Section 3.7.

3.2 Measure the Ultrametricity of Dissimilarity

Let r be a non-negative number and let $\mathcal{D}_r(S)$ be the set of dissimilarities defined on a set S that satisfy the inequality $d(x,y)^r \leqslant d(x,z)^r + d(z,y)^r$ for $x, y, z \in S$. Note that every dissimilarity belongs to the set $\mathcal{D}_0$; a dissimilarity in $\mathcal{D}_1$ is a quasi-metric.

Theorem 3.2.1. *Let (S,d) be a dissimilarity space and let $\mathcal{D}_\infty(S) = \bigcap_{r \geqslant 0} \mathcal{D}_r(S)$. If $d \in \mathcal{D}_\infty(S)$, then d is an ultrametric on S.*

Proof. Let $d \in \mathcal{D}_\infty$ and let $t = xyz$ be a triangle in the dissimilarity space (S, d). Assume that $d(x, y) \geqslant d(x, z) \geqslant d(z, y)$.

Suppose intially that $d(x, z) = d(y, z)$. Then, $d \in \mathcal{D}_r(S)$ implies that $d(x, y)^r \leqslant 2d(x, z)^r$, so

$$\left(\frac{d(x, y)}{d(x, z)}\right)^r \leqslant 2$$

for every $r \geqslant 0$. By taking $r \to \infty$ it is clear that this is possible only if $d(x, y) \leqslant d(x, z)$, which implies $d(x, y) = d(x, z) = d(y, z)$; in other words, t is an equilateral triangle.

The alternative supposition is that $d(x, z) > d(y, z)$. Again, since $d \in \mathcal{D}_r(S)$, it follows that

$$\begin{aligned} d(x, y) &\leqslant (d(x, z)^r + d(z, y)^r)^{\frac{1}{r}} \\ &= d(x, z)\left(1 + \left(\frac{d(z, y)}{d(x, z)}\right)^r\right)^{\frac{1}{r}} \end{aligned}$$

for every $r > 0$. Since $\lim_{r\to\infty} d(x, z)\left(1 + \left(\frac{d(y,z)}{d(x,z)}\right)^r\right)^{\frac{1}{r}} = d(x, z)$, it follows that $d(x, y) \leqslant d(x, z)$ for $x, y, z \in S$. This inequality implies $d(x, y) = d(x, z)$, so the largest two sides of the triangle xyz are equal. This allows us to conclude that d is an ultrametric. □

It is easy to verify that if r and s are positive numbers, then $r \leqslant s$ implies $(d(x, z)^r + d(z, y)^r)^{\frac{1}{r}} \geqslant (d(x, z)^s + d(z, y)^s)^{\frac{1}{s}}$ (see [74], Lemma 6.15). Thus, if $r \leqslant s$ we have the inclusion $\mathcal{D}_s \subseteq \mathcal{D}_r$.

Let d and d' be two dissimilarities defined on a set S. We say that d' dominates d if $d(x, y) \leqslant d'(x, y)$ for every $x, y \in S$. The pair $(\mathsf{DISS}(S), \leqslant)$ is a partially ordered set.

Let r, s be two positive numbers such that $r < s$, and let $d \in \mathcal{D}_r(S)$. The family $\mathcal{D}_{s,d}(S)$ of s-dissimilarities on S that are dominated by d has a largest element.

Indeed, since every element of $\mathcal{D}_{s,d}(S)$ is dominated by d, we can define the mapping $\tilde{e} : S \times S \longrightarrow \mathbb{R}_{\geqslant 0}$ as $\tilde{e}(x, y) = \sup\{e(x, y) \mid e \in \mathcal{D}_{s,d}(S)\}$. It is immediate that e is a dissimilarity on S and that $\tilde{e} \leqslant d$. Moreover, we have $e(x, y)^s \leqslant e(x, z)^s + e(z, y)^s \leqslant \tilde{e}(x, z)^s + \tilde{e}(z, y)^s$ for every $x, y, z \in S$, which implies

$$\tilde{e}(x, y)^s \leqslant \tilde{e}(x, z)^s + < \tilde{e}(z, y)^s.$$

Thus, $\tilde{e} \in \mathcal{D}_{s,d}(S)$, which justified our claim.

For $r > 0$ define the function $F_r : \mathbb{R}^2_{geqs0} \longrightarrow \mathbb{R}_{\geqslant 0}$ as $F_r(a, b) = (a^r + b^r)^{\frac{1}{r}}$. It is straighforward to see that $p \geqslant q$ implies $F_p(a, b) \leqslant F_q(a, b)$ for $a, b \in \mathbb{R}_{\geqslant 0}$. Furthermore for $r > 0$ we have $d \in \mathcal{D}_r(S)$ if and only if $d(x, y) \leqslant F_r(d(x, z), d(z, y))$.

Definition 3.2.2. Let r, s be two positive numbers. An (r, s)-transformation is a function $g : \mathbb{R}_{\geqslant 0} \longrightarrow \mathbb{R}_{\geqslant 0}$ such that

(i) $g(x) = 0$ if and only if $x = 0$;

(ii) g is continuous and strictly monotonic on $\mathbb{R}_{\geqslant 0}$;

(iii) $g(F_r(a, b)) \leqslant F_s(g(a), g(b))$ for $a, b \in \mathbb{R}_{\geqslant 0}$.

□

Note that if $d \in \mathcal{D}_r(S)$ and g is an (r, s)-transformation, then $gd \in \mathcal{D}_s(S)$.

3.3 A Weaker Dissimilarity Measure

The notion of weak ultrametricity that we are about to introduce has some computational advantages over the notion of ultrametricity, especially from the point of view of handling transformations of metrics.

Let (S, d) be a dissimilarity space and let $t = xyz$ be a triangle. Following Lerman's notation [47], we write $S_d(t) = d(x, y), M_d(t) = d(x, z),$ and $L_d(t) = d(y, z)$, if $d(x, y) \geqslant d(x, z) \geqslant d(y, z)$.

Definition 3.3.1. Let (S, d) be a dissimilarity space and let $t = xyz \in S^3$ be a triangle.

The *ultrametricity* of t is the number $u_d(t)$ defined by

$$u_d(t) = \max\{r > 0 \mid S_d(t)^r \leqslant M_d(t)^r + L_d(t)^r\},$$

which is the ultrametricity of the subspace $(\{x, y, z\}, d)$ of (S, d). If $d \in \mathcal{D}_p$, we have $p \leqslant u_d(t)$ for every $t \in S^3$.

The *weak ultrametricity* of the triangle t, $w_d(t)$, is given by

$$w_d(t) = \begin{cases} \frac{1}{\log_2 \frac{S_d(t)}{M_d(t)}} & \text{if } S_d(t) > M_d(t) \\ \infty & \text{if } S_d(t) = M_d(t). \end{cases}$$

If $w_d(t) = \infty$, then t is an *ultrametric triple.*

The *weak ultrametricity* of the dissimilarity space (S, d) is the number $w(S, d)$ defined by

$$w(S, d) = \texttt{median}\{w_d(t) \mid t \in S^3\}.$$

$\square$

The definition of $w(S, d)$ eliminates the influence of triangles whose ultrametricity is an outlier, and gives a better picture of the global ultrametric property of (S, d).

For a triangle t we have

$$0 \leqslant S_d(t) - M_d(t) = \left(2^{\frac{1}{w_d(t)}} - 1\right) M_d(t) \leqslant \left(2^{\frac{1}{w(S,d)}} - 1\right) M_d(t)$$

Thus, if $w_d(t)$ is sufficiently large, the triangle t is almost isosceles. For example, if $w_d(t) = 5$, the difference between the length of longest side $S_d(t)$ and the median side $M_d(t)$ is less than 15%.

For every triangle $t \in S^3$ in a dissimilarity space we have $u_d(t) \leqslant w_d(t)$. Indeed, since $S_d(t)^{u_d(t)} \leqslant M_d(t)^{u_d(t)} + L_d(t)^{u_d(t)}$ we have $S_d(t)^{u_d(t)} \leqslant 2M_d(t)^{u_d(t)}$, which is equivalent to $u_d(t) \leqslant w_d(t)$.

Next we discuss dissimilarity transformations that impact the ultrametricity of dissimilarities.

Theorem 3.3.2. *Let (S, d) be a dissimilarity space and let $f : \mathbb{R}_{\geqslant 0} \longrightarrow \mathbb{R}_{\geqslant 0}$ be a function that satisfies the following conditions:*

(i) *$f(0) = 0$;*

(ii) *f is increasing;*

(iii) *the function $g : \mathbb{R}_{\geqslant 0} \longrightarrow \mathbb{R}_{\geqslant 0}$ given by*

$$g(a) = \begin{cases} \frac{f(a)}{a} & \text{if } a > 0, \\ 0 & \text{if } a = 0 \end{cases}$$

is decreasing.

Then the function $e : S \times S \longrightarrow \mathbb{R}_{\geqslant 0}$ defined by $e(x, y) = f(d(x, y))$ for $x, y \in S$ is a dissimilarity and $w_d(t) \leqslant w_e(t)$ for every triangle $t \in S^3$.

Proof. Let $t = xyz \in S^3$ be a triangle. It is immediate that $e(x, y) = e(y, x)$ and $e(x, x) = 0$.

Since f is an increasing function we have $f(S_d(t)) \geqslant f(M_d(t)) \geqslant f(L_d(t))$, so the ordering of the sides of the tranformed triangle is preserved.

Since g is a decreasing function, we have $g(S_d(t)) \leqslant g(M_d(t))$, that is, $\frac{f(S_d(t))}{S_d(t)} \leqslant \frac{f(M_d(t))}{M_d(t)}$, or

$$\frac{S_d(t)}{M_d(t)} \geqslant \frac{f(S_d(t))}{f(M_d(t))}.$$

Therefore,

$$w_d(t) = \frac{1}{\log_2 \frac{S_d(t)}{M_d(t)}} \leqslant \frac{1}{\log_2 \frac{S_e(t)}{M_e(t)}} = w_e(t).$$

□

Example 3.3.3. Let (S, d) be a dissimilarity space and let e be the dissimilarity defined by $e(x, y) = d(x, y)^r$, where $0 < r < 1$. If $f(a) = a^r$, then f is increasing and $f(0) = 0$. Furthermore the function $g : \mathbb{R}_{\geqslant 0} \longrightarrow \mathbb{R}_{\geqslant 0}$ given by

$$g(a) = \begin{cases} \frac{f(a)}{a} & \text{if } a > 0, \\ 0 & \text{if } a = 0 \end{cases} = \begin{cases} a^{r-1} & \text{if } a > 0, \\ 0 & \text{if } a = 0 \end{cases}$$

is decreasing. Therefore, we have $w_e(t) \geqslant w_d(t)$. □

Example 3.3.4. Let $f : \mathbb{R}_{\geqslant 0} \longrightarrow \mathbb{R}_{\geqslant 0}$ be defined by $f(a) = \frac{a}{a+1}$. It is easy to see that f is increasing on $\mathbb{R}_{\geqslant 0}$, $f(0) = 0$, and

$$g(a) = \begin{cases} \frac{1}{1+a} & \text{if } a > 0, \\ 0 & \text{if } a = 0 \end{cases}$$

is decreasing on the same set. Therefore, the weak ultrametricity of a triangle increases when d is replaced by e given by

$$e(x, y) = \frac{d(x, y)}{1 + d(x, y)}$$

for $x, y \in S$. □

Example 3.3.5. For a dissimilarity space (S, d), the Schoenberg transform of d described in [26] is the dissimilarity $e : S^2 \longrightarrow \mathbb{R}_{\geqslant 0}$ defined by

$$e(x, y) = 1 - e^{-kd(x,y)}$$

for $x, y \in S$. Let $f : \mathbb{R}_{\geqslant 0} \longrightarrow \mathbb{R}_{\geqslant}$ be the function $f(a) = 1 - e^{-ka}$ that is used in this transformation. It is immediate that f is a increasing function and $f(0) = 0$. For $a > 0$ we have $g(a) = \frac{1-e^{-ka}}{a}$, which allows us to write

$$g'(a) = \frac{e^{-ka}(ka + 1) - 1}{a^2}$$

for $a > 0$. Taking into account the obvious inequality $ka + 1 < e^{ka}$ for $k > 0$, it follows that the function g is decreasing. Thus, the weak ultrametricity of a triangle relative to the Schoenberg transform is greater than the weak ultrametricity under the original dissimilarity. □

3.4 Matrices Product and Ultrametricity

Let $\mathbb{P}$ be the set

$$\mathbb{P} = \{x \mid x \in \mathbb{R}, x \geqslant 0\} \cup \{\infty\}.$$

The usual operations defined on $\mathbb{R}$ can be extended to $\mathbb{P}$ by defining

$$x + \infty = \infty + x = \infty, x \cdot \infty = \infty \cdot x = \infty$$

for $x \geqslant 0$.

Let $\mathbb{P}^{m \times n}$ be the set of $m \times n$ matrices over $\mathbb{P}$. If $A, B \in \mathbb{P}^{m \times n}$ we have $A \leqslant B$ if $a_{ij} \leqslant b_{ij}$ that is, if $a_{ij} \geqslant b_{ij}$ for $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant n$.

If $A \in \mathbb{P}^{m\times n}$ and $B \in \mathbb{P}^{n\times p}$ the matrix product $C = AB \in \mathbb{P}^{m\times p}$ is defined as:

$$c_{ij} = \min\{\max\{a_{ik}, b_{kj}\} \mid 1 \leqslant k \leqslant n\},$$

for $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant p$.

If $E_n \in \mathbb{P}^{n\times n}$ is the matrix defined by

$$(E_n)_{ij} = \begin{cases} 0 & \text{if } i = j, \\ \\ \infty & \text{otherwise,} \end{cases}$$

that is the matrix whose main diagonal elements are 0 and the other elements equal ∞, then $AE_n = A$ for every $A \in \mathbb{P}^{m\times n}$ and $E_nA = A$ for every $A \in \mathbb{P}^{n\times p}$.

The matrix multiplication defined above is associative, hence $\mathbb{P}^{n\times n}$ is a semigroup with the identity E_n. The powers of A are inductively defined as

$$\begin{aligned} A^0 &= E_n, \\ A^{n+1} &= A^nA, \end{aligned}$$

for $n \in \mathbb{N}$.

For $A, B \in \mathbb{P}^{m\times n}$ we define $A \leqslant B$ as $a_{ij} \leqslant B_{ij}$ for $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant n$. Note that if $A \in \mathbb{P}^{n\times n}$, then $A \leqslant E_n$. It is immediate that for $A, B \in \mathbb{P}^{m\times n}$ and $C \in \mathbb{P}^{n\times p}$, then $A \leqslant B$ implies $AC \leqslant BC$; similarly, if $C \in \mathbb{P}^{p\times m}$ and $CA \leqslant CB$.

Let $L(A)$ be the finite set of elements in $\mathbb{P}$ that occur in the matrix $A \in \mathbb{P}^{n\times n}$. Since he entries of any power A^n of A are also included in $L(A)$, the sequence $A, A^2, \ldots, A^n, \ldots$ is ultimately periodic because it contains a finite number of distinct matrices.

Let $k(A)$ be the least integer k such that $A^k = A^{k+d}$ for some $d > 0$. The sequence of powers of A has the form

$$A, A^2, \ldots, A^{k(A)-1}, A^{k(A)}, \ldots,$$
$$A^{k(A)+d-1}, A^{k(A)}, \ldots, A^{k(A)+d-1}, \ldots,$$

where d is the least integer such that $A^{k(A)} = A^{k(A)+d}$. This integer is denoted by $d(A)$.

The set $\{A^{k(A)}, \ldots, A^{k(A)+d-1}\}$ is a cyclic group with respect to the multiplication.

If (S, d) is a dissimilarity space, where $S = \{x_1, \ldots, x_n\}$, the matrix of this space is the matrix $A \in \mathbb{P}^{n\times n}$ defined by $a_{ij} = d(x_i, x_j)$ for $1 \leqslant i, j \leqslant n$. Clearly, A is a symmetric matrix and all its diagonal elements are 0, that is, $A \leqslant E_n$.

If, in addition, we have $a_{ij} \leqslant a_{ik} + a_{kj}$ for $1 \leqslant i, j, k \leqslant n$, then A is a *metric matrix*. If this condition is replaced by the stronger condition $a_{ij} \leqslant \max\{a_{ik} + a_{kj}\}$ for $1 \leqslant i, j, k \leqslant n$, then A is *ultrametric matrix*. Thus, for an ultrametric matrix we have $a_{ij} \leqslant \min\{\max\{a_{ik} + a_{kj}\} \mid 1 \leqslant k \leqslant n\}$. This amounts to $A^2 \leqslant A$.

Theorem 3.4.1. *If $A \in \mathbb{P}^{n\times n}$ is a dissimilarity matrix there exists $m \in \mathbb{N}$ such that*

$$\cdots = A^{m+1} = A^m \leqslant \cdots \leqslant A^2 \leqslant A \leqslant E_n$$

and A^m is an ultrametric matrix.

Proof. Since $A \leqslant E_n$, the existence of the number m with the property mentioned in the theorem is immediate since there exists only a finite number of $n \times n$ matrices whose elements belong to $L(A)$. Since $A^m = A^{2m}$, it follows that A^m is an ultrametric matrix. □

For a matrix $A \in \mathbb{P}^{n\times n}$ let $m(A)$ be the least number m such that $A^m = A^{m+1}$. We refer to $m(A)$ as the *stabilization power* of the matrix A. The matrix $A^{m(A)}$ is denoted by A^*.

The previous considerations suggest defining the *ultrametricity* of a matrix $A \in \mathbb{P}^{n\times n}$ with $A \leqslant E_n$ as $u(A) = \frac{n}{m(A)}$. Since $m(A) \leqslant n$, it follows that $u(A) \geqslant 1$. If $m(A) = 1$, A is ultrametric itself and $u(A) = n$.

Theorem 3.4.2. *Let (S, d) be a dissimilarity space, where $S = \{x_1, \ldots, x_n\}$ having the dissimilarity matrix $A \in \mathbb{P}^{n\times n}$. If m is the least number such that $A^m = A^{m+1}$, then the mapping $\delta : S \times S \longrightarrow \mathbb{P}$ defined by $\delta(x_i, x_j) = (A^m)_{ij}$ is the subdominant ultrametric for the dissimilarity d.*

Proof. As we observed, A^m is an ultrametric matrix, so δ is an ultrametric on S. Since $A^m \leqslant A$, it follows that $d(x_i, x_j) \geqslant \delta(x_i, x_j)$ for all $x_i, x_j \in S$.

Suppose that $C \in \mathbb{P}^{n\times n}$ is an ultrametric matrix such that $A \leqslant C$, which implies $A^m \leqslant C^m \leqslant C$. Thus, A^m dominates any ultrametric that is dominated by d. Consequently, the dissimilarity defined by A^m is the subdominant ultrametric for d. □

The subdominant ultrametric of a dissimilarity is usually studied in the framework of weighted graphs [46].

A *weighted graph* is a triple (V, E, w), where V is the set of vertices of G, E is a set of two-element subsets of V called edges. and $w : E \longrightarrow \mathbb{P}$ is the weight of the edges. If $e \in E$, then $e = \{u, v\}$, where u, v are distinct vertices in V. The weight is extended to all 2-elements subsets of V as

$$w(\{v_i, v_j\}) = \begin{cases} w(\{v_i, v_j\}) & \text{if } \{v_i, v_j\} \in E, \\ \infty & \text{otherwise.} \end{cases}$$

A *path of length* n in a weighted graph is a sequence

$$\wp = (v_0, v_1, , v_2, \ldots, v_{n-1}, v_n),$$

where $\{v_i, v_{i+1}\} \in E$ for $0 \leqslant n \leqslant n - 1$.

The set of paths of length n in the graph G is denoted as $\mathsf{Paths}^n(G)$. The set of paths of length n that join the vertex v_i to the vertex v_j is denoted by Paths^n_{ij}. The set of all paths is

$$\mathsf{Paths}(G) = \bigcup_{n \geqslant 1} \mathsf{Paths}^n(G).$$

For a weighted graph $G = (V, E, w)$, the extension of the weight function w to $\mathsf{Paths}^n(G)$ is the function $M : \mathsf{Paths}(G) \longrightarrow \mathbb{P}$ defined as

$$M(\wp) = \max\{w(v_{i-1}, v_i) \mid 1 \leqslant i \leqslant n\},$$

where $\wp = (v_0, v_1, \ldots, v_n)$. Thus, if $\wp' = \wp e$, we have $M(\wp') = \max\{M(\wp), w(e)\}$.

If $G = (V, E, w)$ is a weighted graph, its *incidence matrix* is the matrix $A_G \in \mathbb{P}^{n \times n}$, where $n = |V|$, defined by $(A_G)_{ij} = w(v_i, v_j)$ for $1 \leqslant i, j \leqslant n$.

Let $P^{(\ell)}_{ij}$ be the set of paths of length ℓ that join the vertex v_i to the vertex v_j. Note that

$$\begin{aligned} P^{(\ell+1)}_{ij} &= \{(v_i, \ldots, v_k, v_j) \mid \\ &\quad \wp = (v_i, \ldots, v_k) \in P^{(\ell)}_{ik} \text{ and} \\ &\quad v_j \text{ does not occur in } \wp\}. \end{aligned}$$

Define $a^{(\ell)}_{ij} = \min\{M(\wp) \mid \wp \in P^{(\ell)}_{ij}\}$. The powers of the incidence matrix of the graph are given by

$$\begin{aligned} a^{(\ell+1)}_{ik} &= \min\{M(\wp') \mid \wp' \in P^{(\ell+1)}_{ik}\} \\ &= \min\{\max\{M(\wp), w(e)\} \mid \\ &\quad \wp' = (v_i, \ldots, v_j, v_k) \text{ and} \\ &\quad \wp \in P^{(\ell)}_{ij}, e = (v_j, v_k) \in E\} \\ &= \min_j\{\max\{a^{\ell}_{ij}, w(e)\} \mid e = (v_j, v_k)\}. \end{aligned}$$

Thus, we have

$$(A_G^\ell)_{ij} = \min\{M(\wp) \mid \wp \in P_{ij}^\ell\}$$

for $1 \leqslant i, j \leqslant n$.

3.5 A Measure of Clusterability

We conjecture that a dissimilarity space (D, d) is more clusterable if the dissimilarity is closer to an ultrametric, hence if $m(A_D)$ is small. Thus, it is natural to define the *clusterability of a data set* D as the number $\mathsf{clust}(D) = \frac{n}{m(A_D)}$ where $n = |D|$, A_D is the dissimilarity matrix of D and $m(A_D)$ is the stabilization power of A_D. The lower the stabilization power, the closer A is to an ultrametric matrix, and thus, the higher the clusterability of the data set.

Table 3.1: All clusterable datasets have values greater than 5 for their clusterability; all non-clusterable datasets have values no larger than 5.

Dataset	n	Dip	Silv.	$m(A_D)$	$\mathsf{clust}(D)$
iris	150	0.0000	0.0000	14	10.7
swiss	47	0.0000	0.0000	6	7.8
faithful	272	0.0000	0.0000	31	8.7
rivers	141	0.2772	0.0000	22	6.4
trees	31	0.3460	0.3235	7	4.4
USAJudgeRatings	43	0.9938	0.7451	10	4.3
USArrests	50	0.9394	0.1897	15	3.3
attitude	30	0.9040	0.9449	6	5
cars	50	0.6604	0.9931	15	3.3

Our hypothesis is supported by previous results obtained in [1], where the clusterability of 9 databases were statistically examined using the Dip and Silverman tests of unimodality. The approach used in [1] starts with the hypothesis that the presence of multiple modes in the uni-dimensional set of pairwise distances indicates that the original data set is clusterable. Multimodality is assessed using the tests mentioned above. The time required by this evaluation is quadratic in the number of objects.

The first four data sets, *iris*, *swiss*, *faithful* and *rivers* were deemed to be clusterable; the last five were evaluated as not clusterable. Tests published in [5] have produced low p-values for the first four datasets, which is an indication of clusterability. The last five data sets, *USArrests*, *attitude*, *cars*, and *trees* produce much larger p-values, which show a lack of clusterability. Table 3.1 shows that all data sets deemed clusterable by the unimodality statistical test have values of the clusterability index that exceed 5.

In our approach clusterability of a data set D is expressed primarily through the "stabilization power" $m(A_D)$ of the dissimilarity matrix A_D; in addition, the histogram of the dissimilarity values is less differentiated when the data is not clusterable.

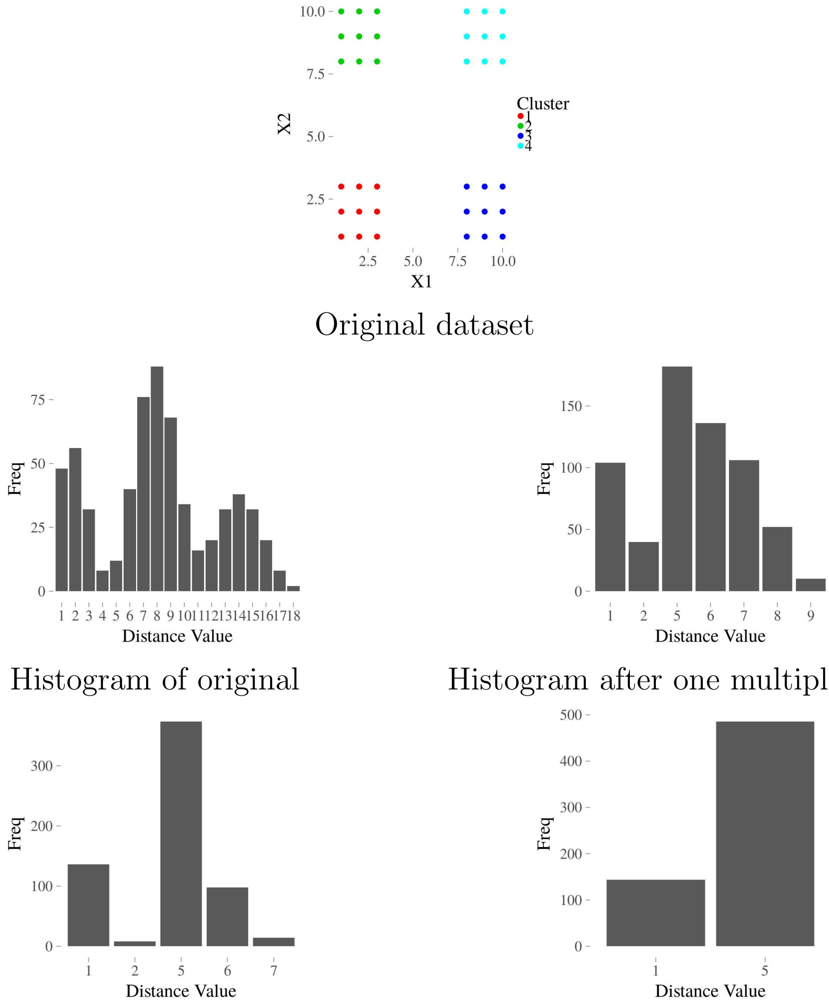
3.6 Experimental Evidence on Small Artificial Data Sets

Another series of experiments involved a series of small datasets having the same number of points in $\mathbb{R}^2$ arranged in lattices. The points have integer coordinates and the distance between points is the Manhattan distance.

By shifting the data points to different locations, we create several distinct structured clusterings that consists of rectangular clusters.

Figures 3.2 and 3.3 show an example of a series of datasets with a total of 36 data points. Initially, the data set has 4 rectangular clusters containing 9 data points each with a gap of 3 distance units between the clusters. The ultrametricity of the dataset and, therefore, its clusterability is affected by the number of clusters, the size of the clusters, and the inter-cluster distances. Figure 3.3 shows that $m(A)$ reaches its highest value and, therefore, the clusterability is the lowest, when there is only one cluster in the dataset (see the third row of Figure 3.3).

If points are uniformly distributed, as it is the case in the third row of Figure 3.3, the clustering structure disappears and $\mathsf{clust}(D)$ has the lowest value.



Original dataset

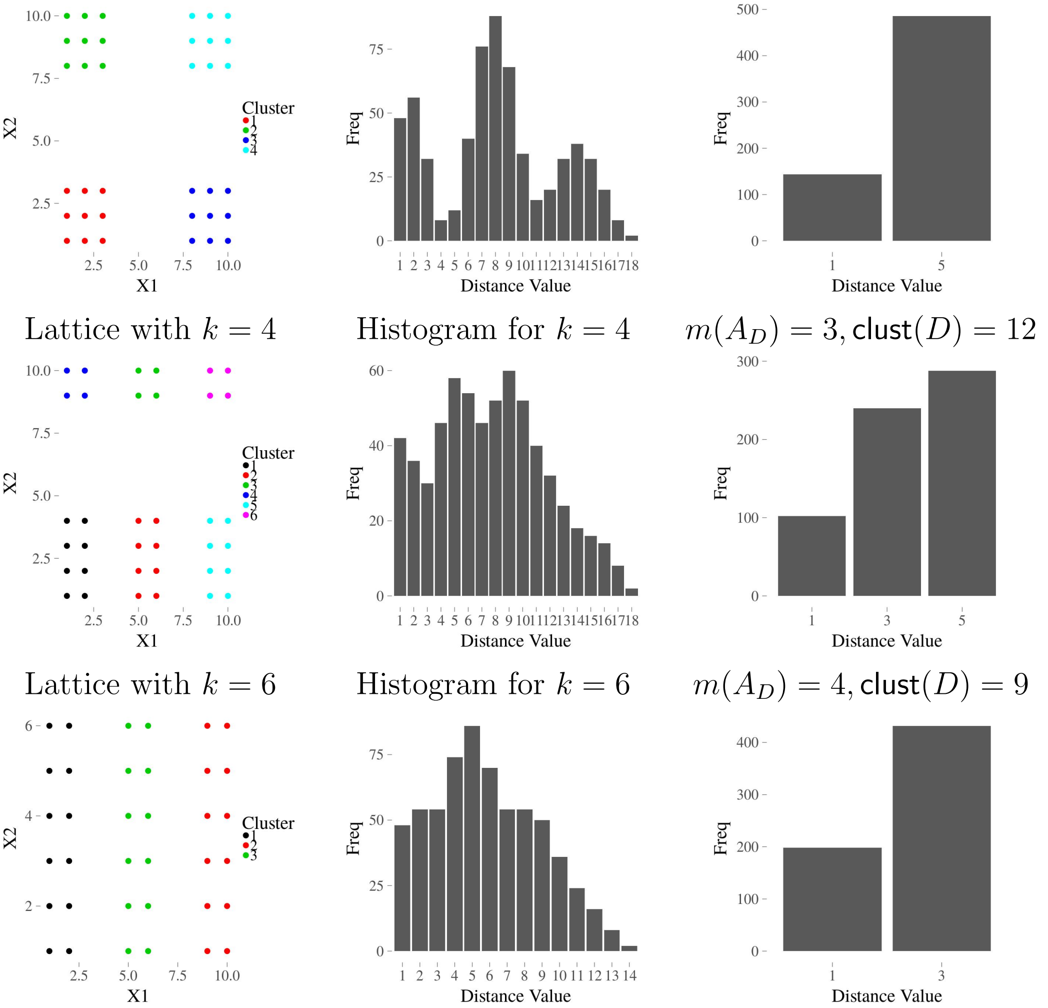
Histogram of original

Histogram after one multiplication

Histogram after two multiplications

Histogram after three multiplications

Figure 3.1: The process of distance equalization for successive powers of the incidence matrix. The matrix A_D^3 is ultrametric.

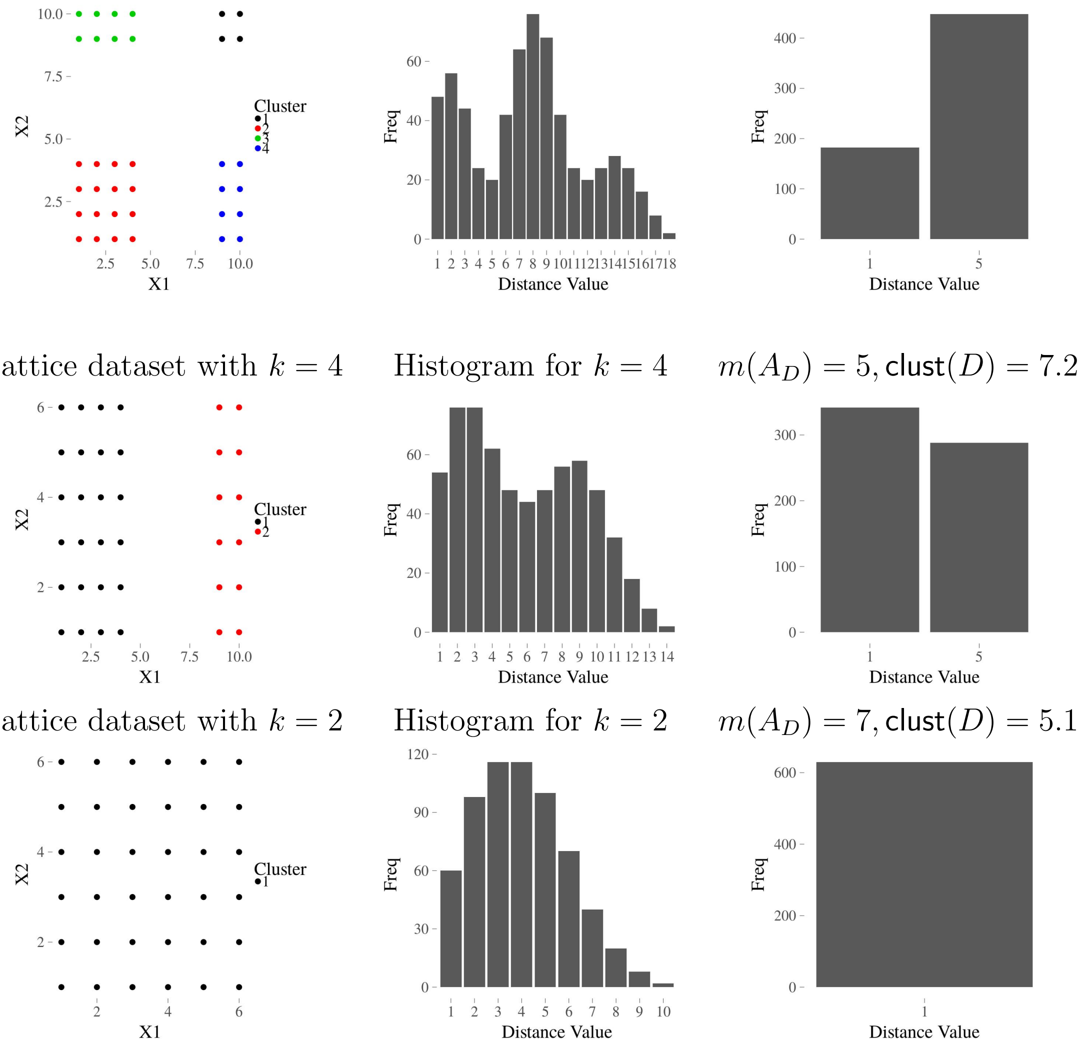


Lattice with $k = 4$ Histogram for $k = 4$ $m(A_D) = 3, \mathsf{clust}(D) = 12$

Lattice with $k = 6$ Histogram for $k = 6$ $m(A_D) = 4, \mathsf{clust}(D) = 9$

Lattice with $k = 3$ Histogram for $k = 3$ $m(A_D) = 5, \mathsf{clust}(D) = 7.2$

Figure 3.2: Cluster separation and clusterability.

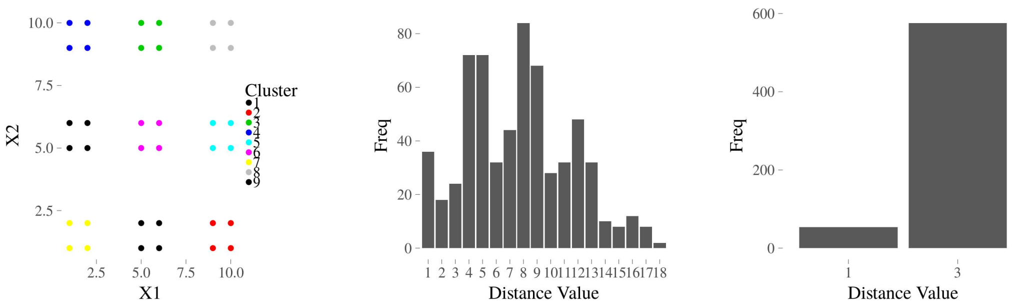


Lattice dataset with $k = 4$ Histogram for $k = 4$ $m(A_D) = 5, \mathsf{clust}(D) = 7.2$

Lattice dataset with $k = 2$ Histogram for $k = 2$ $m(A_D) = 7, \mathsf{clust}(D) = 5.1$

Lattice dataset with $k = 1$ Histogram for $k = 1$ $m(A_D) = 9, \mathsf{clust}(D) = 4$

Figure 3.3: Cluster separation and clusterability (continued).



Lattice dataset with $k = 9$ $k = 9$ $m(A_D) = 6, \mathsf{clust}(D) = 6$

Figure 3.4: Further examples of data sets and their clusterability.

Histograms are used by some authors [3, 15] to identify the degree of clusterability. Note however that in the case of the data shown in Figures 3.2 and 3.3, the histograms of original dissimilarity of the dataset do not offer guidance on the clusterability(second column of Figure 3.2 and 3.3). By applying the "min-max" power operation on the original matrix, we get an ultrametric matrix. The new histogram of the ultrametric shows a clear difference on each dataset. In the third column of Figures 3.2 and 3.3, the histogram of the ultrametric matrix for each dataset shows a decrease of the number of distinct distances after the "power" operation.

If the dataset has no clustering structure the histogram of the ultrametric distance has only one bar.

The number of pics p of the histogram indicate the minimum number of clusters k in the ultrametric space specified by the matrix A^* using the equality $\binom{k}{2} = p$, so the number of clusters is $\left\lceil \frac{1+\sqrt{1+8p}}{2} \right\rceil$. The largest k values of valleys of the histogram indicate the radii of the spheres in the ultrametric space that define the clusters.

If a data set contains a large number of small clusters, these clusters can be regarded as outliers and the clusterability of the data set is reduced. This is the case in the third line of Figure 3.4 which shows a particular case for 9 clusters with 36 data points. Since the size of each cluster is too small to be considered as a real cluster, all of them together are merely regarded as a one cluster dataset with 9 points.

3.7 Conclusions and Future Work

The special matrix powers of the adjacency matrix of the weighted graph of object dissimilarities provide a tool for computing the subdominant ultrametric of a dissimilarity and an assessment of the existence of an underlying clustering structure in a dissimilarity space.

The “power” operation successfully eliminates the redundant information in the dissimilarity matrix of the dataset but maintains the useful information that can discriminate the cluster structures of the dataset.

In a series of seminal papers[62, 63, 64], F. Murtagh argued that as the dimensionality of a linear metric space increases, an equalization process of distances takes place and the metric of the space gets increasingly closer to an ultrametric. This raises the issues related to the comparative evaluation (statistical and algebraic) of the ultrametricity of such spaces and of their clusterability, which we intend to examine in the future.

CHAPTER 4

CLUSTERING WITH ULTRAMETRIC-ADJUSTED DISSIMILARITY MATRIX

In this chapter, we modify the dissimilarity matrix in order to change its ultrametricity, depending on which of the definitions we proposed in Chapter 3 we use. For the first definition of ultrametricity, we evaluate the variation in compactness and separation of clustering results as we increase or decrease the ultrametricity of the dissimilarity matrix. We show that a modification of the ultrametricity of the dissimilarity matrix can improve the clustering quality of some data sets. The other definition of ultrametricity which is measured through the special matrix product is also evaluated on some difficult data sets. We demonstrate that by increasing the ultrametricity of the dissimilarity matrix, we can improve the performances of some traditional clustering algorithms compared to their performances on the original dissimilarity matrix.

4.1 Improved Clustering Results with Altered Dissimilarity Matrix

Clustering validation evaluates and assesses the goodness of the results of a clustering algorithm [58]. We used internal validation measures that rely on information in the data [78], namely and compactness and separation [78, 87].

Compactness measures quantify how well-related the objects in a cluster are. It provides information about the cohesion of objects in an individual cluster with respect to the other objects outside the cluster. A group of measures evaluate cluster compactness based on variance where lower values indicate better compactness. Other measures are based on distance, such as maximum or average pairwise distance, and maximum or average center-based distance.

Separation is a measure of distinctiveness between a cluster and the rest of the world. The pairwise distances between cluster centers or the pairwise minimum distances between objects in different clusters are often used as measures of separation.

The compactness of each cluster was evaluated using the average dissimilarity between the observations in the cluster and the medoid of the cluster. Separation was computed using the minimal dissimilarity between an observation of the cluster and an observation of another cluster.

Based on the definition of ultrametricity in Section 3.2 of Chapter 3, we investigate the impact of this ultrametricity on compactness and separation of clusters by using the Partition Around Medoids (PAM) algorithm [43] to cluster objects originally in the Euclidean Space and later in a transformed dissimilarity space with lower or higher ultrametricity.

Experiments show that a transformation on the distance matrix that decreases the ultrametricity of the original Euclidean space can actually improve compactness but also decrease separation of the clusters generated by PAM. However, the compactness improves at a faster ratio than the decrease in separation. We also observed that the increase of ultrametricity produces the reverse effect, degrading compactness and increasing separation, at different ratios. In this case, compactness decreases in a faster ratio than the increase in separation.

Let (S, d) be a dissimilarity space, (S, d') be the transformed dissimilarity space, where $d' = f(d)$ is obtained by applying one of the transformations described in Section 3.2 and let u and u' be the weak ultrametricities of these two dissimilarity spaces, respectively.

The increase of ultrametricity from (S, d) to (S, d') promotes the equalization of dissimilarity values. In the extreme case, we have an ultrametric space where the pairwise distances involved in all triplets of points form an equilateral or isosceles triangle. To explore how the equalization (or the reverse process) may affect clustering quality, a better study of the effects of increased (or decreased) ultrametricity on the results generated by a widely known and robust clustering algorithm was performed.

In order to study the impact of ultrametricity on cluster compactness and separation, we have implemented an algorithm that runs PAM on the original and transformed spaces, and computes those measure for each cluster from S and S'.

Our experiments considered a initial Euclidean space (S, d) where S corresponds to a set of objects and d to the Minkowski distance with exponent 2. To obtain a valid comparison of compactness and separation, the clusters obtained from a specific data set S must contain the same elements in the original and transformed spaces.

Dissimilarities d^x where $x > 1$ tend to decrease the ultrametricity of the original space, whereas dissimilarities where $0 < x < 1$ tend to increase ultrametricity.

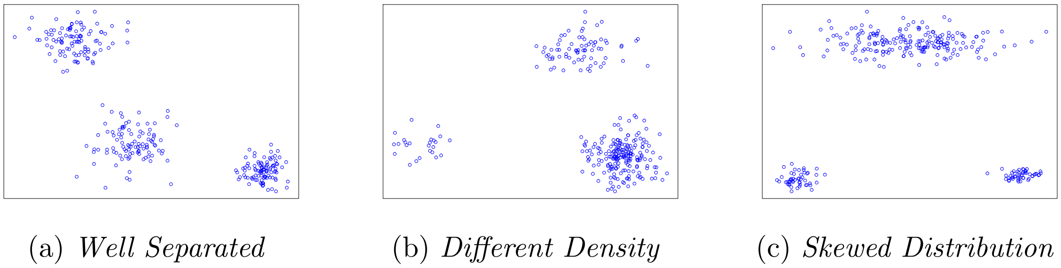

(a) *Well Separated* (b) *Different Density* (c) *Skewed Distribution*

Figure 4.1: Synthetic data containing 3 different data aspects: good separation, different density and skewed distributions

Current existing clustering validation measures and criteria can be affected by various data characteristics [53]. For instance, data with variable density is challenging for several clustering algorithms. It is known that k-means suffers from an uniformizing effect which tends to divide objects into relatively equal sizes [85]. Likewise, k-means and PAM do not have a good performance when dealing with skewed distribution data sets where clusters have unequal sizes. To determine the impact of ultrametricity in the presence of any of those characteristics, experiments were carried considering 3 different data aspects: good separation, density, and skewed distributions in three synthetic data sets named *WellSeparated*, *DifferentDensity* and *SkewDistribution*, respectively.

Figure 4.1 shows the synthetic data that was generated for each aspect. Each data set contains 300 objects.

Tables 4.1 shows the results for data sets *WellSeparated*, *DifferentDensity* and *SkewDistribution*, respectively. The measure (compactness or separation) ratio is computed dividing the transformed space measure by the original space measure. The average measure ratio computed for the 3 clusters is presented in each table.

Note that the average measure ratio is less than one for spaces with lower ultrametricity (obtained with dissimilarities d^5 and d^{10}). In this case, the average compactness ratio is also lower than the average separation ratio, showing that the transformations generated intra-cluster dissimilarities that shrunk more than the inter-cluster ones, relatively to the original dissimilarities. In spaces with higher ultrametricity (obtained with dissimilarities $d^{0.1}$ and $d^{0.01}$), the average measure ratio is higher than one. The average compactness ratio is also higher than the average separation ratio, showing that the transformations generated intra-cluster dissimilarities that expanded more than the inter-cluster ones. This explains the equalization effect obtained with the increase in ultrametricity.

Figures 4.2a, 4.2b and 4.2c show the relation between compactness and a separation ratio for each data set.

In Figure 4.2 we show the relationship between compactness and separation ratio for the three synthetic data sets and for the *Iris* data set which exhibit similar variation patterns.

As previously mentioned, data with characteristics such as different density and different cluster sizes might impose a challenge for several clustering algorithms.

Diss.	Compactness Avg.	Compactness Std.	Compactness Ratio Avg.	Normalized Mutual Info.
d	0.1298267265	0.0364421138	1	1
d^{10}	7.4595950908E-009	9.0835007432E-009	5.7458085055E-008	1
d^5	0.000048905	4.3815641482E-005	0.0003766941	1
$d^{0.1}$	0.8231766265	0.0254415565	6.3405790859	1
$d^{0.01}$	0.9722292326	0.0030358862	7.4886678515	1

Compactness and Clustering quality results for a data set with well-separated clusters

Diss.	Separation Avg.	Separation Std.	Separation Ratio Avg.
d	0.5904521462	0.339733487	1
d^{10}	0.0020607914	0.0035682378	0.0034901921
d^5	0.0473640032	0.0795298042	0.0802164976
$d^{0.1}$	0.9752251248	0.0521762794	1.6516581929
$d^{0.01}$	0.9979573861	0.0052696787	1.6901579451

Separation results for a data set with well-separated clusters

Diss.	Compactness Avg.	Compactness Std.	Compactness Ratio Avg.	Normalized Mutual Info.
d	0.2599331876	0.0225831458	1	0.4179803376
d^{10}	1.7193980983E-009	8.1299728150E-010	6.6147694106E-009	0.4413622617
d^5	4.4663622551E-005	7.7685178838E-006	0.0001718273	0.4197247625
$d^{0.1}$	0.8942911252	0.0073467836	3.4404653496	0.4186734225
$d^{0.01}$	0.9729198463	0.0174965529	3.7429612403	0.4186734225

Compactness and Clustering quality results for a data set with clusters with varied densities

Diss.	Separation Avg.	Separation Std.	Separation Ratio Avg.
d	0.8716430647	1.4832867815	1
d^{10}	0.0244453421	0.0423405745	0.0280451288
d^5	0.2484825264	0.4303843596	0.2850737147
$d^{0.1}$	0.8400992968	0.2757718021	0.9638111411
$d^{0.01}$	0.9777162094	0.0325513479	1.1216933272

Separation results for a data set with clusters with varied densities

Diss.	Compactness Avg.	Compactness Std.	Compactness Ratio Avg.	Normalized Mutual Info.
d	0.1072664803	0.098564337	1	1
d^{10}	0.000000449	7.7773337902E-007	4.1860674698E-006	1
d^5	0.0002096486	0.0003626508	0.0019544651	1
$d^{0.1}$	0.7880494471	0.0792970382	7.3466514879	1
$d^{0.01}$	0.9633479044	0.0171811278	8.9808848178	1

Compactness and Clustering quality results for a data set with skewed distributions.

Diss.	Separation Avg.	Separation Std.	Separation Ratio Avg.
d	0.971795701	0.0185685451	1
d^{10}	0.0029611253	0.0005897832	0.0030470656
d^5	0.0932204575	0.0090867253	0.0959259826
$d^{0.1}$	1.0416448857	0.001980664	1.0718764083
$d^{0.01}$	1.0047158048	0.0001909503	1.0338755396

Separation results for a data set with skewed distributions.

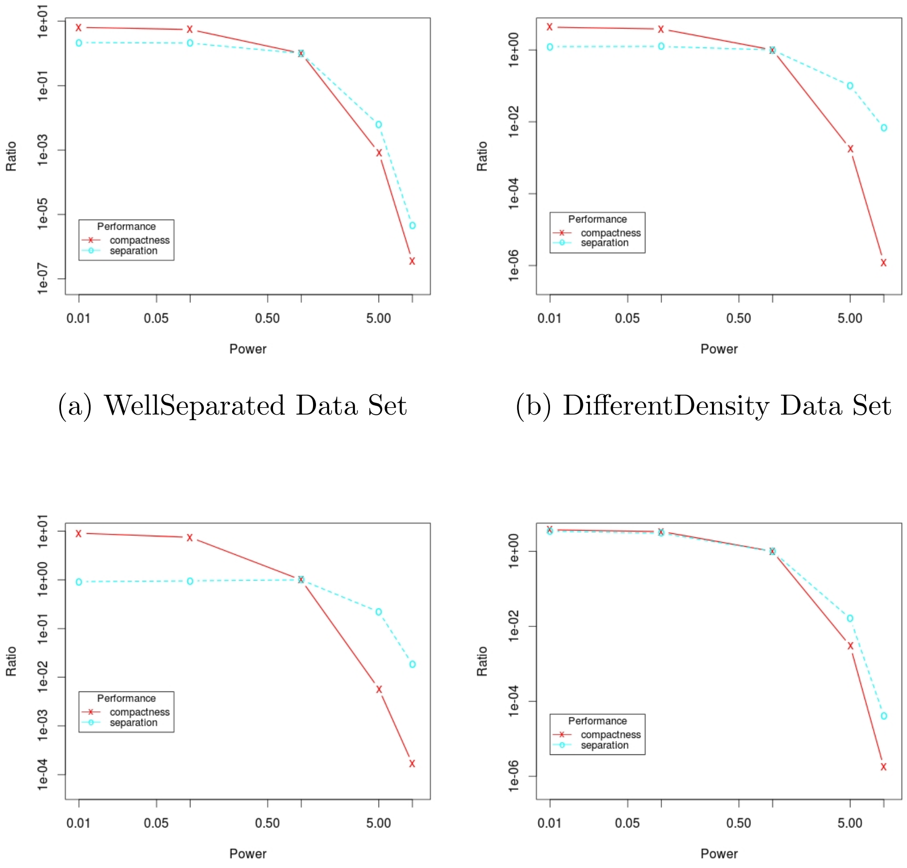
Diss.	Compactness Avg.	Compactness Std.	Compactness Ratio Avg.
d	0.2564313287	0.0572997859	1
d^{10}	4.495583902E-007	3.0731794825E-007	1.7531336456E-006
d^5	0.0007628527	0.0004963497	0.0029748809
$d^{0.1}$	0.8664974196	0.0223773478	3.379062238
$d^{0.01}$	0.9630194558	0.0029079036	3.7554672456

Compactness and Clustering quality results for the *Iris* data set.

Diss.	Separation Avg.	Separation Std.	Separation Ratio Avg.
d	0.2841621289	0.3120959612	1
d^{10}	1.1716078298E-005	2.0292841461E-005	4.1230259434E-005
d^5	0.0045832156	0.0079357613	0.0161288754
$d^{0.1}$	0.8715968561	0.0944160231	3.0672519923
$d^{0.01}$	0.9858840558	0.0108572902	3.4694421086

Separation results for the *Iris* data set.

Table 4.1: Cluster compactness and separation using PAM on three synthetic data sets and *Iris*. Both ratio averages are computed relative to the data set cluster compactness and separation values given by the original dissimilarity d.



(a) WellSeparated Data Set

(b) DifferentDensity Data Set

(c) SkewDistr Data Set

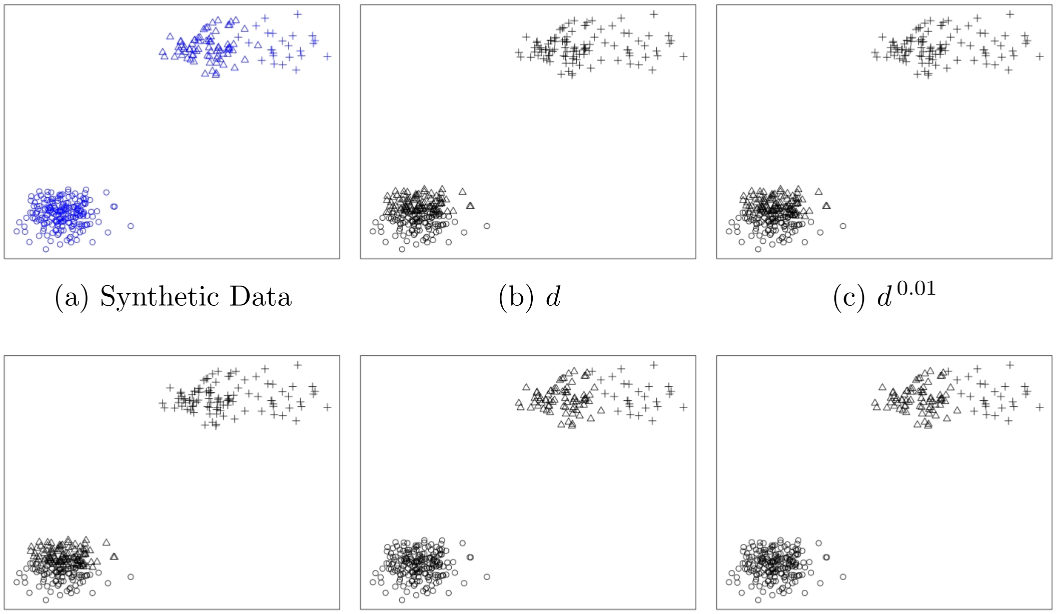
(d) Iris Data Set

Figure 4.2: Relation between Compactness and Separation Ratio for three synthetic data set and for the Iris data set

We show a scenario where PAM, when applied to the original Euclidean space, does not perform well. Nevertheless, we are able to improve the PAM's results by applying a transformation that decreases the ultrametricity of the original space and running PAM on the transformed space.

Consider the data set presented in Figure 4.3a which was synthetically generated in an Euclidean Space with pairwise metric d by three normal distributions with similar standard deviation but different densities. It has 300 points in total, with the densest group including 200 points and the other two containing 75 and 25 points.

Note that the somewhat sparse groups are also located very close to each other. Different symbols are used to identify the three distinct distributions. PAM's objective function tries to minimize the sum of the dissimilarities of all objects to their nearest medoid. However, it may fail to partition the data into the original distributions when dealing with different density data since the split of the densest cluster may occur. In our example, PAM does exactly that and also combines the two sparse clusters that are not well separated. Notice that unlike k-means (which also does not perform well in these scenarios but eventually can find the right partition due to the randomness on the selection of the centroids), PAM will most likely fails due to the determinism of its BUILD and SWAP steps combined and the choice of the objective function.



(a) Synthetic Data

(b) d

(c) $d^{0.01}$

(d) $d^{0.1}$

(e) d^{5}

(f) d^{10}

Figure 4.3: 4.3a shows the synthetic data generated from distributions with different density. 4.3b to 4.3f show the results of PAM using Euclidean distance d and other dissimilarities obtained by transformations on d.

To explore the positive effect of increased intra-cluster compactness generated by new spaces with lower ultrametricity on data containing those characteristics, we applied the same transformations with positive integer exponents to the original Euclidean distance matrix obtained from d. Results show significant improvement of the clustering. Figure 4.3b shows the result of applying PAM to cluster the synthetic data with dissimilarity d. Note that the clustering result does not correspond to a partition resembling the distributions that were used to generate the data. Figures 4.3d and 4.3c show that PAM also fails to provide a good partition with dissimilarities $d^{0.1}$ and $d^{0.01}$ since the increase in ultrametricity promotes equalization of dissimilarities which may degrade even more the results. Note however that the partitions obtained by PAM using the dissimilarities d^5 and d^{10} form similar clusters to the ones generated by the original distributions. Indeed, the increase in compactness helps PAM to create boundaries that are compliant with the original normal distributions.

Table 4.2 shows the measures and ratios for this data set. Figure 4.4 shows the relationship between compactness and separation ratio.

Diss.	Compactness Avg.	Compactness Std.	Compactness Ratio Avg.	Normalized Mutual Info.
d	0.1386920089	0.0558906421	1	0.6690207374
d^{10}	1.2953679952E-009	6.3343701540E-010	9.3398891934E-009	0.9365672372
d^{5}	2.8689799313E-005	1.0529323158E-005	0.0002068598	0.9365672372
$d^{0.1}$	0.8428018314	0.0308261682	6.0767872501	0.6702445506
$d^{0.01}$	0.9745718848	0.0037669287	7.026878423	0.6702445506

Diss.	Separation Avg.	Separation Std.	Separation Ratio Avg.
d	0.4604866874	0.7771228672	1
d^{10}	0.0114269071	0.0197919837	0.0248148479
d^{5}	0.104837087	0.1815831588	0.2276658368
$d^{0.1}$	0.8160827216	0.2379010818	1.7722178381
$d^{0.01}$	0.978284428	0.0270049282	2.1244575681

Table 4.2: Data set comprising clusters with different density.

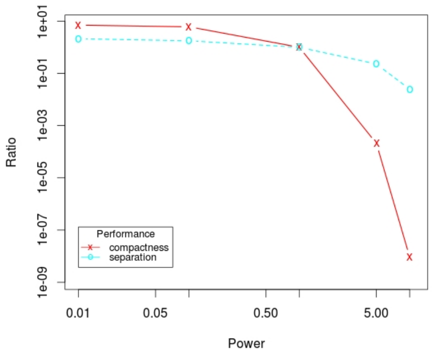


Figure 4.4: Relation between Compactness and Separation Ratio for the test data set

4.2 Promoting Clustering results by increasing clusterabiliy of the dataset

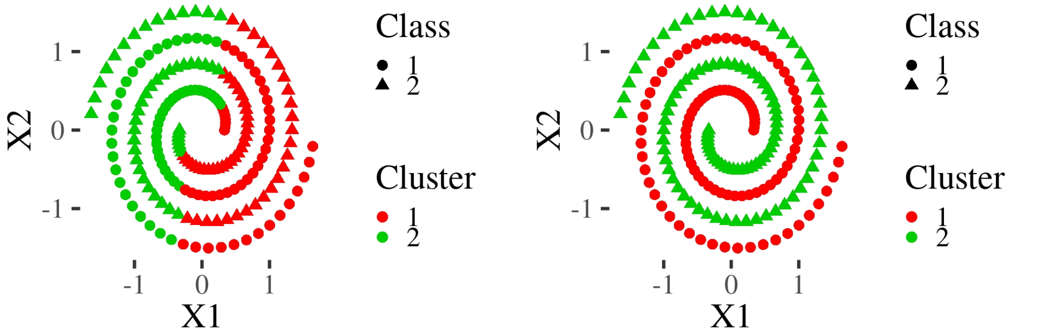
From definitions of clusterability and ultrametricity, we note that the higher the ultrametricity of a data set, the easier it is to cluster. For data sets with very high ultrametricity, it is easy to get clustering results consistent with the natural cluster structure by applying arbitrary clustering algorithms.

This idea is most useful when we encounter a data set that is difficult to cluster using typical clustering algorithms. We can modify the dissimilarity matrix to elevate the clusterability (or ultrametricity) of the dataset, and then perform clustering on the resulting higher ultrametricity data set. Distance represents the pairwise relationships between data points in the dataset. One way to improve the clusterability of a dataset is to raise the ultrametricity through the special matrix product which we proposed in Section 3.4 of Chapter 3.

Suppose we have a distance matrix A of a dataset S. If we need m min-max matrix multiplications to achieve stability, then we define the clusterability of S to be $\frac{|S|}{m}$. The corresponding ultrametricity of its distance matrix is also $\frac{|S|}{m}$. If we view $A^2 = A \cdot A$ as the distance matrix of dataset S, then the ultrametricity of distance matrix A^2 is $\frac{|S}{m-1}$. Thus when applying the same clustering algorithm on both distance matrices, we expect a better clustering performance on distance matrix A^2 than on A.

Along these lines, we can continue using higher powers of A as the distance matrix of S to improve clusterability. Since $A^m = A^{m+1}$, A^m is the optimal distance matrix for S with highest ultrametricity. In this case, the ultrametricity of the distance matrix of S becomes $|S|$, which means the distance matrix of S is pure ultrametric. If we apply clustering algorithms on this distance matrix, we expect better clustering performance than on any lower power of A.

We verified this theory on a difficult data set. During the experiment, we first apply a particular clustering algorithm on the original distance matrix A, and we get a clustering result π_1. Then we use the same clustering algorithm on distance matrix A^m, and we get a new clustering result π_m. We compare the two results to ground truth partition π_0 and graph them in Figure 4.5. Since we are dealing with distance matrix only, we pick k-medoids or Partitioning Around Medoids (PAM) algorithm as the testing tool. PAM is one of the most common clustering algorithms that can be applied to distance matrices. However, similar to k-means, PAM will also fall into the local optima and is not suitable for non-sphere shaped clusters.



(a) Clustering result on Spiral dataset based on original dissimilarity matrix

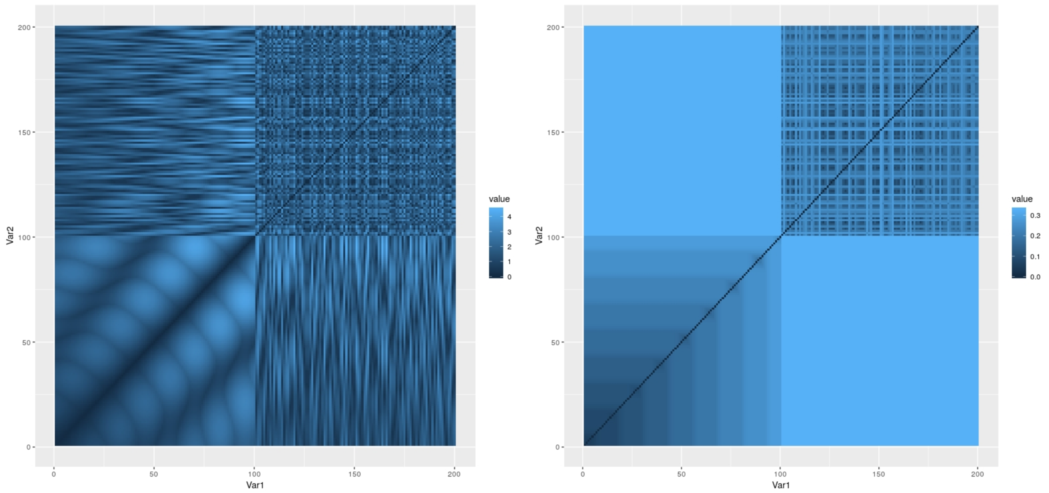
(b) Clustering result on Spiral dataset based on maximum ultrametricity matrix

Figure 4.5: Two entangled spirals with total of 200 data points and perform k-medoids on it.

To exploit the weaknesses of PAM, we select a data set with two entangled spirals. Figure 4.5 shows the comparison of the clustering results of PAM on the original distance matrix of the data set and on the maximum ultrametricity distance matrix. From Figure 4.5a, we can see that the PAM could not clearly distinguish one spiral from the other. This is the result of running the clustering algorithm on the original distance matrix. The distance we use here is the Manhattan distance. Since PAM is not friendly to non-sphere shaped clusters, it is unable to extract the real groups from this data set correctly. Nevertheless, in Figure 4.5b, we promote the ultrametricity of the distance matrix (the stabilization power is 123) of the spiral data set. When we apply PAM on the new distance matrix, the clustering algorithm becomes powerful enough to separate the data set according to its natural cluster shape.

Similar phenomenon can be seen in the distance matrix. We draw the heatmap for the original distance matrix and the new ultrametric distance matrix (Figure 4.6). In both figures, the darker the color is, the smaller the pairwise distance between two points. The label of points ranges from 1 to 200. The first 100 points come from the first spiral and the rests are from the second. In the heatmap, the smaller labels are put at the lower-left corner of the graph. Therefore, the lower-left and upper- right corners of the distance matrix represent the pairwise distances between the point of the same spiral. The other regions of the distance matrix represent the distance between the two spirals.

The left graph(Figure 4.6a) represents the original distance matrix. It is the Manhattan distance between points. In Euclidean space, it is hard to distinguish points from two spirals. Therefore, PAM will fail to partition the two spirals correctly. However, after promoting the ultrametricity of the distance matrix, we get a new ultrametric distance matrix. The heatmap for this matrix is shown in Figure 4.6b. From it, we can see a clear separation in the distance matrix. The distances between points from different spirals are all larger than the distances between points within



(a) Original distance matrix on Spiral dataset

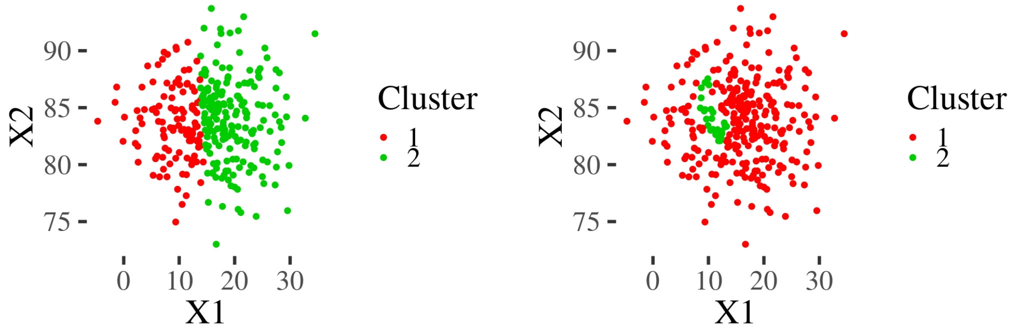
(b) Maximum ultrametricity distance matrix on Spiral dataset

Figure 4.6: Distance matrix of two entangled spirals with total of 200 data points

one spiral. The differences between points in different spirals are enlarged after the increment of ultrametricity of the distance matrix. Thus, PAM algorithm is more effective on the new distance matrix than on the original one.

We perform a similar experiment on the data set depicted in Figure 1.1a of Chapter 1. Obviously, there is no clustering structure inside this data set; it is generated from one Gaussian distribution. If we apply PAM to the original data set, we will still get some clustering results (Figure 4.7a). However, such results can not meaningfully express the inherent data structure. Nevertheless, when we run PAM on the ultrametric matrix generated by the special matrix product (the stabilization power is 31) on the original distance matrix, we get a partition in which the vast majority of points in one cluster and only a few points in the center are in the other cluster. The comparison can be seen in Figure 4.7.

Although the clustering result on the ultrametric matrix is still not a perfect representation of the lack of cluster structure in the data set, it still indicates to the users that the majority of the data set is in one cluster. This is closer to the true

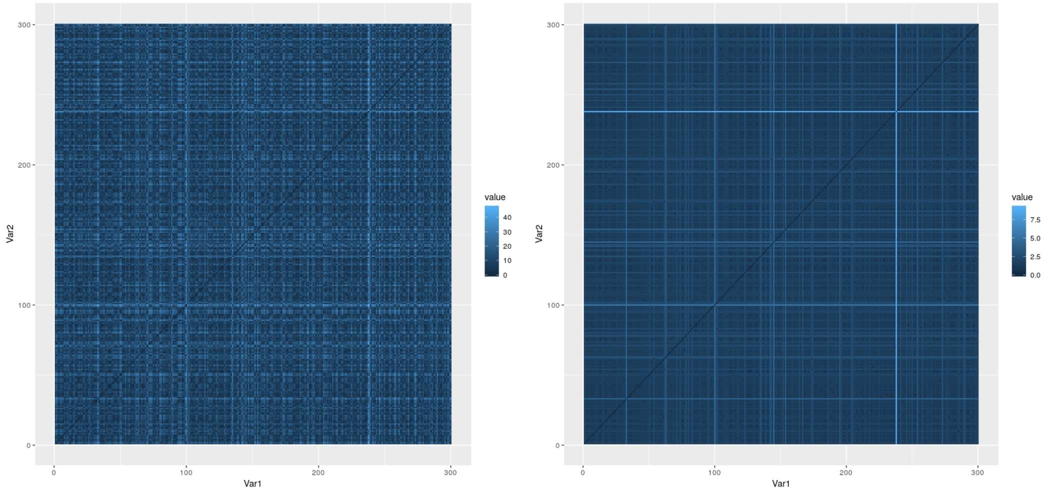


(a) Clustering Result based on original dissimilarity matrix of data set with no clustering structure

(b) Clustering Result based on maximum ultrametricity matrix of data set with no clustering structure

Figure 4.7: Data set with no data structure and has total of 300 data points and perform k-medoids on it.

structure than the previous result, which created equal sized clusters. The Heatmaps of the two distance matrices behave similarly to those of the spiral data set. Since there is no clustering structure inside the data set, the ultrametric distances between points are expected to be identical. In the corresponding heatmap, there are many regions of uniform color. This demonstrates the similarity of the distance values in the ultrametric matrix. On the other hand, the original distance matrix has a large variation in its values. Figure 4.8 represents the heatmap of these two matrices.



(a) Original Distance matrix on the one Gaussian cluster dataset

(b) Maximum Ultrametricity distance matrix on the one Gaussian cluster dataset

Figure 4.8: Distance matrix of data set with one Gaussian distributed cluster. It contains total of 300 data points

CHAPTER 5

ON FINDING NATURAL CLUSTERING STRUCTURES IN DATA

In this chapter, we illustrate the dual-criteria method that assists in the unsupervised model selection process. Experiments are performed on both real and synthetic data sets. The results indicate our approach is effective at detecting the natural number of clusters. An evaluation algorithm is also introduced to validate the results generated by the dual-criteria method. Multiple choices of the parameter β are explored, and the significance of varying β on solving problems with the data set of imbalance-distributed clusters is also discussed.

5.1 Introduction

A simple approach to the problem of determining the number of clusters is to generate several partitions with different number of clusters and to choose the best partition based on an internal evaluation index. By plotting the dependency of this index on the number of clusters, it is possible to determine the number of clusters. One of the best-known techniques for the determination of the number of clusters is to check the elbow point on the resulting curve [69]. This elbow point is loosely defined as the point of maximum curvature and the desired number of clusters is the cluster coordinate of the elbow point.

An alternative method is the gap statistics which aims to formalize the intuitive approach of the elbow method by comparing of the logarithm of the cohesion with a reference distribution of the data [80]. However, this method only works on well-separated datasets. An alternative approach proposed in [79] regards clustering as a supervised classification problem which requires the estimation of "true" class labels. The prediction strength measure evaluates the number of groups that can be predicted from data.

In [20] the largest ratio difference between two adjacent points is used to locally find the elbow point along the curve. Other authors use more than one pairs of points. The first data point with a second derivative above some threshold value is used to specify the elbow point [29, 32], while in [71] the data point with the largest second derivative is used. All these techniques are sensitive to outliers and local trends, which may not be globally significant [69].

Yet another approach to the estimation of the number of clusters is applying consensus clustering [59] and resampling [68]. This involves clustering many samples of the data set, and determining the number of clusters where clusterings of the various samples are the most stable [69]. Consensus clustering or clustering aggregation, has been explored for decades. A formal definition is given in [35], where consensus clustering is defined as a clustering that minimizes the total number of disagreements with a set of clusterings. This technique can deal with a variety of problems such as developing a natural clustering algorithm for categorical data, improve the clustering robustness by combining the results of many clustering algorithms, as well as determine the appropriate number of clusters. In recent years, many approaches have been developed to solve ensemble clustering problems [48, 49, 27, 11, 31, 52] and [84].

As a task of consensus clustering, determining the number of clusters has been considered in several publications. In [88] a hierarchical clustering framework is proposed that combines partitional clustering (k-means) and hierarchical clustering. A random walk technique on the graph defined by a consensus matrix of clusterings is used in [82] to determine the natural number of clusters.

Information-theoretical methods are also applicable for detecting the number of clusters in a dataset by defining a "jump method" of the transformed distortion d

on a partition π_d. The highest increase of d indicates the number of clusters with respect to π_d [77]. However, this approach is based on a strong assumption that the clusters are generated based on Gaussian distributions. By integrating *Rényi* entropy and complement entropy together, Liang *et al* [50] propose a method which can determine the number of clusters on a dataset that has mixed set of feature types. Their approach proposes a clustering validation index which considers within-cluster entropy and between-cluster entropy and the best number of clusters is chosen when such index reaches the maximum. The relationship between the k-means and the expectation maximization algorithm was studied within the framework of information theory by Kearns, Mansour and Ng in [44].

There also several other methods on detecting the number of clusters in a dataset. In [42] the Maximum Stable Set Problem (MSSP) combined by Continuous Hopfield Network (CHN) is used to find the natural number of clusters of a data set. The algorithm detects the number of stable sets and uses this to represent the number of clusters.

Shaqsi and Wang [9, 72] work with a similarity parameter and observe that in a certain range of this parameter, the number of clusters formed by their 3-staged algorithm remains constant. The number of clusters that corresponds to the longest interval is chosen as the most appropriate number.

Kolesnikov, Trichina, and Kauranne [45] create a parametric modeling of the quantization error to determine the optimal number of clusters in a dataset. This method treats the model parameter as the effective dimensionality of the dataset. By extending the decision-theoretic rough set model an efficient method to detect the number of clusters is presented in [86]. This model applies the Bayesian decision procedure for the construction of probabilistic approximations. Hamerly and Elkan [37] propose an algorithm that based on a statistical test for the hypothesis that a subset of data follows a Gaussian distribution. It only requires one parameter and avoids the calculation of the covariance matrix.

An incremental approach called "dip-means", is introduced in [41] with the underlying assumption that each cluster admits a unimodal distribution. The statistic hypothesis test for unimodality (dip-test) is applied on the distribution of distances between one cluster member and others.

In [83] a new method for automatically detecting the number of clusters based on image processing techniques is discussed. This method adopts the key part of Visual Assessment of Cluster Tendency(VAT), and regards the dissimilarity matrix as an image matrix. Image segmentation techniques are applied to an image generated by this matrix, followed by filtering and smoothing to decide the number of clusters in the original data.

Cheung [19] proposes a new novel algorithm that can automatically select the number of clusters by presenting a mechanism to control the strength of rival penalization dynamically.

We propose a new methods to evaluate the number of clusters. It seeks to minimize both clustering partition entropy and the cohesion of clustering. Since partition entropy is anti-monotonic and cluster cohesion is a monotonic function relative to the partial order set of partitions, we can use the Pareto Front to identify the natural number of clusters existent in a data set.

After that, we evaluate the results using an approach based on the metric space of partitions of a dataset that makes use of β-entropy, a generalization of Shannon's entropy introduced in [24, 38], and axiomatized in [76]. Other significant generalizations of entropy belong to C. Tsallis [81, 73].

The main idea is that if a natural clustering structure exists in data, two methods (e.g. k-means and one of several variants of hierarchical clustering) produce similar clustering results. The extent to which partitions are distinct is evaluated using a distance between partitions that generalizes the distance between partitions induced by Shannon's entropy studied by L. de Mántaras [25]. We extend his results to distances produced by β-entropies and suggest that this generalization may be useful for clustering imbalanced data.

This chapter is organized as follows. In Section 2.2 partition entropy is introduced. Section 5.2 illustrates the compromise between cluster entropy and its corresponding cohesion and how it is used on determining the number of clusters. The experimental results are analyzed in Section 5.3. We further validate the previous approach using contour maps in Section 5.4. Conclusions and future work are discussed in Section 5.6.

5.2 Dual Criteria Clustering using Entropy and Cohesion

Our approach in identifying the natural number of clusters is seeking a compromise between the partition entropy and the cohesion of clustering.

Partition entropy evaluates the imbalance between the sizes of the clusters that constitute a partition. For a fixed number of blocks, the entropy is maximal when blocks have equal sizes. As we saw in Section 2.2, the smaller the partition in the poset $(\mathsf{PART}(S), \leqslant)$ the larger the entropy. Thus, the largest value of the entropy of a partition of S is achieved for ι_S; the smallest value is obtained for the one-block partition ω_S.

Cohesion is a measure of the quality of a clustering, defined as the within-cluster sum of squared errors and denoted by sse.

Let S be the set of objects to be clustered. We assume that S is a subset of $\mathbb{R}^n$ equipped with the Euclidean metric. The *center* $\mathbf{c}_C$ *of a subset* C of S is defined as $\mathbf{c}_C = \frac{1}{|C|}\sum_{\mathbf{o}\in C}\mathbf{o}$.

For a partition $\pi = \{C_1, C_2, \ldots, C_m\}$ of S the sum of square errors sse of π is defined as

$$\mathsf{sse}(\pi) = \sum_{i=1}^{m}\sum_{\mathbf{o}\in C_i} d^2(\mathbf{o}, \mathbf{c}_{C_i}). \tag{5.1}$$

It is possible to show (see [74]) that if $\kappa, \lambda \in \mathsf{PART}(S)$ and $\kappa \leqslant \lambda$, then $\mathsf{sse}(\kappa) \leqslant \mathsf{sse}(\lambda)$. Thus, cohesion is an anti-monotonic function on the partially ordered set $(\mathsf{PART}(S), \leqslant)$; we have $\mathsf{sse}(\iota_S) = 0$ and $\mathsf{sse}(\omega_S) = \sum_{\mathbf{o} \in S} \parallel \mathbf{o} \parallel^2 - |S| \parallel \mathbf{c}_S \parallel^2$. We may conclude that the entropy varies inversely with the cohesion of partitions.

Entropy and cohesion describe the clustering results from two different perspectives and this suggests that a bi-criterial optimization would be helpful for choosing the best clusterings.

We aim to simultaneously minimize $H(\pi)$ and $\mathsf{sse}(\pi)$ that have inverse types of variations with clusterings considered as partitions. This will allow us to define a natural number of clusters using the Pareto front of this bi-criterial problem. Let $\mathbf{F} : \mathsf{PART}(S) \longrightarrow \mathbb{R}^2$, where

$$\mathbf{F}(\pi) = (H(\pi), \mathsf{sse}(\pi)), \tag{5.2}$$

where $\pi \in \mathsf{PART}(S)$.

Definition 5.2.1. *Let $\pi, \sigma \in \mathsf{PART}(S)$. The partition σ* dominates *π if $H(\sigma) \leqslant H(\pi)$ and $\mathsf{sse}(\sigma) \leqslant \mathsf{sse}(\pi)$.*

A partition $\tau \in \mathsf{PART}(S)$ is Pareto optimal *if there is no other partition that dominates τ.*

In principle, several optimal partitions may exist, each with a specific number of clusters. The set of partitions that are not dominated by other partitions is the *Pareto front* of this problem (see [57, 66]).

If a partition π is Pareto optimal, then it is no worse than another partitions from the point of view of ($H(\pi)$ and $\mathsf{sse}(\pi)$) and is better in at least one of these criteria.

To speed up the search for the members of the Pareto front we scalarize the problem by computing a single objective optimization function defined utilizing the concept of hypervolume [89] on entropy and sse. The hypervolume measure is the size of the space covered or size of dominated space (see [89]), is the Lebesgue measure Λ of the union of hypercubes a_i defined by a non-dominated point m_i and a reference point x_{ref} [21].

In our case, we set the reference point at the position that both entropy and sse reaches its maximum. The maximum of entropy will be reached on partition ι_S, while the maximum value of sse is obtained at partition ω_S. Then, the hypervolume that corresponds to a partition π is

$$\mathsf{HV}(\pi) = (H(\iota_S) - H(\pi))(\mathsf{sse}(\omega_S) - \mathsf{sse}(\pi)) \tag{5.3}$$

The optimal partition for a dataset is obtained as

$$\pi_{opt} = \underset{\pi}{\operatorname{argmax}} \ \mathsf{HV}(\pi) \tag{5.4}$$

The optimal number of clusters is computed by Algorithm 1.

```
Algorithm 1: Computation of the optimal number of clusters
Input: Dataset $\mathbf{S}$, maximum number of clusters $k_{max}$
Output: The optimal number of clusters $k$ for the input dataset
Initialize a list $\mathbf{HV}$ with length $k_{max}$;
for $i$= 1 to $k_{max}$ do
    Compute clustering $\pi_i$ on $\mathbf{S}$ with $i$ clusters;
    Calculate $\mathsf{sse}(\pi_i)$ for partition $\pi_i$;
    Calculate the entropy $H(\pi_i)$ for partition $\pi_i$;
    Calculate hypervolume $\mathsf{HV}(\pi_i)$ using Equality (5.3);
    Set $\mathbf{HV}_i = \mathsf{HV}(\pi_i)$;
return $k = \underset{i}{\operatorname{argmax}} \ \mathbf{HV}_i$
```

5.3 Experimental Results

The previous approach was tested on different datasets to evaluate its performance. We used 5 synthetic datasets and 7 real-world datasets described in Table 5.1.

Table 5.1: Data Set Information

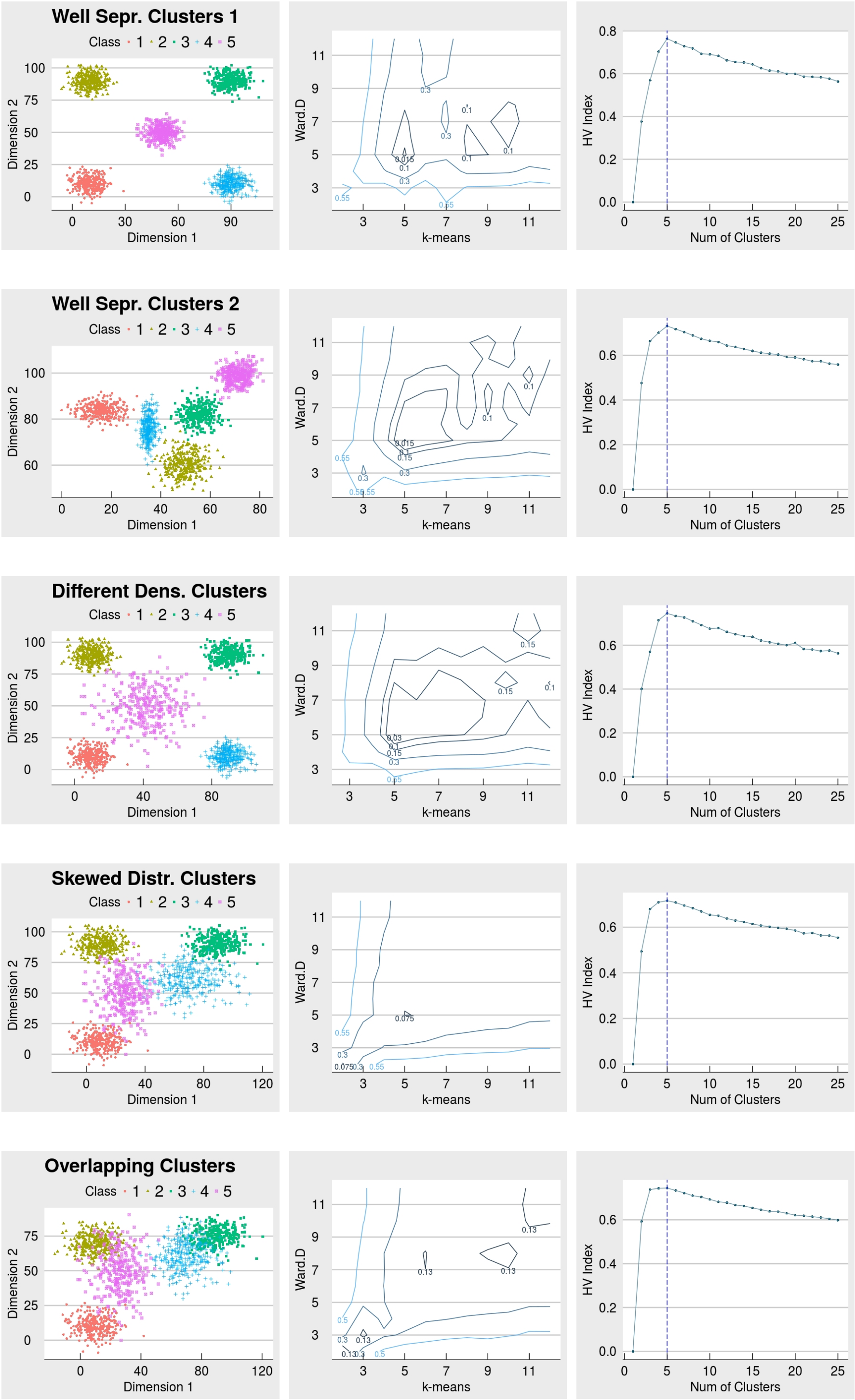
Data Set	Cardinality	Attributes	Class Number
Well Sepr. I	900	2	5
Well Sepr. II	900	2	5
Diff. Density	900	2	5
Skewed Dist.	900	2	5
Overlapping	900	2	5
Iris	150	4	3
Wine	178	13	3
Libras	360	90	15
Ecoli	336	7	8
Vowel	990	12	11
PenDigits	10992	16	10
Poker(1-9)	511308	10	9

The 2-dimensional synthetic datasets contain 5 Gaussian distributed clusters; each cluster contains 300 data points produced using the R function RMVNORM implemented by Leisch, F. *et al* [34]. By varying the means and standard deviations, we obtained five different types of clusterings shown in the first column of Figure 5.1. having the following features:

- clusters that are well separated;
- clusters that are well separated but closer with each other;
- clusters that have different density;
- clusters that have different sizes and number of points;
- clusters that overlap.

Also, we used several real-world data sets which originate from UCI machine learning repository [51]:

Iris Data: This dataset contains 150 cases and 4 variables named Sepal.Length, Sepal.Width, Petal.Length and Petal.Width corresponding to 3 species of iris (*setosa*, *versicolor*, and *virginica*) [13].



(a) Data Structure (b) Contour Map (c) HV-index

Figure 5.1: The Contour Map of the overlapping datasets and its corresponding clustering structure; the x-axis of the contour graph represents the number of cluster of k-means clustering while the y-axis represents those of hierarchical clustering.

Wine Recognition Data: These data are the results of a chemical analysis of wines. The analysis determined the quantities of 13 constituents found in each of the three types of wines [6], which contain 59, 71, and 48 records, respectively.

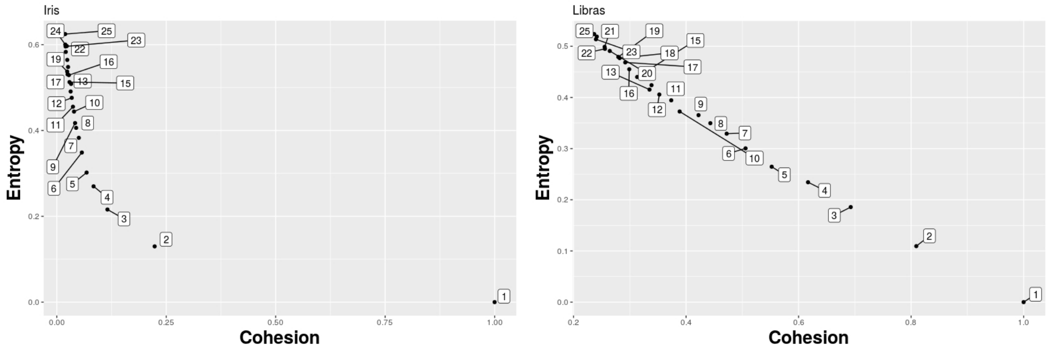
LIBRAS Movement Database: LIBRAS, acronym of the Portuguese name "*LÍngua* BRAsileira de Sinais", is the official Brazilian sign language. The dataset contains 15 classes of 24 instances each, where each class refers to a hand movement type in LIBRAS. Each instance represents 45 points on a 2-dimensional space, which can be plotted in an ordered way (from 1 through 45 as the x-coordinate) in order to draw the path of the movement.

Pen-Based Recognition of Handwritten Digits: The digit database was created by collecting 250 samples from 44 writers. Digits are represented as feature vectors by using linear interpolation between pairs of (x_t, y_t) points. Here x_t and y_t is the coordinate information for the digits at when the writer is written. There are 10 different digits in the data set and the numbers of instance for each digits are roughly the same.

Ecoli Dataset This dataset contains 360 instances and 7 features and is used to predict the protein localization site. 8 classes are embedded into the dataset with the largest class of 143 data points and the smallest one of only 2.

Vowel Recognition The dataset is generated from speakers' independent recognition of the eleven steady state vowels of British English using a specified training set of LPC derived log area ratios. It consists of a three-dimensional array: voweldata [speaker, vowel, input]. The speakers are indexed by integers 0-89. (Actually, there are fifteen individual speakers, each saying each vowel six times.) The vowels are indexed by integers 0-10. For each utterance, there are ten floating-point input values, with array indices 0-9. It has 990 instances.

Poker Dataset This dataset records a set of card types people hold in their hands. Each record is an example of a hand consisting of five playing cards drawn from a standard deck of 52. Each card is described using two attributes (suit and rank), for a total of 10 predictive attributes. There is one Class attribute that describes the "Poker Hand". We omit the Class 0 (Nothing in hand; not a recognized poker hand).



(a) Pareto Front for Iris Dataset

(b) Pareto Front for Libras Dataset

Figure 5.2: The Pareto Front of solutions of Equation (5.2) for Iris and Libras dataset using k-means clustering algorithm. The labelled points represent Pareto optimal partitions and the labels show the corresponding number of clusters. x-axis represents the cohesion while y-axis is the entropy. Both are normalized into $[0, 1]$.

The algorithms described in Section 5.2 are applied on the datasets previously mentioned. To verify the stability of our method, several popular methods on determining number of clusters are used for comparison.

Gap Statistics: This method proposed in [80] gives the natural number of clusters by defining a gap function as follows:

$$Gap_n(k) = E_n^* log(W_k) - log(W_k),$$

where W_n is the pooled within-cluster sum of squares around the cluster means for k clusters, E_n^* denotes the expectation under a sample of size n from the reference distribution. The estimated $\hat{k}$ will be the value maximizing $Gap_n(k)$ after taking the sampling distribution into account.

The idea of this criterion is to standardize the graph of $\log(W_k)$ by comparing it with its expectation under an appropriate full reference distribution of the data. The estimate of the optimal number of clusters is then the value of k for which $\log(W_k)$ falls the farthest below this reference curve. We use the function `clusGap` in R package "cluster" for simulation [54].

Jump method: It uses the concept of distortion to describe the within-cluster dispersion for a particular partition [77]. The definition of the minimum achievable distortion associated with fitting k centers to the data is

$$d_k = \frac{1}{p} \min_{\mathbf{c}_1,\dots,\mathbf{c}_k} E[(\mathbf{X}-\mathbf{c}_\mathbf{x})^T \Gamma^{-1} (\mathbf{X}-\mathbf{c}_\mathbf{x})], \tag{5.5}$$

where $\mathbf{X}$ is a p-dimensional random variable having a mixture distribution of k components, each with covariance Γ. The $\mathbf{c}_1, \dots, \mathbf{c}_k$ are a set of candidate cluster centers and $\mathbf{c}_\mathbf{x}$ is the one closest to $\mathbf{X}$.

Equality (5.5) gives the average Mahalanobis distance, per dimension, between $\mathbf{X}$ and $\mathbf{c}_\mathbf{x}$. If Γ is the identity matrix, distortion will be the mean squared error. The number of clusters k is determined as

$$k = \underset{k}{\operatorname{argmin}}\ d_k^{-Y} - d_{k-1}^{-Y},$$

where Y is an arbitrary value called transformation power and it usually equals to $\frac{p}{2}$.

Prediction Strength: For a particular dataset S, let $\mathcal{X}_{tr}$ and $\mathcal{X}_{te}$ be the training and testing subset of the data, where $\mathcal{X}_{tr} \cup \mathcal{X}_{te} = S$. Then we partition both $\mathcal{X}_{tr}$ and $\mathcal{X}_{te}$ into k clusters. Let $\pi_{te} = \{A_1, \dots, A_k\}$ and $\pi_{tr} = \{B_1, \dots, B_k\}$ be the partitions for $\mathcal{X}_{te}$ and $\mathcal{X}_{tr}$, respectively. The prediction strength of S given k is defined in [79] as

$$PS(k) = \min_{1 \leqslant l \leqslant k} \frac{\sum_{i \neq j} \{\delta((x_i, x_j), \mathcal{X}_{tr}) \mid x_i, x_j \in A_l\}}{|A_l|(|A_l|-1)},$$

where $\delta((x_i, x_j), \mathcal{X}_{tr})$ is defined for pairs (x_i, x_j) that belong to the same cluster in the testing sets as $\delta((x_i, x_j), \mathcal{X}_{tr}) = 1$ if x_i and x_j are assigned to the same closest centroid in $\mathcal{X}_{tr}$ and is 0 otherwise. This method is mainly implemented with the help of function `prediction.strength` in R package “fpc” [39].

Regularized Information Maximization (RIM): This technique was introduced in [36]. It seeks to optimize a criterion that captures class separation, class balance and classifier complexity and may be interpreted as maximizing the mutual information between the empirical input and implied label distributions.

The Akaike Information and the Bayesian Information Criteria: An alternative approach for determining the number of clusters for the k-means algorithm is using the Akaike Information Criterion [7, 8] (AIC), or the Bayesian Information Criterion (BIC) [70] for model selection. When these criteria are applied to the k-means clustering they can be written as

$$\begin{aligned} \mathsf{AIC} &= \underset{k}{\operatorname{argmin}}[-2L(k) + 2kd] \text{ and,} && (5.6)\\ \mathsf{BIC} &= \underset{k}{\operatorname{argmin}}[-2L(k) + ln(n)kd], && (5.7) \end{aligned}$$

where k is the number of clusters, $-L(k)$ is the negative maximum log-likelihood, d is the number of features, and n is the number of points in the dataset. For clustering the first term $-2L(k)$ in these definition is the minimum of the sum of squared errors for k clusters, $\min\{\mathsf{sse}(\pi) \mid |\pi| = k\}$ and both models aim to balance the model distortion (a measure of the extent data points differ from the prototype of the their clusters) and model complexity, where a penalty is incurred for each newly created cluster [55]. Therefore, the previous expressions become:

$$\mathsf{AIC} = \underset{k}{\operatorname{argmin}}\{\mathsf{sse}(\pi) + 2kd \mid \pi \in \mathsf{PART}_k(S)\}, \tag{5.8}$$

and

$$\mathsf{BIC} = \underset{k}{\operatorname{argmin}}\{\mathsf{sse}(\pi) + ln(n)kd \mid \pi \in \mathsf{PART}_k(S)\}. \tag{5.9}$$

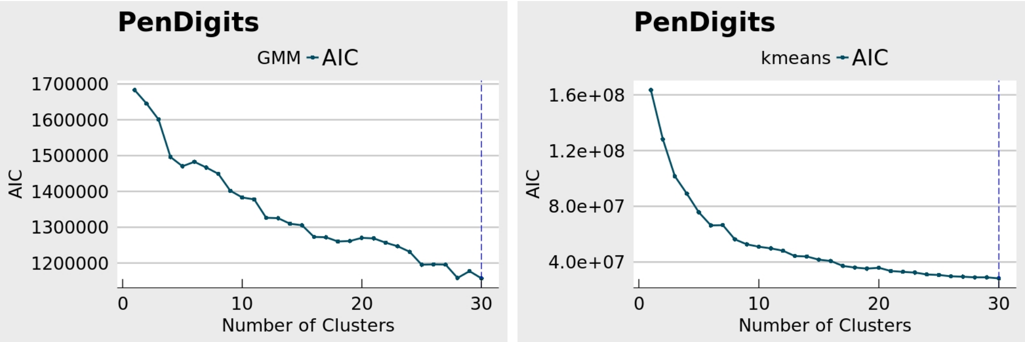
Note that BIC has a larger penalty term and the model selected by BIC tends to be simpler.

The minimization of both model distortions (sums of squared errors) and the model complexity (number of clusters) can be regarded as a bi-criteria optimization problem and thus, it is similar to our approach. However, both the penalty terms of both AIC and BIC depend on the values of k not on the actual clusterings.

From the point of view of multi-objective optimization the two methods are just the traditional weighted sum method of optimizing $\mathsf{sse}(\pi)$ and kd with weights $2d$ and $\ln(n)d$, respectively. Nevertheless, the lack of consistency of the scale of two terms (distortion and complexity) may lead to a significant problem. For datasets with high dimensionality, the penalty term dominates the smaller $\mathsf{sse}(\pi)$ term and the minimum of the expression is be reached for $k = 1$. Such cases occur in text clustering [55]. If the dataset is pretty sparse (which results in a large value of $\mathsf{sse}(\pi)$) and the the number of dimension is relatively small, the whole expression will be dominated by the $\mathsf{sse}(\pi)$ term. Examples can be seen from large datasets *PenDigits* and *Poker*. Figure 5.3 and 5.4 demonstrates the phenomenon on *PenDigits* dataset. In both cases, the value of the criterion does not show any minimum for the values of the AIC or BIC criteria.

Our proposed hypervolume does not require weighting the contributions of the balancing and cohesion term, unlike the variational approaches that seek to integrate these contravariant factors [36, 65, 10] and require hyper-parameters that affect the resulting number of clusters. In most cases, we use the Shannon entropy (for $\beta = 1$) avoiding the use of hyper-parameters. Note that the parameter β is used for $\beta \neq 1$ only when we apply our approach to imbalanced data sets.

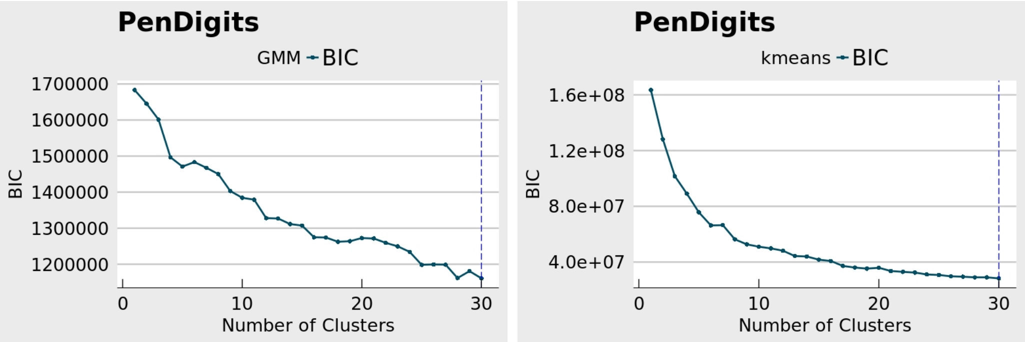
We used the functions `Optimal_Clusters_GMM` and `Optimal_Clusters_KMeans` of the R package `ClusterR` [60] to seek the number of clusters in several data sets using both AIC and $\mathsf{BIC}\dot{}$The first function seeks the optimal number of clusters for



(a) AIC for PenDigit using the Gaussian Mixture Model

(b) AIC for PenDigit by *k*-means clustering

Figure 5.3: AIC behavior on PenDigit dataset.



(a) BIC for PenDigit using the Gaussian Mixture Model

(b) BIC for PenDigit by *k*-means clustering

Figure 5.4: BIC behavior on PenDigit dataset.

a Gaussian mixture model using the EM algorithm, while the second computes the number of clusters for *k*-means. To avoid of losing generality, we apply both AIC and BIC methods.

Our method performs well on several synthetic datasets with the ambiguity of the dataset with overlapping clusters, as shown in Figure 5.1.

Nevertheless, this dataset can also be viewed as a 3 cluster dataset in which 2 clusters have 2 overlapping subclusters as shown in the last row of Figure 5.1. From

Table 5.2: Comparison between the number of clusters for datasets; g represents the number of clusters obtained by using the log-likelihood function of Gaussian Mixture Model while k represents those numbers when using the sum of squared errors.

Data Sets	β	natural number of clusters(CPU Times[seconds])							
		Gap Stat.	Jump Mthd.	Pred. Strgth.	AIC(g/k)	BIC(g/k)	RIM	HV Index	Cntr. Mthd.
Well Sepr. I	1.00	5(3.92)	5(0.87)	3(2.90)	8(1.23)/30(0.29)	8(1.14)/30(0.34)	12(976)	**5**(0.92)	5
Well Sepr. II	1.00	5(4.04)	5(0.92)	5(2.82)	13(1.19)/30(1.11)	5(1.23)/30(1.12)	6(977)	**5**(0.90)	5
Diff. Density	1.00	5(4.13)	5(0.97)	5(2.96)	5(1.30)/30(0.31)	5(1.11)/30(0.37)	4(968)	**5**(0.95)	5
Skewed Dist.	1.00	5(4.17)	30(1.06)	5(3.05)	6(1.49)/30(0.32)	5(1.13)/30(0.33)	3(968)	**5**(0.99)	5
Overlapping	0.95	3(4.26)	3(1.09)	**5**(2.87)	6(1.34)/30(0.41)	5(1.19)/30(0.41)	1(960)	**5**(0.97)	3/6
Iris	1.00	4(0.65)	24(0.33)	3(1.60)	30(0.11)/5(0.48)	30(0.13)/4(0.53)	25(962)	**3**(0.55)	3
Wine	1.0	1(1.22)	28 (0.93)	**3** (2.01)	30(0.59)/30(0.26)	7(0.50)/30(0.50)	19(964)	4 (0.65)	8
Libras	1.00	6(9.65)	30(1.96)	2(5.52)	30(1.66)/2(1.27)	30(1.42)/1(1.09)	**13**(964)	**13**(1.95)	15/16
Ecoli	0.9	6 (1.90)	25 (1.32)	3 (1.96)	30(0.51)/2(0.12)	11(0.38)/1(0.41)	**9**(967)	**7**(0.65)	7
Vowel	0.8	4 (5.67)	29 (1.53)	4 (2.9)	30(1.21)/27(0.32)	30(1.07)/19(0.33)	5(983)	**9**(1.35)	13
PenDigits	1.20	22(206.2)	29(19.41)	6(25.10)	30(7.52)/30(5.53)	30(7.16)/30(5.38)	-	**9**(9.27)	15
Poker(1-9)	1.4	4 (1889)	29 (1574)	2 (2080)	30(256)/30(926)	30(240)/30(915)	-	**10**(477)	-

the HV-index curve, we can still see that the index also achieved relatively high value at 3 number of clusters.

The HV-index is designed to optimize simultaneously the criteria of Equality (5.2). Since we are seeking to minimize both the entropy and the cohesion, the region of feasible solution has to be convex in the left-lower bound. If the algorithm can cluster the dataset well, the partition generated from it will be close to the bound of the region of feasible solution. Thus, the set of pairs $(H(\pi), \mathsf{sse}(\pi))$ for different partitions will form a convex curve. Figure 5.2 shows pairs of entropy and sse of k-means clustering results with different number of clusters on *Iris* and *Libras* dataset.

HV index scalarizes the Equation (5.2) using the hypervolume indicator. Both entropy and cohesion are normalized to values in $[0, 1]$. The entropy on partition is defined as the generalized entropy in Equation (2.1) with parameter β. As we will show in Section 5.4, different values of β will affects detection of the natural number of clusters. The β value we pick is given in Tables 5.2 for each dataset.

The natural number of clusters is successfully determined for all synthetic data sets. Especially, the HV index did not fail on the dataset with overlapping clusters for $\beta = 0.95$ even if the value on 3 is also relatively high (Figure 5.1c).

The variation of the HV index on other four synthetic and four real-world datasets, respectively are shown in Figure 5.1.

Table 5.2 gives the results of the application of the algorithm on total of 12 different datasets (5 synthetic and 7 real datasets). The HV index works well on all of those 5 synthetic datasets. Although HV failed to be the best for dataset *Wine*, it still achieves the second closest value on the optimal number of clusters.

In Table 5.2, we also show that our method outperforms some existing algorithms. Despite the large size of *Poker* dataset, HV-index proved to be strongly scalable. All experiments were performed on a 64-bit, Lenovo X1-Carbon laptop with Core i7 and 8GiB memory.

5.4 k-means, Hierarchical Clustering and Contour Curves

The natural number of clusters existing in a data set is evaluated by a repeated application of the k-means clustering algorithm in conjunction with a hierarchical clustering technique.

Hierarchical clustering is an agglomerative clustering technique that does not require the number of clusters in advance. Hierarchical algorithms cluster data by generating a tree structure or dendrogram. At each level of the tree the closest clusters (in the sense of a specific dissimilarity measure between clusters referred to as a merging scheme M) are fused into a cluster located on an upper level in the tree.

A particular partition of the dataset is obtained by cutting the cluster tree at a certain level. The successive cuts yield a chain of clusterings $(\pi_1, \ldots, \pi_q)$. These partitions are compared with the partitions $\sigma_1, \ldots, \sigma_p$ produced by the k-means algorithm for $1 \leqslant k \leqslant p$. Distances between k-means partitions π_i and hierarchical partitions σ_j are evaluated using the marching square representation [56].

A two-dimensional plot has the number of k-means clusters on the x-axis and the number of hierarchical clusters on the y-axis (as produced by the Ward hierarchical algorithm [61]).

We used the `hclust` and the `kmeans` functions of the R package [67] to compute the Ward-link hierarchical clustering and k-means clustering, respectively.

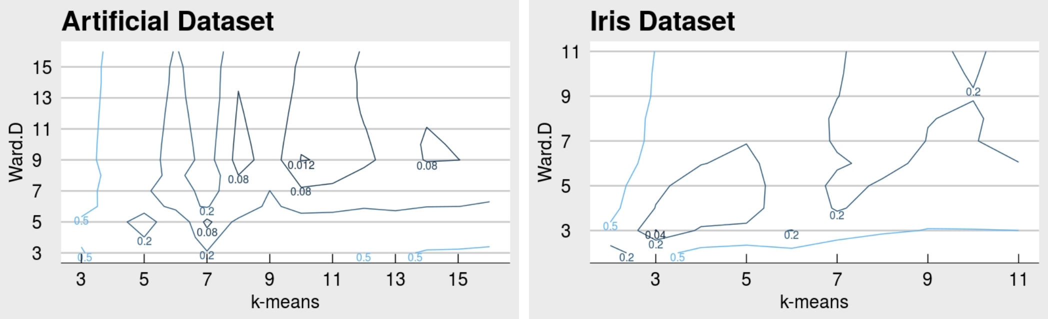
Using an interpolation process the marching square representation produces a *contour curve* defined, in our case, as curves that correspond to values of entropic distances between partition pairs of the form (π_i, σ_j).

Algorithm 2: Evaluation of the choice of number of clusters

```
Input: Dataset S, the range of the number of clusters $[k_{m_l}, k_{m_u}]$ for the
       $k$-means clustering, the range of number of clusters $[k_{h_l}, k_{h_u}]$ obtained
       by cutting the dendrogram of a hierarchical clustering, the merging
       scheme $M$ of a hierarchical clustering
Output: the contour C of the distance between $\pi_i, \forall i \in [k_{m_l}, k_{m_u}]$ and
        $\pi_j, \forall j \in [k_{h_l}, k_{h_u}]$
initialize the distance matrix D with size = $(k_{m_u} - k_{m_l}) \times (k_{h_u} - k_{h_l})$
  apply hierarchical clustering on S with merging scheme $M$ and generate a dendrogram $H$
  for $i$=$k_{m_l}$ to $k_{m_u}$ do
      $k$-means clustering on S with $i$ clusters and generate partition $\pi_i$
      for $j$=$k_{h_l}$ to $k_{h_u}$ do
          cut dendrogram $H$ and get the partition $\pi_j$ with $j$ clusters
            Calculate distance $d$ between $\pi_i$ and $\pi_j$;
          Set $\mathbf{D}_{i,j} = d(\pi_i, \pi_j)$;
plot the contour C from the distance matrix D by applying Marching Square
Algorithm [56]
return C
```

Note that Algorithm 2 needs no input parameters. The natural number of clusters can be determined using the contour curve that corresponds to a minimum distance.

Figure 5.5a gives the output for an artificially produced 10-cluster dataset of Algorithm 2. The contour curve that corresponds to the smallest distance (0.012) suggests that the natural number of clusters is indeed 10.



(a) 10-cluster Artificial Dataset

(b) *Iris* Dataset

Figure 5.5: The left figure is the contours of an artificial dataset with 10 Gaussian Distributed clusters. The right figure is the contours of *Iris* dataset. The y-axis represents the number of clusters with respect to Ward.D link hierarchical clustering, while the x-axis gives the parameter k for the k-means clustering.

The choice of clustering methods in Algorithm 2 must be done judiciously; indeed, if these clustering algorithms are very different, we could encounter datasets for which it is impossible to achieve a consensus on the number of clusters produced by these algorithms. On the other hand, if two algorithms are too similar to each other, then many clustering results will be similar to each other. The result will also not be plausible.

In our experiments, we use Ward-link hierarchical clustering and k-means clustering algorithm. Although these two algorithms share the same cost functions on clustering data, their procedures of clustering are quite different. When the number of clusters is equal to the number of data points in the dataset, the clustering results will always be the same no matter what clustering algorithms are applied on it. However, this is not the case in practice. Therefore, we only consider the number of clusters for a particular dataset within a particular range. In our experiments, we selected the range between 2 to $\sqrt{|S|} + 10$, $|S|$ is the number of points in set S.

5.5 Experiments with Unbalance Data

The flexibility afforded by generalized entropies allows choosing β to improve results in the case of imbalanced data sets.

Experiments suggest that small values of β may compensate for the sizes of small clusters and thus provide a more accurate estimation of the natural number of clusters. We verified this assumption on both synthetic and real data. A random portion of one of the clusters of the fourth synthetic data set shown in the fourth row of Figure 5.1 was removed and we sought to determine the number of clusters in the resulting imbalanced data using the dual criteria algorithm.

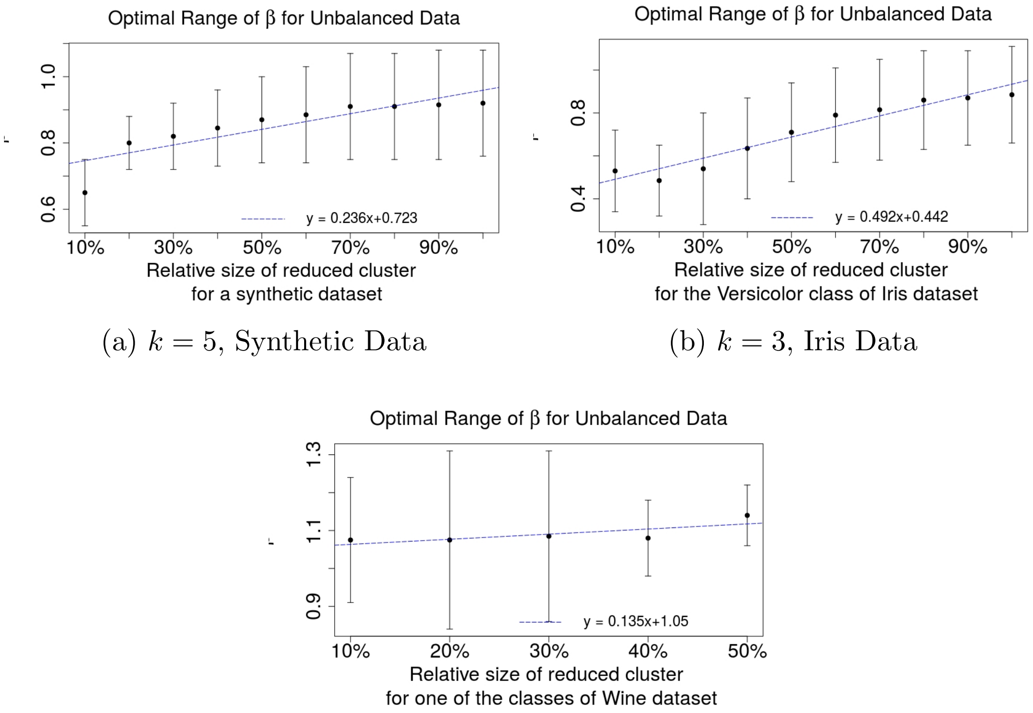
The same cluster modification was performed on the *Iris* data set by eliminating a portion of the *versicolor* cluster. As shown in Figure 5.6a, 5.6b, to retrieve the correct number (in our case, it is 5 and 3) better results are obtained with values of β that are less than 1.

For the data set *Wine*, there are three unbalanced clusters: the size of the largest one is almost twice as the size of the smallest one. To maintain consistency, we randomly removed 50% to 90% of the largest cluster, so that we can have roughly the same situation as in previous two examples. Figure 5.6c still shows a similar dependency of the quality of the clustering for smaller values of β.

For all three data sets, we record the average point of the range for each portion of the reduced cluster and apply linear regression. The regression results presented in Figure 5.6 show that all regression lines have positive slopes, which indicate that smaller values of β yield better results for large imbalances created by reduction in size of one of the clusters.

5.6 Conclusions

Our approach seeks to minimize both clustering partition entropy and the cohesion of clustering. The concept of Pareto Front is utilized to illustrate how to identify



(a) $k = 5$, Synthetic Data

(b) $k = 3$, Iris Data

(c) $k = 3$, Wine Data

Figure 5.6: Range of β that yields correct k clusters for the modified dataset.

the natural number of clusters existent in a data set. Experiments performed on both synthetic and real datasets confirm that this technique gives a relatively better indication on the natural number of clusters, comparing with existing methods.

Contour maps of the comparative results of two quite distinct cluster algorithms are used as a supplementary validation technique. If a natural clustering exists in the dataset, these distinct clustering algorithms will produce similar results with approximately the same number of clusters.

We intend to focus our attention on clustering imbalanced data, where the generalized entropy and a metric generated by this entropy seem promising. Qualitatively, we discover that if the majority of clusters are smaller in size, compared to the group with the largest size, we could choose a smaller β above 1 to achieve the optimal number of clusters, while if only few clusters are smaller in size, β should be chosen to be small as well and likely less than 1. This phenomenon can be seen in the results from previous experiments.

CHAPTER 6

CONCLUSIONS AND FURTHER STUDYING

In this thesis, we take efforts to solve the problems of cluster tendency and cluster validity. For dealing with cluster tendency, we applied the concept of ultrametric and designed a new definition of ultrametricity on a data set. To achieve this, we created a novel operation for matrix multiplication. If we apply this special matrix operation to the dissimilarity matrix of a data set, we can obtain a tool for computing the subdominant ultrametric of a dissimilarity and assessing the existence of an underlying clustering structure in a dissimilarity space.

The "power" operation successfully eliminates the redundant information in the dissimilarity matrix of the dataset but maintains the useful information that can discriminate the cluster structures of the dataset.

In a series of seminal papers[62, 63, 64], F. Murtagh argued that as the dimensionality of a linear metric space increases, an equalization process of distances takes place and the metric of the space gets increasingly closer to an ultrametric. This raises the issues related to the comparative evaluation (statistical and algebraic) of the ultrametricity of such spaces and of their clusterability, which we intend to examine in the future.

Based on the determination of clusterability of a data set, we can take advantage of the information of the clusterability to help us improve the quality of clustering tasks. Moreover, the intrinsic matrix multiplication process naturally has the ability of being parallelized. A GPU based parallel algorithm for calculating the clusterability can be easily executed. Since the ultrametric we finally get is the maximum sub-dominate

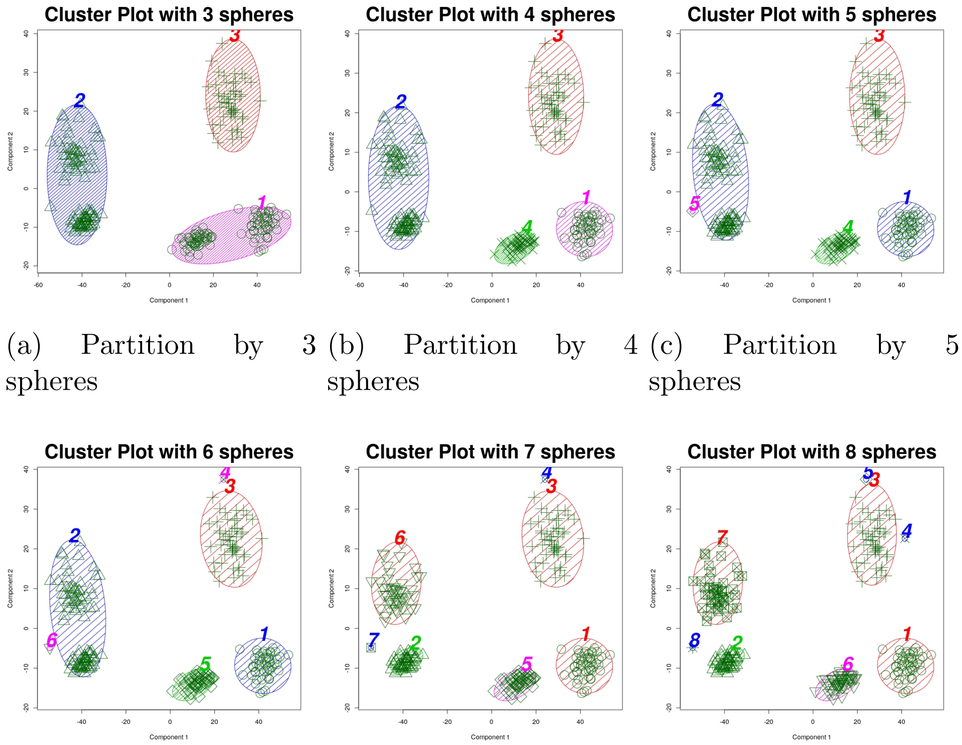
Algorithm 3: Get partition from ultrametric distance matrix (*getPrtUlt*)

```
Data: ultrametric distance value $u$, ultrametric distance matrix $U$ and
      dataset $D$
Result: partition $\sigma$ on distance matrix $U$
$Ut \leftarrow U$;
$\sigma \leftarrow \emptyset$;
$dLst \leftarrow \{i\}_{i=1}^{|D|}$; // create an index list to represent each data
 point
while $Ut \neq \emptyset$ do
    Let $Ut_1$ be the first row of $Ut$;
    $cid \leftarrow \{i \mid u \geq r_i, r_i \in Ut_1\}$; // all data points that close to
     point 1($\leq u$) should be in the same sphere.  The sphere is a
     cluster $C$
    $C = \{a_j \mid j \in dLst[cid], a_j \in D\}$;
    $\sigma \leftarrow \sigma \cup \{C\}$;
    $idx \leftarrow \{i \mid u < r_i, r_i \in Ut_1\}$;
    $Ut \leftarrow Ut[idx, idx]$; // update distance matrix with points outside
     sphere
    $dLst \leftarrow dLst[idx]$; // update data index list with points outside
     sphere
end
return $\sigma$
```

ultrametric, it will have a strong potential in resolving several current challenges in data clustering, such as outlier detection and determination of the number of clusters.

Retrieving the knowledge from Chapter 2 and Section 3.5 of Chapter 3, ultrametric space is a so-called partitional metric space. If we pick a distance value of r, we can separate the whole data space into several disjoint balls with radius r. If r is larger, we expect to have a smaller number of balls, and if r is smaller, we expect to have more balls after separation. Algorithm 3 gives a simple way to partition the dataset with a given distance value.

When we create an r-spherical clustering of the dataset for a particular r, we seek to determine whether each ball of the partition contains points from only one of the natural clusters of the dataset.Since an r-spherical clustering is generated from



(a) Partition by 3 spheres

(b) Partition by 4 spheres

(c) Partition by 5 spheres

(d) Partition by 6 spheres

(e) Partition by 7 spheres

(f) Partition by 8 spheres

Figure 6.1: Experiment on a dataset with five Gaussian distributed clusters.

an ultrametric distance matrix, we want to make sure it also reflects the clustering structure of the original metric space.

If we choose a sufficiently large value of r, namely the largest value from the ultrametric distance matrix, we get one ball which contains the whole dataset. Here, the ball will surely contain more than one natural cluster of the data set. Conversely, by choosing r to be 0, we get a partition where every point forms a cluster itself. In this minimal case, every cluster must contain points from only one of the natural clusters. Thus, Between the minimum and maximum of r values, there exists a maximum value u, such that if we use u to partition the dataset, each ball with radius u only contains one natural cluster of the dataset. We consider this u to be the ideal radius of partition of the data set. Algorithm 4 explicates the procedure

Algorithm 4: Get maximum value of radius

```
Data: Dataset $D$ with natural cluster structures, groudtruth partition $\pi$ on
      $D$
Result: $u_k$, it is the value that we need ultrametric sphere with radius at
        most $u_k$ to cover original dataset such that each sphere only
        contains data points from one original cluster of the dataset
$U \leftarrow CalcUltraMtx(D)$ ; // calculate ultrametric distance matrix
 of $D$
$\{u_i\}_{i=1}^{|D|} \leftarrow sortUnique(U)$; // sort distinct distance values in
 decreasing order
$k = 1$;
repeat
    $\sigma \leftarrow getPrtUlt(u_k, U, D)$;
    $check \leftarrow True$;
    foreach cluster $C \in \sigma$ do ; // The shape of $C$ is a sphere in
     ultrametric space with radius $u_k$

        Let $\pi_C \leftarrow$ the trace of $\pi$ on $C$;
        if $|\pi_C| > 1$ then
            $check \leftarrow Flase$;
            break the inner loop;
        end
    end
    $k \leftarrow k + 1$;
until $check \leftarrow True$ or $k = |D|$;; // no sphere contain data points
 from more than one cluster

return $u_k$
```

of finding u. $CalcUltraMtx$ is the function of using min-max matrix multiplication to calculate the ultrametric distance matrix from dataset D. $sortUnique$ sorts the distinct distance values of ultrametric distance matrix in decreasing order.

We perform the algorithm on a well-separated dataset with five Gaussian distributed clusters. Figure 6.1 shows the distribution of natural clusters on different clustering results. As we can see, when we partition the dataset with 7^{th} largest value of the ultrametric distance matrix, each ball contains only one natural cluster. Among these seven balls, two of them have 0 radii, because they only include one element respectively. If we regard these two elements as the outliers, the remaining five balls each correspond to a natural cluster. This phenomenon suggests we can use the size of spheres generated by the r-spherical clustering to detect the number of clusters and outliers.

In checking cluster validity, we mainly focus on the determination of the number of clusters. Besides the method we proposed in Chapter 5, another direction of investigation is seeking to integrate mean-shift techniques with other evaluation criteria for clustering such as cohesion or size balancing. Mean-shift was introduced by Fukunaga and Hostetler [33] for seeking the mode of a density function represented by a set S of samples. This procedure uses kernels (defined as decreasing functions of the distance from a given point $\mathbf{t}$ to a point $\mathbf{s}$ in S). For every point $\mathbf{t}$ in a given set T , the sample means of all points in S weighted by a kernel at $\mathbf{t}$ are computed to form a new version of T . This computation is repeated until convergence. The resulting set T contains estimates of the modes of the density underlying set S. Cheng [18] developing a more general formulation and demonstrated its uses in clustering and global optimization. He showed that mean shift is an instance of gradient ascent and that mean shift has an adaptive step size.

More recently, M. Fashing and C. Tomasi [30] proved that for of piecewise constant kernels, the step is exactly the Newton step and, in all cases, it is a step to the maximum of a quadratic bound. They proved that mean-shift is an optimization procedure.

The work of Comaniciu and Meer [22] refocused the attention on the mean-shift procedure due to its applications in image segmentation and discontinuity-preserving smoothing. They proved that for discrete data the convergence of a recursive mean-shift to the nearest stationary point of the underlying density function and, therefore, its utility in detecting the the modes of the density, and, therefore, the “natural” number of clusters existent in data.

BIBLIOGRAPHY

[1] Ackerman, M., Adolfsson, A., and Brownstein, N. An effective and efficient approach for clusterability evaluation. *CoRR abs/1602.06687* (2016).

[2] Ackerman, M., Ben-David, S., and Loker, D. Towards property-based classification of clustering paradigms. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pp. 10–18.

[3] Ackerman, Margareta, and Ben-David, Shai. Clusterability: A theoretical study. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009* (2009), pp. 1–8.

[4] Ackerman, Margareta, Ben-David, Shai, Brânzei, Simina, and Loker, David. Weighted clustering. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.* (2012), pp. 858–863.

[5] Adolfsson, Andreas, Ackerman, Margareta, and Brownstein, N. C. To cluster, or not to cluster: An analysis of clusterability methods. *CoRR abs/1808.08317* (2018).

[6] Aeberhard, S., Coomans, D., and Vel, O. De. Comparison of classifiers in high dimensional settings. *Dept. Math. Statist., James Cook Univ., North Queensland, Australia, Tech. Rep*, 92-02 (1992).

[7] Akaike, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control 19* (1974), 667–674.

[8] Akaike, H. Information theory and an extension of the maximum likelihood principle. In *Selected papers of Hirotugu Akaike*. Springer, 1998, pp. 199–213.

[9] Al-Shaqsi, J., and W.Wang. A novel three staged clustering algorithm. In *IADIS European Conference on Data Mining, Algarve,Portugal* (2009), pp. 9–16.

[10] Ayed, I. B., and Mitiche, A. A region merging prior for variational level set image segmentation. *IEEE transactions on image processing 17*, 12 (2008), 2301–2311.

[11] Azimi, Javad, and Fern, Xiaoli. Adaptive cluster ensemble selection. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009* (2009), pp. 992–997.

[12] Balcan, M. F., Blum, A., and Vempala, S. A discriminative framework for clustering via similarity functions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008* (2008), pp. 671–680.

[13] Becker, R. A., Chambers, J. M., and Wilks, A. R. The new s language. *Pacific Grove, Ca.: Wadsworth & Brooks, 1988 1* (1988).

[14] Ben-David, S. Computational feasibility of clustering under clusterability assumptions. *CoRR abs/1501.00437*.

[15] Ben-David, S., and Ackerman, M. Measures of clustering quality: A working set of axioms for clustering. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008* (2008), pp. 121–128.

[16] Ben-Hur, A., Elisseeff, A., and Guyon, I. A stability based method for discovering structure in clustered data. In *Proceedings of the 7th Pacific Symposium on Biocomputing, PSB 2002, Lihue, Hawaii, USA, January 3-7, 2002* (2002), pp. 6–17.

[17] Birkhoff, G. *Lattice Theory*, third ed. American Mathematical Society, Providence, RI, 1973.

[18] Cheng, Y. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence 17*, 8 (1995), 790–799.

[19] Cheung, Yiu-ming. On rival penalization controlled competitive learning for clustering with automatic cluster number selection. *IEEE Transactions on Knowledge and Data Engineering 17*, 11 (2005), 1583–1588.

[20] Chiu, Tom, Fang, DongPing, Chen, John, Wang, Yao, and Jeris, Christopher. A robust and scalable clustering algorithm for mixed type attributes in large database environment. In *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining* (2001), pp. 263–268.

[21] Coello, C. A., Lamont, G. B., Van Veldhuizen, D., et al. *Evolutionary algorithms for solving multi-objective problems.* Springer, 2007.

[22] Comaniciu, D., and Meer, P. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence 24*, 5 (2002), 603–619.

[23] Daniely, Amit, Linial, Nati, and Saks, Michael E. Clustering is difficult only when it does not matter. *CoRR abs/1205.4891* (2012).

[24] Daroczy, Z. Generalized information function. *Information and Control 16* (1970), 36–51.

[25] de Mántaras, Ramon López. A distance-based attribute selection measure for decision tree induction. *Machine Learning 6* (1991), 81–92.

[26] Deza, M. M., and Laurent, M. *Geometry of Cuts and Metrics.* Springer, Heidelberg, 1997.

[27] Ding, Chris H. Q., and He, Xiaofeng. Cluster aggregate inequality and multilevel hierarchical clustering. In *Knowledge Discovery in Databases: PKDD 2005, 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3-7, 2005, Proceedings* (2005), pp. 71–83.

[28] Epter, S., Krishnamoorthy, M., and Zaki, M. Clusterability detection and initial seed selection in large datasets. In *The International Conference on Knowledge Discovery in Databases* (1999), vol. 7.

[29] Ester, Martin, Kriegel, Hans-Peter, Sander, Jörg, Xu, Xiaowei, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (1996), vol. 96, pp. 226–231.

[30] Fashing, M., and Tomasi, C. Mean shift is a bound optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 3 (2005), 471–474.

[31] Fern, Xiaoli Zhang, and Brodley, Carla E. Solving cluster ensemble problems by bipartite graph partitioning. In *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004* (2004).

[32] Foss, Andrew, and Zaïane, Osmar R. A parameterless method for efficiently discovering clusters of arbitrary shape in large datasets. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on* (2002), pp. 179–186.

[33] Fukunaga, K., and Hostetler, L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory 21*, 1 (1975), 32–40.

[34] Genz, Alan, Bretz, Frank, Miwa, Tetsuhisa, Mi, Xuefei, Leisch, Friedrich, Scheipl, Fabian, Bornkamp, Bjoern, Maechler, Martin, Hothorn, Torsten, and Hothorn, Maintainer Torsten. Package mvtnorm.

[35] Gionis, Aristides, Mannila, Heikki, and Tsaparas, Panayiotis. Clustering aggregation. *TKDD 1*, 1 (2007), 1–30.

[36] Gomes, R., Krause, A., and Perona, P. Discriminative clustering by regularized information maximization. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.* (2010), pp. 775–783.

[37] Hamerly, Greg, and Elkan, Charles. Learning the k in k-means. In *Advances in neural information processing systems* (2004), pp. 281–288.

[38] Havrda, J. F., and Charvat, F. Qualification method of classification processes, the concept of structural k-entropy. *Kybernetika 3* (1967), 30–35.

[39] Hennig, Christian. *fpc: Flexible Procedures for Clustering*, 2015. R package version 2.1-10.

[40] Jain, Anil K. Data clustering: 50 years beyond k-means. *Pattern recognition letters 31*, 8 (2010), 651–666.

[41] Kalogeratos, Argyris, and Likas, Aristidis. Dip-means: an incremental clustering method for estimating the number of clusters. In *Advances in neural information processing systems* (2012), pp. 2393–2401.

[42] Karim, A., Loqman, C., and Boumhidi, J. Determining the number of clusters using neural network and max stable set problem. *Procedia Computer Science 127* (2018), 16–25.

[43] Kaufman, L., and Rousseeuw, P. J. *Finding Groups in Data – An Introduction to Cluster Analysis.* John Wiley & Sons, New York, 1990.

[44] Kearns, M., Mansour, Y., and Ng, A. Y. An information-theoretic analysis of hard and soft assignment methods for clustering. In *Learning in graphical models.* Springer, 1998, pp. 495–520.

[45] Kolesnikov, Alexander, Trichina, Elena, and Kauranne, Tuomo. Estimating the number of clusters in a numerical data set via quantization error modeling. *Pattern Recognition 48*, 3 (2015), 941–952.

[46] Leclerc, B. Description combinatoire des ultramétriques. *Mathématiques et science humaines 73* (1981), 5–37.

[47] Lerman, I. C. *Classification et Analyse Ordinale des Données.* Dunod, Paris, 1981.

[48] Li, Tao, and Ding, Chris H. Q. Weighted consensus clustering. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2008, April 24-26, 2008, Atlanta, Georgia, USA* (2008), pp. 798–809.

[49] Li, Tao, Ding, Chris H. Q., and Jordan, Michael I. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA* (2007), pp. 577–582.

[50] Liang, Jiye, Zhao, Xingwang, Li, Deyu, Cao, Fuyuan, and Dang, Chuangyin. Determining the number of clusters using information entropy for mixed data. *Pattern Recognition 45*, 6 (2012), 2251–2265.

[51] Lichman, M. UCI machine learning repository, 2013.

[52] Liu, Hongfu, Liu, Tongliang, Wu, Junjie, Tao, Dacheng, and Fu, Yun. Spectral ensemble clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), pp. 715–724.

[53] Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. Understanding of internal clustering validation measures. In *2010 IEEE 10th International Conference on Data Mining* (2010), IEEE, pp. 911–916.

[54] Maechler, Martin, Rousseeuw, Peter, Struyf, Anja, Hubert, Mia, and Hornik, Kurt. *cluster: Cluster Analysis Basics and Extensions*, 2016. R package version 2.0.5 — For new features, see the 'Changelog' file (in the package source).

[55] Manning, Christopher D., Raghavan, Prabhakar, and Schütze, Hinrich. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.

[56] Maple, C. Geometric design and space planning using the marching squares and marching cube algorithms. In *Proceedings of the International Conference on Geometric Modeling and Graphics* (2003), IEEE, pp. 90–95.

[57] Marler, R. T., and Arora, J. S. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization 26*, 6 (2004), 369–395.

[58] Maulik, U., and Bandyopadhyay, S. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 12 (2002), 1650–1654.

[59] Monti, Stefano, Tamayo, Pablo, Mesirov, Jill, and Golub, Todd. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning 52*, 1-2 (2003), 91–118.

[60] Mouselimis, Lampros. *ClusterR: Gaussian Mixture Models, K-Means, Mini-Batch-Kmeans, K-Medoids and Affinity Propagation Clustering*, 2018. R package version 1.1.6.

[61] Murtagh, F., and Legendre, P. Wards hierarchical agglomerative clustering method: Which algorithms implement Wards criterion? *Journal of Classification 31*, 3 (2014), 274–295.

[62] Murtagh, Fionn. Quantifying ultrametricity. In *COMPSTAT* (2004), pp. 1561–1568.

[63] Murtagh, Fionn. Clustering in very high dimensions. In *UK Workshop on Computational Intelligence* (2005), p. 226.

[64] Murtagh, Fionn. Identifying and exploiting ultrametricity. In *Advances in Data Analysis*. Springer, 2007, pp. 263–272.

[65] Neal, R. M., and Hinton, G. E. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models, NATO Science Series*, M. I. Jordan, Ed. Kluwer, 1998, pp. 355–368.

[66] Pareto, V. *Manuale di economia politica*, vol. 13. Societa Editrice, 1906.

[67] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2016.

[68] Roth, Volker, Lange, Tilman, Braun, Mikio, and Buhmann, Joachim. A resampling approach to cluster validation. In *Compstat* (2002), Springer, pp. 123–128.

[69] Salvador, Stan, and Chan, Philip. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on* (2004), pp. 576–584.

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

[71] Scott Harris, R, Hess, Dean R, and Venegas, José G. An objective analysis of the pressure-volume curve in the acute respiratory distress syndrome. *American Journal of Respiratory and Critical Care Medicine 161*, 2 (2000), 432–439.

[72] Shaqsi, J. Al, and Wang, W. Estimating the predominant number of clusters in a dataset. *Intelligent Data Analysis 17*, 4 (2013), 603–626.

[73] Silva, S. Da, and Rathie, P. Shannon, Lévy and Tsallis: A note. *Applied Mathematical Sciences 2* (2008), 1359–1363.

[74] Simovici, D., and Djeraba, C. *Mathematical Tools for Data Mining*, second ed. Springer, London, 2014.

[75] Simovici, Dan A. On generalized entropy and entropic metrics. *Multiple-Valued Logic and Soft Computing 13*, 4-6 (2007), 295–320.

[76] Simovici, Dan A., and Jaroszewicz, Szymon. An axiomatization of partition entropy. *IEEE Trans. Information Theory 48*, 7 (2002), 2138–2142.

[77] Sugar, Catherine A, and James, Gareth M. Finding the number of clusters in a dataset. *Journal of the American Statistical Association* (2011).

[78] Tang, P.N., Steinbach, M., and Kumar, V. *Introduction to Data Mining.* Addison-Wesley, Reading, MA, 2005.

[79] Tibshirani, R., and Walther, G. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics 14*, 3 (2005), 511–528.

[80] Tibshirani, Robert, Walther, Guenther, and Hastie, Trevor. 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 (2001), 411–423.

[81] Tsallis, C. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics 52* (1988), 479–487.

[82] von Winckel, G. Determining the number of clusters via iterative consensus clustering.

[83] Wang, Liang, Leckie, Christopher, Ramamohanarao, Kotagiri, and Bezdek, James. Automatically determining the number of clusters in unlabeled data sets. *IEEE Transactions on knowledge and Data Engineering 21*, 3 (2009), 335–350.

[84] Wu, Junjie, Liu, Hongfu, Xiong, Hui, Cao, Jie, and Chen, Jian. K-means-based consensus clustering: A unified view. *IEEE Transactions on Knowledge and Data Engineering 27*, 1 (2015), 155–169.

[85] Xiong, H., Wu, J., and Chen, J. K-means clustering versus validation measures: a data-distribution perspective. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39*, 2 (2009), 318–331.

[86] Yu, H., Liu, Z., and Wang, G. An automatic method to determine the number of clusters using decision-theoretic rough set. *International Journal of Approximate Reasoning 55*, 1 (2014), 101–115.

[87] Zhao, Y., and Karypis, G. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management* (2002), ACM, pp. 515–524.

[88] Zheng, Li, Li, Tao, and Ding, Chris H. Q. A framework for hierarchical ensemble clustering. *TKDD 9*, 2 (2014), 9:1–9:23.

[89] Zitzler, E., and Thiele, L. Multiobjective optimization using evolutionary algorithms-a comparative case study. In *International Conference on Parallel Problem Solving from Nature* (1998), Springer, pp. 292–301.